

IBM® WebSphere Commerce



Guía para el desarrollo de tiendas

Versión 5.5

IBM® WebSphere Commerce



Guía para el desarrollo de tiendas

Versión 5.5

Aviso:

Antes de utilizar esta información y el producto al que da soporte, lea la información del apartado Avisos.

Primera edición (Junio 2003)

Esta edición se aplica a los productos siguientes:

- IBM WebSphere Commerce Business Edition, Versión 5.5
- IBM WebSphere Commerce Professional Edition, Versión 5.5

Esta edición también se aplica a todos los releases y modificaciones posteriores de los productos listados anteriormente, hasta que se indique lo contrario en nuevas ediciones. Asegúrese de utilizar la edición correcta para el nivel del producto.

Efectúe el pedido de publicaciones a través del representante de IBM o de la sucursal de IBM que atiende a su localidad. En la dirección que figura a continuación no hay existencias de publicaciones.

IBM agradece sus comentarios. Puede enviar sus comentarios utilizando la hoja de comentarios de la documentación de IBM WebSphere Commerce que encontrará en el URL siguiente:

www.ibm.com/software/webservers/commerce/ref/html

Cuando se envía información a IBM, se otorga a IBM un derecho no exclusivo para utilizar o distribuir la información de la forma que considere apropiada, sin incurrir por ello en ninguna obligación para con el remitente.

Cuando se envía información a IBM, se otorga a IBM un derecho no exclusivo para utilizar o distribuir la información de la forma que considere apropiada, sin incurrir por ello en ninguna obligación para con el remitente.

© Copyright International Business Machines Corporation 2003. Reservados todos los derechos.

Contenido

Antes de empezar	ix
Convenios y terminología utilizados en este manual	ix
Variables utilizadas en este manual	x
Variables de vía de acceso.	xi
Dónde encontrar la información más reciente	xii

Parte 1. Visión general 1

Capítulo 1. Visión general del desarrollo de tiendas 3

Descripción del desarrollo de tiendas en WebSphere Commerce	3
La finalidad de la tienda	3
El modelo de negocio representativo de su tienda	3
El número de tiendas que se están desarrollando	4
La base de la tienda	5
Grado de personalización necesario.	8
Escenario: Desarrollo y despliegue de una tienda de producción	10

Parte 2. Modelos de negocio soportados por WebSphere Commerce 15

Capítulo 2. Modelos de negocio soportados en WebSphere Commerce . 17

Descripción de los modelos de negocio soportados en WebSphere Commerce.	17
Venta directa	17
Alojamiento	18
Cadena de valor.	19
Tiendas de ejemplo en WebSphere Commerce	22

Parte 3. Arquitectura de WebSphere Commerce 25

Capítulo 3. Estructura organizativa de WebSphere Commerce 27

Descripción de la estructura organizativa de WebSphere Commerce.	27
Cómo la estructura organizativa da soporte a los modelos de negocio	28
Directo al consumidor	28
Directo a B2B.	29
Cadena de demanda	31
Cadena de oferta	32
Alojamiento	34
Estructuras de organización de ejemplo	36
Creación de estructuras de organización.	36

Capítulo 4. Control de acceso en WebSphere Commerce 37

Descripción del control de acceso en WebSphere Commerce.	37
Políticas de control de acceso	37
Descripción del control de acceso en los modelos de negocio.	40
Estructura de control de acceso básica	40
Directo al consumidor	42
Directo a B2B.	44
Cadena de demanda	45
Cadena de oferta	52
Alojamiento	55
Control de acceso en negocios de ejemplo	61
Adición de control de acceso a las tiendas	61

Capítulo 5. Infraestructura de políticas de negocio de WebSphere Commerce . 63

Descripción de la infraestructura de políticas de negocio de WebSphere Commerce	63
Políticas de negocio.	63
Cuentas de negocio.	63
Contratos y acuerdos de servicio	63
Términos y condiciones	64
Políticas de negocio en negocios de ejemplo	64
Adición de políticas de negocio al sitio	64

Capítulo 6. Arquitectura de instancias 65

WebSphere Commerce Server	65
Instancia de WebSphere Commerce Server	65

Capítulo 7. Arquitectura de la tienda . . 67

Descripción de la arquitectura de tienda de WebSphere Commerce.	67
Elementos de tienda	67
Varias tiendas en una sola instancia	68
Relaciones entre tiendas	70
Descripción del soporte de modelos de negocio en la arquitectura de tienda	70
Tiendas cara al público	70
Tiendas proxy	73
Tiendas con elementos.	74
Tiendas de los modelos de negocio soportados	75

Parte 4. Desarrollo del escaparate 79

Capítulo 8. Desarrollo del escaparate 81

Arquitectura del escaparate	81
Mandatos y vistas por omisión	81
Creación de las páginas de tienda	82
Desarrollar una lista de las páginas de la tienda	82
Desarrollar una lista de los URL de mandatos y vistas	86
Asociar nombres de archivo JSP a las vistas	87

Capítulo 9. Almacenamiento en antememoria de las páginas de tienda . 89

Planificación de la estrategia de almacenamiento en antememoria	89
Páginas que deben almacenarse en la antememoria	89
Las páginas se han de almacenar en la antememoria como páginas completas o como fragmentos de páginas.	89
Desarrollo de una estrategia de almacenamiento en antememoria más detallada	90
Cómo se solicita la página o fragmento	90
La página o fragmento se basa en una relación de tienda	90
Cómo se invalidarán los datos almacenados en la antememoria	91
Implementación de la estrategia de almacenamiento	91
Descripción del archivo cachespec.xml	92
Invalidación de datos almacenados en antememoria en el archivo cachespec.xml	96
Implementación del almacenamiento en antememoria de las páginas de tienda que utilicen relaciones entre tiendas.	97
Sustitución de las funciones de CacheCommand por el almacenamiento en antememoria dinámico	101

Parte 5. Visión general de los datos de la tienda 103

Capítulo 10. Datos de la tienda 105

¿Qué son los datos de la tienda?	105
Modelo de información de los datos de la tienda	105
Modelo de información de datos de la tienda visto por subsistema	107
Modelo de información de datos de la tienda visto por tipo de datos	108
Tipos de datos de la tienda y negocios de ejemplo	112
Herramientas para la creación de datos.	112
Loader package de WebSphere Commerce.	112
Consola de administración	112
WebSphere Commerce Accelerator	112
Consola de administración de organizaciones	112
Diagrama de herramientas y datos de la tienda	113

Parte 6. Desarrollo de los datos de la tienda 117

Capítulo 11. Elementos de sitio. 119

Los elementos de sitio en WebSphere Commerce	119
Idioma	120
Atributos de miembros	120
Tipos de atributo	120
Tipos de grupos de miembros	120
Usuario	120
Organización	120
Rol	121
Conversión de unidades de cantidad	121

Unidades de cantidad	121
Tipos de impuestos	121
Uso de cálculo	121
Moneda	121
Utilización de números	122
Tipos de artículo	122
Formatos de dispositivo	122
Tipos de relaciones entre tiendas	122
Datos de acuerdo comercial a nivel de sitio	122
Tipo de acuerdo comercial	123
Rol de participante	123
Tipo de política.	123
Tipo de términos y condiciones	123
Atributo de personalización	124
Tipo de atributo	124
Operador.	124
Uso de adjuntos	124
Creación de elementos de sitio en WebSphere Commerce	124

Capítulo 12. Elementos de miembro . 125

Los elementos de miembro en WebSphere Commerce	125
Miembros	126
Atributos de miembro	127
Roles	127
Los elementos de cliente en WebSphere Commerce	128
Información de dirección	128
Listas de intereses	129
Descripción de los elementos de vendedor en WebSphere Commerce	129
Tiendas	130
Cuentas	131
Contratos.	131
Conjuntos de productos	131
Listas de precios	132
Catálogos.	132
Centros de despacho de pedidos	132
Artículos de inventario	133
Descripción de los elementos de administrador en WebSphere Commerce	133
Creación de elementos de miembro en WebSphere Commerce	133

Capítulo 13. Elementos de tienda. . . 135

Los elementos de tienda en WebSphere Commerce	135
Entidad de tienda	135
Creación de elementos de tienda en WebSphere Commerce	136
Creación de elementos de datos de tienda en un archivo XML	136

Capítulo 14. Relaciones entre tiendas . 143

Descripción de las relaciones entre tiendas en WebSphere Commerce	143
Relaciones entre tiendas	143
Tipos de relaciones entre tiendas	143
Descripción del tipo de relación entre tiendas	146
Creación de relaciones entre tiendas en WebSphere Commerce	146

Capítulo 15. Datos de los registros de URL, de mandatos y de vistas 149

Descripción de los registros de URL, de mandatos y de vistas en WebSphere Commerce	149
Registro de URL	149
Registro de mandatos	150
Registro de vistas	150
Creación de nuevos mandatos, vistas y URL	151
Cómo registrar mandatos, vistas y URL en WebSphere Commerce	151
Creación de un archivo XML para registrar mandatos, vistas y URL	151

Capítulo 16. Elementos de catálogo 155

Los catálogos en WebSphere Commerce	155
Catálogo	156
Grupos de catálogo	156
Entradas de catálogo	157
Conjuntos de productos	158
Atributos	159
Valores de atributo	159
Atributos de paquete	159
Valores de atributo de paquete	159
Creación de elementos de catálogo en WebSphere Commerce	159
Creación de un catálogo maestro	160
Visualización de los elementos de catálogo de la tienda	177
Creación de un catálogo de ventas	179
Adición de un producto a una segunda categoría	180
Gestión de elementos de catálogo en WebSphere Commerce	182
Grupos de catálogo	182
Entradas de catálogo	183
Herramientas de Gestión de productos	184
Loader Package	185

Capítulo 17. Elementos de fijación de precios 187

Descripción de la fijación de precios en WebSphere Commerce	187
Oferta	188
Precio de oferta	188
Contenedor de propuestas de comercio	188
Términos y condiciones	188
Tipos de términos y condiciones de fijación de precios	188
Acuerdo comercial	189
Participante	189
Rol de participante	190
Contrato	190
Política de negocio	190
Política de precios	190
Envío de entrada de catálogo	190
Otros elementos asociados a la fijación de precios	190
Creación de elementos de fijación de precios en WebSphere Commerce	191

Creación de elementos de fijación de precios en un archivo XML	191
--	-----

Capítulo 18. Elementos de contrato 195

Descripción de los contratos en WebSphere Commerce	196
Cuentas (cuentas de negocio)	196
Contratos	197
Acuerdos comerciales	197
Participantes	198
Términos y condiciones	198
Políticas de negocio	202
Adjunto	203
Artículo de pedido	203
Creación de un elemento de contrato por omisión en WebSphere Commerce	203
Creación de archivos XML de políticas de negocio	205
Creación de un archivo de contrato por omisión	206

Capítulo 19. Elementos de despacho de pedidos 215

Los elementos de despacho de pedidos en WebSphere Commerce	216
Centro de despacho de pedidos	216
Recepciones	217
RaDetail	217
Inventario	217
Acuerdos de envío	217
Otros elementos relacionados con el despacho de pedidos	217
Creación de elementos de despacho de pedidos en WebSphere Commerce	218
Creación de elementos de despacho de pedidos (no ATP)	219

Capítulo 20. Elementos de campaña 223

Las campañas en WebSphere Commerce	223
Creación de elementos de campaña en WebSphere Commerce	225

Capítulo 21. Herramientas de Payments 227

Creación de elementos de pago utilizando un archivo XML	228
Personalización del entorno para un casete de pago	229
Modificar el archivo .jsp de la tienda	229
Comprobar el perfil de cajero para el casete	231
Comprobar el archivo .jsp del casete	232
Configurar los valores de comerciantes en WebSphere Commerce Payments	232

Capítulo 22. Elementos de idioma . . . 233

Los elementos de idioma en WebSphere Commerce	233
Idioma por omisión	233
Idiomas soportados	234
Idioma alternativo	234
Creación de elementos de idioma en WebSphere Commerce	234

Capítulo 23. Elementos de moneda 235

Los elementos de moneda en WebSphere Commerce	235
Formato de moneda	236
Utilización de números	236
Descripción del formato de moneda	237
Moneda soportada	237
Norma de conversión de moneda	237
Moneda de contravalor	237
Creación de elementos de moneda en WebSphere Commerce	238
Creación de elementos de moneda utilizando un archivo XML	238

Capítulo 24. Elementos de unidad de medida 243

Las unidades de medida en WebSphere Commerce	243
Unidad de cantidad y formato de la unidad de cantidad	244
Creación de unidades de medida en WebSphere Commerce	245

Capítulo 25. Elementos de jurisdicción 247

Los elementos de jurisdicción en WebSphere Commerce	247
Creación de elementos de jurisdicción en WebSphere Commerce	248

Capítulo 26. Elementos de envío . . . 249

Los elementos de envío en WebSphere Commerce	249
Modalidades de envío	250
Códigos de cálculo	250
Jurisdicciones y grupos de jurisdicciones	251
Creación de elementos de envío en WebSphere Commerce	251
Creación de elementos de envío utilizando un archivo XML	252
Creación de elementos de despacho de envío	261
Creación de elementos de tienda-catálogo-envío	263
Creación de una modalidad de envío por omisión	264

Capítulo 27. Elementos de impuestos 265

Los elementos de impuestos en WebSphere Commerce	265
Categoría de impuestos	266
Código de cálculo	266
Jurisdicciones y grupos de jurisdicciones	268
Creación de elementos de impuestos en WebSphere Commerce	268
Creación de elementos de impuestos utilizando un archivo XML	269
Creación de elementos de despacho de impuestos	279
Creación de elementos tienda-catálogo-impuestos	280

Capítulo 28. Elementos de descuento 283

Descripción de los descuentos basados en normas en WebSphere Commerce	283
Moneda por omisión de la tienda	284
Código de cálculo	284
RLPromotion	284
Proyecto de normas de Blaze	285
Servicio de normas de Blaze	285
Servicio de descuentos	285
Servidor de normas de Blaze	285
Descripción de los descuentos basados en esquema en WebSphere Commerce	286
Código de cálculo	286
Creación de elementos de descuento en WebSphere Commerce	287

Capítulo 29. Elementos de inventario 289

Los elementos de inventario en WebSphere Commerce	290
Inventario ATP	290
Inventario no ATP	292
Creación de elementos de inventario en WebSphere Commerce	293
Gestión de códigos de ajuste de inventario	293
Adición de códigos de ajuste de inventario	295
Cambio de códigos de ajuste de inventario	296
Supresión de códigos de ajuste de inventario	296

Capítulo 30. Elementos de pedido . . 299

Los elementos de pedido en WebSphere Commerce	299
Pedidos y artículos de pedido	299
Artículos de pedido	301
Relaciones de solicitud de presupuesto	303
Creación de elementos de pedido en WebSphere Commerce	304

Capítulo 31. Elementos de proveedor 305

Descripción de los elementos de proveedores de WebSphere Commerce	305
Creación de elementos de proveedor	306

Capítulo 32. Perfiles de cliente 307

Descripción de los perfiles de cliente en WebSphere Commerce	307
--	-----

Parte 7. Adición de control de acceso a la tienda 309**Capítulo 33. Control de acceso en la tienda. 311**

Descripción del control de acceso en WebSphere Commerce	311
Control de acceso en las tiendas	311
Adición de control de acceso a la tienda	315
Creación o edición de control de acceso en la tienda	315

Parte 8. Globalización de la tienda 319

Capítulo 34. Globalización	321
Soporte de la globalización	321
Tiendas de ejemplo	322
Formatos de visualización	326
Creación de un formato de visualización nuevo	326
Creación de una tienda globalizada	330
Creación de una tienda	330
Gestión de la plantilla para un sitio globalizado	330
Modelo de programación de una plantilla para	
todas las tiendas e idiomas	332
Adición de un idioma a una tienda	334
Creación de un catálogo en línea globalizado	335
Gestionar elementos de globalización	336
Traducción de archivos de propiedades.	336

Parte 9. Empaquetado de la tienda 337

Capítulo 35. Empaquetado de una tienda.	339
Creación de un archivador de tienda	341
Creación de un archivador de tienda de ejemplo	343

Parte 10. Publicación de la tienda 345

Capítulo 36. Publicación de una tienda completa	347
Descripción de la publicación en WebSphere	
Commerce	347
Iniciar la publicación	348
Desempaquetado de los elementos del	
archivador de tienda	350
Actualización de los parámetros de publicación	351
Datos de publicación	352
Archivos de anotaciones cronológicas de la	
publicación	357
Disponibilidad del archivador de tienda en la	
Consola de administración	359
Registro del archivador de tienda en el archivo	
SARRegistry.xml	359
Copia del archivador de tienda en el directorio	
de archivador de tienda aplicable.	360

Capítulo 37. Visión general de la carga de datos de tienda	361
Descripción de la carga de datos en WebSphere	
Commerce	362
Mandatos de Loader Package para cargar datos	
de tienda	365
Mandatos de Loader package para transformar	
y extraer datos	384
Herramientas relacionadas con los mandatos de	
Loader package	393
Carga de los datos de la tienda	393
Utilización de los mandatos y scripts de Loader	
package	394
Ejemplos de resolución de identificadores	394
Ejemplo de carga de datos	402

Capítulo 38. Carga de grupos de elementos de base de datos de WebSphere Commerce	405
Grupos de elementos de base de datos	405
Secuencia de carga de elementos de base de	
datos	406
Carga de una tienda	407
Carga de grupos de elementos de base de datos	413

Capítulo 39. Publicación de cuentas de negocio y contratos	419
Publicación de cuentas de negocio y contratos	
utilizando la Consola de administración o la línea	
de mandatos	419
Publicación de cuentas de negocio y contratos	
utilizando mandatos	420
Publicación de elementos de cuenta de negocio	420
Publicación de elementos de contrato	420

Capítulo 40. Publicación de los elementos de escaparate y de los archivos de configuración de tiendas	423
Publicación de los elementos de escaparate y de los	
archivos de configuración de tienda utilizando la	
Consola de administración o la línea de mandatos	423
Publicación de elementos de escaparate y archivos	
de configuración de tienda copiándolos en	
WebSphere Commerce Server	423

Parte 11. Adición de características de WebSphere Commerce a la tienda 427

Capítulo 41. Adición del servicio de atención al cliente a la tienda	429
Descripción del servicio de atención al cliente en	
una tienda	430
Utilización del conjunto de marcos	431
Utilización del servicio de atención al cliente sin	
un conjunto de marcos	432
Definición del servicio de atención al cliente	435
Supervisión de los clientes que utilizan el	
servicio de atención al cliente	442
Envío de peticiones directamente a una cola de	
atención al cliente	447
Personalización de la atención al cliente	447
Adición del servicio de atención al cliente a la	
tienda	451
Parte 1: Instalación de requisitos previos	451
Parte 2: Copia de los archivos de integración del	
servicio de atención al cliente de la tienda de	
ejemplo	451
Parte 3: Adición de código para determinar qué	
página está examinando el cliente	453
Parte 4: Adición de un enlace al servicio de	
atención al cliente	453

Parte 5: Creación de una página de entrada que redirigirá la página del conjunto de marcos de atención al cliente 453

Capítulo 42. Adición de zonas de e-marketing a la tienda 455

Zona de e-marketing 455

 Bean e-MarketingSpot 459

Adición de una zona de e-marketing a las páginas de la tienda 459

Parte 12. Apéndices 461

Apéndice A. Descripción de UML . . . 463

Apéndice B. Creación de los datos . . . 465

Creación de datos para tiendas de ejemplo 465

Apéndice C. Grupos de elementos de base de datos 467

Dependencias de grupos de elementos de base de datos 467

Apéndice D. Avisos 473

Marcas registradas. 475

Antes de empezar

La publicación *IBM WebSphere Commerce, Guía para el desarrollo de tiendas* proporciona información acerca de la arquitectura de tiendas de WebSphere Commerce y el proceso de desarrollo de tiendas. En particular, proporciona detalles sobre los siguientes temas:

- Proceso de desarrollo de la tienda
- Modelos de negocio soportados por WebSphere Commerce
- Arquitectura de WebSphere Commerce
- Desarrollo del escaparate
- Desarrollo de los datos de la tienda
- Arquitectura de los datos de la tienda
- Modelo de información de los datos de la tienda
- Globalización de la tienda
- Adición de control de acceso a la tienda
- Empaquetado de la tienda
- Publicación de la tienda
- Adición de características de WebSphere Commerce a la tienda

Convenios y terminología utilizados en este manual

Este manual utiliza los siguientes convenios para resaltar el texto:

Negrita	Indica mandatos o controles de interfaz gráfica de usuario (GUI) como, por ejemplo, nombres de campos, iconos u opciones de menú.
Monoespaciado	Indica ejemplos de texto que se escriben exactamente como se muestra, además de nombres de archivos y nombres y vías de acceso a directorios.
<i>Cursiva</i>	Se utiliza para enfatizar palabras. La cursiva también le indica que debe sustituir los valores por los valores correctos para su sistema.



Este icono indica una sugerencia, esto es, información adicional que puede ayudarle a realizar una tarea.

Importante









Estas secciones resaltan la información que tiene una importancia especial.

Nota

Estas secciones resaltan la información significativa.



Indica información específica de WebSphere Commerce Business Edition.

	Indica información específica de WebSphere Commerce Professional Edition.
	Indica información específica a los programas que se ejecutan en AIX.
	Indica información específica a los programas que se ejecutan en OS/400.
	Indica información específica de WebSphere Commerce para Linux para xSeries, información específica de WebSphere Commerce para Linux para Eserver zSeries yS/390, información específica de WebSphere Commerce para Linux para Eserver iSeries e información que es específica de WebSphere Commerce para Linux para Eserver pSeries.
	Indica información específica a los programas que se ejecutan en el entorno operativo Solaris.
	Indica información específica a los programas que se ejecutan en Windows 2000.
	Indica información específica a DB2 Universal Database.
	Indica información específica de Oracle9i Database.

Variables utilizadas en este manual

Algunas de las variables clave de este manual son las siguientes:

modelo_negocio

El nombre del modelo de negocio de ejemplo con el que está trabajando (por ejemplo, Directo al consumidor, Directo a B2B).

nombre_célula

Las células son agrupaciones lógicas y arbitrarias de uno o varios nodos de una red distribuida WebSphere Application Server gestionadas mediante WebSphere Network Deployment Manager. En esta definición, un nodo es una sola entidad de WebSphere Application Server y las aplicaciones que se ejecutan en el mismo WebSphere Application Server. Esta variable representa el nombre de la célula de WebSphere Application Server.

nombre_sistema_principal

Esta variable representa el nombre de sistema principal totalmente calificado de WebSphere Commerce Server (por ejemplo, server.midominio.ibm.com está totalmente calificado).

nombre_instancia

Esta variable representa el nombre de la instancia de WebSphere Commerce con la que está trabajando (por ejemplo, mall1).

dir_tienda

Esta variable representa el nombre del directorio de tienda en el que está ubicada la tienda.

nombre_instancia_WAS

Esta variable representa el nombre del WebSphere Application Server con el que está asociado WebSphere Commerce.

Variables de vía de acceso

Esta guía utiliza las variables siguientes para representar las vías de acceso a directorios:

dir_instalación_WC

Este es el directorio de instalación para WebSphere Commerce. Los siguientes son los directorios de instalación por omisión para WebSphere Commerce en diferentes sistemas operativos.

▶ AIX	/usr/WebSphere/CommerceServer55
▶ 400	/QIBM/ProdData/CommerceServer55
▶ Linux	/opt/WebSphere/CommerceServer55
▶ Solaris	/opt/WebSphere/CommerceServer55
▶ Windows	C:\Archivos de programa\WebSphere\CommerceServer55

▶ 400 *dir_usuario_WC*

Este es el directorio para todos los datos que utiliza WebSphere Commerce que el usuario necesita configurar o puede modificar. Un ejemplo de este tipo de datos es la información de instancias de WebSphere Commerce. Este directorio es exclusivo de OS/400.

La variable *dir_usuario_WC* representa el directorio siguiente:

/QIBM/UserData/CommerceServer55

dir_instalación_WAS

Este es el directorio de instalación para WebSphere Application Server. Los siguientes son los directorios de instalación por omisión para WebSphere Application Server en diferentes sistemas operativos.

▶ AIX	/usr/WebSphere/AppServer
▶ 400	/QIBM/ProdData/WebAS5/Base
▶ Linux	/opt/WebSphere/AppServer
▶ Solaris	/opt/WebSphere/AppServer
▶ Windows	C:\Archivos de programa\WebSphere\AppServer

▶ 400 *dir_usuario_WAS*

Este es el directorio para todos los datos que utiliza WebSphere Application Server que el usuario necesita configurar o puede modificar. Un ejemplo de este tipo de datos es la información de instancias de WebSphere Application Server. Este directorio es exclusivo de OS/400.

La variable *dir_usuario_WAS* representa el directorio siguiente:

/QIBM/UserData/WebAS5/Base/*nombre_instancia_WAS*

dir_usuario_WC

La variable *dir_usuario_WC* representa el directorio siguiente:
/QIBM/UserData/WebAS5/Base/nombre_instancia_WAS

dir_espacio_trabajo

Se utiliza en el entorno de desarrollo. La variable representa
unidad:\WebSphere\espacio_trabajo_db2

Dónde encontrar la información más reciente

Es posible que esta publicación se actualice más adelante. Encontrará las actualizaciones en el siguiente sitio Web de WebSphere Commerce:

<http://www.ibm.com/software/commerce/library/>

Las actualizaciones pueden incluir información nueva.

Parte 1. Visión general

Capítulo 1. Visión general del desarrollo de tiendas

Este capítulo proporciona una visión general del proceso de desarrollo del sitio o de la tienda en WebSphere Commerce y describe muchos de los conceptos que se utilizan en esta guía.

Nota: Esta guía utiliza el término *desarrollo de tiendas* para hacer referencia a los procesos necesarios para crear una sola tienda y a los procesos necesarios para crear un entorno de varias tiendas o sitios.

Descripción del desarrollo de tiendas en WebSphere Commerce

Antes de comenzar a desarrollar su sitio o tienda con WebSphere Commerce, debe comprender cómo los factores siguientes afectan el proceso de desarrollo de tiendas. En este capítulo se describen brevemente todos estos factores pero, en la mayor parte de los casos, encontrará una descripción más detallada en otros apartados de esta guía y, en algunos casos, en otros documentos de la biblioteca de WebSphere Commerce.

El modo en que decida desarrollar la tienda en WebSphere Commerce dependerá de los factores siguientes:

- La finalidad de su tienda
- El modelo de negocio representativo de su tienda
- El número y el tipo de tiendas que se están desarrollando
- La base de la tienda
- El grado de personalización necesario

La finalidad de la tienda

Generalmente, las tiendas se desarrollan para una de las finalidades siguientes:

- **Producción:** Las tiendas de producción son tiendas totalmente funcionales en un entorno de producción, preparadas para que puedan utilizarlas clientes o asociados.
- **Demostración:** Las tiendas de demostración muestran determinadas posibilidades para fines de ventas. Es posible que las tiendas de demostración no tengan todas las funciones.
- **Ejemplo:** Las tiendas de ejemplo son tiendas totalmente funcionales que están diseñadas para utilizarlas como base para crear la tienda en línea.

El modelo de negocio representativo de su tienda

Antes de desarrollar la tienda, es necesario que comprenda cuál de los modelos de negocio soportados por WebSphere Commerce es el que mejor se ajusta a su tienda. WebSphere Commerce da soporte a sitios o tiendas que sean una instancia de los modelos de negocio siguientes:

- Nota:** Estos modelos de negocio se describen detalladamente en el Capítulo 2, "Modelos de negocio soportados en WebSphere Commerce", en la página 17, pero aquí se incluye una breve descripción del modelo de negocio.
- **Modelo de negocio de Ventas directas:** Al igual que en releases anteriores, WebSphere Commerce da soporte al modelo de negocio de ventas directas. Con

WebSphere Commerce puede crear sitios o tiendas que den soporte a transacciones comerciales relacionadas con productos, servicios o información directamente entre empresas y consumidores o entre dos empresas o partes. WebSphere Commerce da soporte a los siguientes tipos de modelos de negocio de venta directa:

- Modelo de negocio Directo al consumidor: soporta transacciones comerciales relacionadas con productos, servicios o información entre empresas y consumidores. Normalmente, los consumidores compran las mercancías o los servicios directamente a una empresa en un escenario directo al consumidor.
- **► Business** Modelo de negocio Directo a B2B: Directo a B2B da soporte a transacciones comerciales relacionadas con productos, servicios o información entre dos empresas o partes. Generalmente, las transacciones de tipo Directo a B2B se producen entre compradores, proveedores, fabricantes, revendedores, distribuidores y socios comerciales.
- **► Business** Modelo de negocio de Alojamiento: WebSphere Commerce también da soporte al alojamiento de comerciantes o de otros negocios mediante un ISP (Proveedor de servicios de Internet) u otro proveedor de alojamiento.
- **► Business** Modelo de negocio de Cadena de valor: Las cadenas de valor soportan transacciones en las que participan varias empresas o partes. Los productos, mercancías, servicios o información se entregan a través de las partes integrantes de la cadena de valor, desde los fabricantes a los usuarios finales. Una cadena de valor también tiene aspectos de relación y administrativos, es decir, puede gestionar la relación de los asociados o empresas de su cadena de valor, así como ofrecer ciertos servicios administrativos a esas partes. WebSphere Commerce soporta las transacciones y la gestión de relaciones de los dos tipos siguientes de cadenas de valor:
 - Cadenas de demanda: Las cadenas de demanda dan soporte a los canales de venta indirecta y a los canales de venta directa.
 - Cadenas de oferta: Las cadenas de oferta dan soporte a la compra y distribución y de productos y servicios. WebSphere Commerce da soporte a la distribución de artículos a través de los Marketplaces privados. Un Marketplace privado proporciona un foro para que los proveedores ofrezcan sus productos y servicios en venta a los compradores con los que tienen relaciones contractuales.

Nota: Estos modelos de negocio se describen detalladamente en el Capítulo 2, “Modelos de negocio soportados en WebSphere Commerce”, en la página 17.

El número de tiendas que se están desarrollando

Dependiendo de su negocio, es posible que tenga que desarrollar más de una tienda o más de un tipo de tienda. Por ejemplo, si su empresa vende directamente a los clientes, es posible que solamente necesite una tienda, a la que puedan acceder sus clientes para comprar artículos. No obstante, si está dando soporte a la cadena de demanda, es posible que necesite una tienda central principal para el negocio y varias tiendas que le permitan conectar con o administrar los canales. Es posible que también decida alojar tiendas para las organizaciones o empresas de sus canales. Para obtener más información sobre la cadena de demanda, consulte el Capítulo 2, “Modelos de negocio soportados en WebSphere Commerce”, en la página 17.

Si el suyo es un negocio de alojamiento de tiendas para comerciantes u otros negocios, también tendrá que desarrollar una tienda central para gestionar los

comerciantes y manejar las peticiones de registro y un método para desarrollar sitios para aquellos a los que está alojando. Para obtener más información sobre el modelo de negocio de alojamiento, consulte el Capítulo 2, “Modelos de negocio soportados en WebSphere Commerce”, en la página 17.

Con WebSphere Commerce puede desarrollar diversos tipos de tiendas y también varias tiendas para cada sitio. Para obtener más información sobre los tipos de tiendas, consulte el Capítulo 7, “Arquitectura de la tienda”, en la página 67.

La base de la tienda

Antes de crear un sitio o tienda con WebSphere Commerce, debe decidir dónde desea comenzar el desarrollo. WebSphere Commerce ofrece varios ejemplos que puede utilizar como punto de partida para el desarrollo, o puede empezar de cero. Para obtener más información acerca de los ejemplos que se proporcionan con WebSphere Commerce, consulte la publicación *WebSphere Commerce, Guía de tiendas de ejemplo*.

Comienzo a partir de un ejemplo

Los ejemplos que se proporcionan con WebSphere Commerce están empaquetados como archivadores de tienda.

El archivador de tienda: Un archivo archivador de tienda (.sar) es un archivo ZIP que contiene todos los elementos necesarios para crear un sitio o una tienda. Se utiliza principalmente como vehículo para empaquetar y entregar tiendas. Sólo tiene que publicar un archivador de tienda en WebSphere Commerce para crear una tienda en funcionamiento que se puede ver, visitar y en la que se puede comprar.

Normalmente, un archivador de tienda está compuesto de los archivos siguientes:

- Elementos Web: los archivos utilizados para crear las páginas de tienda como, por ejemplo, archivos HTML, archivos JSP, imágenes, gráficos y archivos de inclusión.
- Paquete de recursos de propiedades: Contiene el texto para las páginas de la tienda. Si la tienda da soporte a más de un idioma, el archivador de tienda contendrá varios paquetes de recursos, uno por idioma soportado, además de un paquete de recursos por omisión (que no incluye un entorno nacional). Por ejemplo, `AddressText_en_US.properties` y `AddressText.properties`.
- Elementos de datos de tienda: Los datos que se deben cargar en la base de datos. Los elementos de datos de tienda incluyen datos como las campañas, las entradas de catálogo, las monedas, la información de despacho de pedidos, de fijación de precios, de envío, de la tienda y de los impuestos. Para obtener una lista más detallada de los elementos de datos de tienda, consulte la Parte 6, “Desarrollo de los datos de la tienda”, en la página 117.

Los elementos de datos de tienda de los archivadores de tienda de ejemplo proporcionados con WebSphere Commerce tienen el formato de archivos XML válidos para Loader Package. Los archivos XML del archivador de tienda están pensados para ser portables y no deben contener claves primarias generadas que sean específicas de una instancia determinada de la base de datos. En lugar de ello, utilizan alias internos que ID Resolver resuelve en el momento en que se publica la tienda. El uso de estas convenciones permite la portabilidad de los archivadores de tienda de ejemplo. Para obtener más información, consulte la Parte 9, “Empaquetado de la tienda”, en la página 337.

Para obtener más información sobre Loader Package, consulte el Capítulo 37, “Visión general de la carga de datos de tienda”, en la página 361.

Nota: Los elementos de datos de tienda también incluyen información acerca de cómo crear un contrato. La información del contrato no se carga mediante Loader package; éste proporciona entrada para un mandato que crea los contratos.

- Elementos de pago: Información de configuración para WebSphere Commerce Payments. La información de pago no se carga mediante Loader package; éste proporciona entrada para un mandato que configura WebSphere Commerce Payments.
- Descriptores: Archivos XML que describen el archivador de tienda e información sobre cómo se debe publicar. Estos archivos son store-refs.xml, ibm-wc-load.xml, unpack.xml y ForeignKeys.dtd.

Para obtener más información sobre el archivador de tienda, consulte el Capítulo 35, "Empaquetado de una tienda", en la página 339.

Publicación de un archivador de tienda: Puede publicar un archivador de tienda utilizando el programa de utilidad de publicación de la Consola de administración, o mediante la línea de mandatos. Para obtener información detallada acerca de cómo publicar un archivador de tienda, consulte el tema "Publicación de un archivador de tienda" de la Ayuda en línea a la producción de WebSphere Commerce.

Tipos de ejemplos: Los ejemplos que se proporcionan con WebSphere Commerce se clasifican en las categorías siguientes:

- Archivadores de tienda compuestos
- Archivadores de tienda de componente
- Archivador de tienda básico

Archivadores de tienda compuestos: Un archivador de tienda compuesto contiene todos los elementos necesarios para crear un sitio operativo. Los archivadores de tienda compuestos de ejemplo que se proporcionan con WebSphere Commerce suelen contener la estructura de la organización, roles de usuario predefinidos y las políticas de control de acceso necesarias para crear el entorno adecuado para el modelo de negocio correspondiente. Los archivadores de tienda compuestos contienen también los elementos necesarios para crear las tiendas o sitios necesarios. Por ejemplo, el archivador de tienda compuesto del ejemplo de cadena de demanda contiene un sitio de centro de canal, un catálogo compartido y tiendas de revendedor y de distribuidor.

WebSphere Commerce incluye varios archivadores de tienda compuestos que contienen sitios de ejemplo en línea totalmente funcionales que puede utilizar como base para crear su propia tienda. Estos ejemplos, que incluyen tiendas de venta directa (tanto Directo al consumidor como Directo a B2B), un negocio de cadena de demanda, un negocio de proveedor y un sitio de alojamiento, implementan muchas de las características que se utilizan más comúnmente en los sitios de comercio electrónico actuales y proporcionan todos los elementos de tienda necesarios. Para obtener más información acerca de los ejemplos que se proporcionan con WebSphere Commerce, consulte la publicación *WebSphere Commerce, Guía de tiendas de ejemplo*.

Por qué comenzar con un archivador de tienda compuesto de ejemplo: Cuando comienza con un archivador de tienda compuesto de ejemplo, se cargan todos los datos necesarios en WebSphere Commerce Server para crear un sitio totalmente funcional.

WebSphere Commerce requiere que se carguen determinados datos en la base de datos de WebSphere Commerce Server para poder crear un sitio funcional y que estos datos se carguen en el orden que determine el esquema. Dado que los archivadores de tienda de componente de ejemplo incluyen todos los datos obligatorios en el orden y la estructura que requiere la base de datos de WebSphere Commerce Server, si utiliza uno como base para su propio sitio se ahorrará una cantidad de tiempo considerable durante el período de creación inicial.

Después de publicar un archivador de tienda compuesto de ejemplo, puede editarlo todo o un poco, dependiendo de las necesidades de la tienda. Por ejemplo, es posible que sólo tenga que editar los datos utilizando las herramientas disponibles con WebSphere Commerce y cambiar el aspecto visual y operacional de las páginas de tienda utilizando WebSphere Commerce Studio. O es posible que tenga que editar los archivos XML o la base de datos directamente para efectuar cambios de más envergadura en los datos, y volver a escribir las páginas de tienda para cambiar las características y el flujo de la tienda. O es posible que en su caso el método de desarrollo de tienda que mejor funcione sea utilizar la tienda de ejemplo y desarrollar elementos de tienda nuevos. Por ejemplo, si algunos de los elementos de base de datos de una de las tiendas de ejemplo coinciden exactamente con las necesidades de la tienda, pero el flujo de las páginas de dicha tienda no coincide, puede copiar los elementos de base de datos de la tienda y personalizarlos mientras desarrolla elementos Web totalmente nuevos. Para obtener más información acerca de cómo editar los datos de tienda, consulte la Parte 6, “Desarrollo de los datos de la tienda”, en la página 117.

Archivadores de tienda de componente: Cada una de las partes que forman un archivador de tienda compuesto también están disponibles como archivadores de tienda independientes. Estos archivadores de tienda se conocen como archivadores de tienda de componente. Un archivador de tienda de componente puede ser un archivador de tienda de estructura de organización, que contiene la estructura de la organización y roles de usuario predefinidos, o puede ser una tienda funcional o un conjunto de elementos de archivo o de datos que otros tipos de tiendas pueden utilizar como recursos. Para obtener más información acerca de los ejemplos que se proporcionan con WebSphere Commerce, consulte la publicación *WebSphere Commerce, Guía de tiendas de ejemplo*.

Por qué comenzar con un archivador de tienda de componente de ejemplo:

Comenzar con un archivador de tienda de componente de ejemplo, o con una combinación de archivadores de tienda de componente de ejemplo, le proporcionará más flexibilidad que comenzar con un archivador de tienda compuesto de ejemplo, ya que cuando se publica un archivador de tienda compuesto se crea un sitio con todas sus funciones. Algunas partes de este sitio pueden ajustarse a sus necesidades y otras no. Por ejemplo, si el flujo de sus páginas de tienda es muy diferente de cualquiera de los ejemplos proporcionados, o si piensa personalizar mucho el esquema de base de datos de WebSphere Commerce Server, tal vez decida publicar solamente determinadas partes de uno de los ejemplos proporcionados y no todo el ejemplo. Por ejemplo, es posible que desee publicar solamente la estructura de organización de ejemplo y luego desarrollar todos los elementos para crear las tiendas en su sitio. O es posible que desee publicar una estructura de organización de ejemplo y uno o más de los archivadores de componente de ejemplo que crean una tienda en la estructura de la organización, o proporcionan recursos que pueden utilizar otras tiendas.

Nota: Si va a crear una instancia de un modelo de negocio de cadena de valor, se le recomienda que comience por publicar la estructura de organización de ejemplo, ya que la estructura de organización necesaria para sitios que

contienen varias entidades es bastante compleja. Para obtener más información acerca de cómo funcionan las estructuras de organización en WebSphere Commerce, consulte el apartado “Descripción de la estructura organizativa de WebSphere Commerce” en la página 27.

Archivador de tienda básico: WebSphere Commerce proporciona también una tienda de ejemplo básica que ofrece el conjunto mínimo de elementos necesarios para crear una tienda en WebSphere Commerce Server.

Por qué comenzar con una tienda de ejemplo básica: Comenzar con la tienda de ejemplo básica le permite establecer una entidad de tienda en Commerce Server, en la que se pueden invocar los archivos JSP utilizando el ID de tienda. Si está creando una tienda que es muy diferente de cualquiera de las tiendas de ejemplo que se proporcionan con WebSphere Commerce, es posible que desee comenzar con la tienda de ejemplo básica, ya que esto permite al desarrollador añadir elementos según sea necesario sin tener que suprimir ni modificar los elementos que no son aplicables a su tienda. Para obtener más información acerca de la tienda básica proporcionada con WebSphere Commerce, consulte la publicación *WebSphere Commerce, Guía de tiendas de ejemplo*.

Nota: Puede utilizar la tienda de ejemplo básica junto con una de las estructuras de organización de ejemplo proporcionadas.

Comienzo a partir de cero

También es posible empezar de cero, es decir, no utilizar ninguno de los ejemplos que se proporcionan con WebSphere Commerce.

Grado de personalización necesario

Cuando haya decidido cuál va a ser la base para su tienda, ya sea una tienda de ejemplo, una estructura de organización de ejemplo, la tienda de ejemplo básica o si comienza de cero, tiene que determinar el tipo de cambios que realizará. En general, la mayor parte de las tareas de desarrollo de tiendas en WebSphere Commerce entran en una de las categorías siguientes:

- Añadir o cambiar las funciones de la tienda, incluido añadir características nuevas o cambiar el flujo de la tienda.
- Crear o cambiar el aspecto visual y operacional de una tienda
- Crear o cambiar los datos de tienda

En muchos casos, el trabajo de desarrollo de tiendas incluirá una combinación de las tres.

Adición o modificación de las funciones de la tienda

Para añadir o modificar las funciones de la tienda, incluido modificar el flujo de la tienda o añadir nuevas características a la tienda, suele ser necesario modificar la lógica de negocio. En la publicación *WebSphere Commerce, Guías de programación y aprendizaje* se describen las herramientas para desarrollar la lógica de negocio, incluyendo crear y ampliar mandatos, crear código personalizado e implementar la lógica de negocio.

Nota: Los desarrolladores que vayan a crear o modificar la lógica de negocio deberán tener conocimientos de programación en Java, Enterprise JavaBeans, WebSphere Studio Application Developer, J2EE y estar familiarizados con el con modelo de programación de WebSphere Commerce y el modelo de objeto.

WebSphere Commerce Accelerator permite modificar algunas de las características y los flujos de tienda proporcionados con las tiendas de ejemplo Directo al

consumidor y Directo a B2B. Para obtener más información acerca de los flujos y las funciones que puede modificar y cómo puede hacerlo, consulte la Ayuda en línea a la producción de WebSphere Commerce.

Creación o modificación del diseño de una tienda

Generalmente, crear o cambiar el diseño de una tienda supone cambiar el escaparate. Los elementos del escaparate incluyen elementos Web como, por ejemplo, las páginas HTML, los archivos JSP, las hojas de estilo, las imágenes, gráficos y otros tipos de archivos multimedia. Para desarrollar los elementos del escaparate es posible que sea necesario personalizar las páginas de la tienda de ejemplo, sustituirlas por páginas propias, crear páginas nuevas o una combinación de estas tres posibilidades.

WebSphere Commerce proporciona las siguientes herramientas para crear o editar elementos de escaparate:

- **WebSphere Studio Application Developer**

WebSphere Studio Application Developer (incluido en WebSphere Commerce Studio) proporciona las herramientas necesarias para crear y editar los elementos de escaparate, incluidos archivos HTML, de gráficos, multimedia y archivos JSP (JavaServer Pages). Page Designer, incluido en WebSphere Studio Application Developer, permite crear archivos HTML o JSP como imágenes animadas. También puede configurar WebSphere Studio Application Developer para que utilice otra herramienta de desarrollo Web que desee. Consulte la ayuda en línea de WebSphere Studio para obtener más información acerca de cómo registrar sus propias herramientas.

Para obtener más información acerca de cómo utilizar las herramientas en WebSphere Commerce Studio para crear y editar los elementos de escaparate, consulte la ayuda en línea de WebSphere Commerce Studio. Para obtener más información acerca de cómo crear el escaparate en WebSphere Commerce, consulte el Capítulo 8, “Desarrollo del escaparate”, en la página 81.

Nota: Los desarrolladores que vayan a crear o modificar el escaparate deberán tener conocimientos de programación en tecnología Java, JavaScript, HTML y JSP, y estar familiarizados con la arquitectura de tiendas de WebSphere Commerce.

- **WebSphere Commerce Accelerator**

WebSphere Commerce Accelerator incluye las herramientas siguientes para modificar el diseño de la tienda.

- Cuaderno Cambiar páginas
- Cuaderno Subir el logotipo
- Asistente para Cambiar estilo
- Cuaderno Gestionar archivos
- Cuaderno Perfil de tienda

Para obtener más información sobre estas herramientas, consulte la Ayuda en línea a la producción de WebSphere Commerce.

Nota: Las herramientas listadas anteriormente sólo funcionan con las tiendas basadas en la tienda de ejemplo Directo al consumidor y con las tiendas alojadas (del modelo de alojamiento y del modelo de cadena de demanda) que se hayan creado con el Asistente de creación de tiendas.

Creación o modificación de los datos de la tienda

Tiene varias opciones para desarrollar y editar los elementos de base de datos de la tienda.

- Loader Package de WebSphere Commerce

Loader Package de WebSphere Commerce consta principalmente de programas de utilidad para preparar y cargar datos en una base de datos de WebSphere Commerce. Puede utilizar Loader Package para cargar grandes cantidades de datos y actualizar datos de la base de datos de WebSphere Commerce. El programa de utilidad Loader de este paquete utiliza XML válido y con formato correcto como entrada para cargar datos en la base de datos. Los elementos del documento XML se correlacionan con los nombres de tabla de la base de datos y los atributos de elementos se correlacionan con las columnas.

Para obtener información acerca de cómo utilizar Loader Package para desarrollar y cargar elementos de datos, consulte el Capítulo 37, “Visión general de la carga de datos de tienda”, en la página 361.

Cuándo se ha de utilizar Loader Package de WebSphere Commerce: Utilice Loader Package de WebSphere Commerce para cargar inicialmente los elementos de base de datos en la base de datos de WebSphere Commerce y para actualizarlos. También se puede utilizar Loader package para automatizar la entrada de datos regular desde un sistema ya existente.

- WebSphere Commerce Accelerator

WebSphere Commerce Accelerator es un conjunto de herramientas en línea que se utiliza principalmente para el mantenimiento de tiendas en línea mediante diversas operaciones de tienda. No obstante, dado que WebSphere Commerce Accelerator le permite crear o editar datos, puede utilizarlo también como una herramienta de desarrollo de tiendas, en especial cuando vaya a modificar pequeñas cantidades de datos. Para obtener una lista de los elementos de base de datos que puede editar con WebSphere Commerce Accelerator consulte la Parte 6, “Desarrollo de los datos de la tienda”, en la página 117.

Cuándo se ha de utilizar WebSphere Commerce Accelerator: Utilice WebSphere Commerce Accelerator cuando vaya a crear o actualizar datos.

- Editar la base de datos directamente

También tendrá siempre la opción de editar la base de datos directamente utilizando inserciones, actualizaciones o supresiones SQL.

Nota: SQL es un lenguaje específico de base de datos. Es posible que para Oracle se requiera una sintaxis de SQL diferente. Tenga en cuenta que las sentencias SQL tendrán necesariamente valores específicos de base de datos y que las sentencias SQL quizá no puedan volverse a utilizar en otra instancia de WebSphere Commerce Server.

Los desarrolladores de tiendas que vayan a crear o modificar los datos de tienda deberán estar familiarizados con la arquitectura de tiendas de WebSphere Commerce, los datos de tienda y los archivadores de tienda. Para modificar y ampliar el esquema de base de datos de WebSphere Commerce a fin de implementar las funciones de tienda personalizadas, o de integrarlas con la información de la base de datos existente, el desarrollador deberá tener experiencia como administrador de bases de datos DB2 u Oracle.

Escenario: Desarrollo y despliegue de una tienda de producción

En este apartado se describe el escenario recomendado para desarrollar una tienda de producción con WebSphere Commerce.

Tabla 1. Escenario: Desarrollo y despliegue de una tienda de producción

<i>Tarea</i>	<i>Subtareas</i>	<i>Consulte</i>
Determinar qué modelo de negocio soportado se asemeja a su negocio		Capítulo 2, “Modelos de negocio soportados en WebSphere Commerce”, en la página 17
Determinar el flujo de la tienda		Parte 4, “Desarrollo del escaparate”, en la página 79
Crear ejemplos de uso		Parte 4, “Desarrollo del escaparate”, en la página 79
Analizar las tiendas de ejemplo que se proporcionan con WebSphere Commerce		<i>WebSphere Commerce, Guía de tiendas de ejemplo</i>
Determinar qué tienda de ejemplo o qué otro ejemplo se ha de utilizar como punto de partida		<i>WebSphere Commerce, Guía de tiendas de ejemplo</i>
Crear un conjunto de elementos de tienda básicos	Crear un proyecto en el entorno de desarrollo (específico del entorno de desarrollo de WebSphere Commerce Studio)	Documentación del producto WebSphere Studio
	Publicar uno de los archivadores de tienda de ejemplo en el entorno de desarrollo	La Ayuda en línea a la producción de WebSphere Commerce y el tema de ayuda “Publicación de un archivador de tienda”
	Si es posible, configure la tienda utilizando la herramienta Cambiar flujo de WebSphere Commerce Accelerator	La Ayuda en línea a la producción de WebSphere Commerce y el tema de ayuda “Cambio de los flujos de tienda mediante WebSphere Commerce Accelerator”
	Efectuar los cambios necesarios en el esquema de base de datos	<i>WebSphere Commerce, Guías de programación y aprendizaje</i>
	Comprobar los elementos de tienda en un sistema de control de origen para crear una copia maestra	Manual técnico <i>WebSphere Commerce V5.5 Customization and Deployment Handbook</i> SG24-6969.

Tabla 1. Escenario: Desarrollo y despliegue de una tienda de producción (continuación)

Determinar las tareas de desarrollo necesarias para crear la tienda a partir de los elementos básicos (escaparate, datos y servidor de desarrollo)	Determinar los cambios del diseño de la tienda	<i>WebSphere Commerce, Guía de tiendas de ejemplo</i> Parte 4, “Desarrollo del escaparate”, en la página 79
	Determinar la estrategia de almacenamiento en antememoria para las páginas de la tienda	Parte 4, “Desarrollo del escaparate”, en la página 79
	Determinar los cambios en los datos de la tienda	<i>WebSphere Commerce, Guía de tiendas de ejemplo</i> Parte 6, “Desarrollo de los datos de la tienda”, en la página 117
	Comprender la implementación que utiliza la tienda de ejemplo	<i>WebSphere Commerce, Guía de tiendas de ejemplo</i>
	Analizar las funciones del servidor existentes para determinar dónde necesita mejoras o personalización	Ayuda en línea a la producción y el desarrollo de WebSphere Commerce
	Determinar el grado de integración necesario con los sistemas ya existentes	Ayuda en línea a la producción y el desarrollo de WebSphere Commerce
Establecer el entorno de equipo	Los desarrolladores configuran un proyecto de desarrollo en IDE y proporcionan al proyecto los elementos de la copia maestra del sistema de control de origen y los elementos de servidor que se entregan con el producto	Manual técnico <i>WebSphere Commerce V5.5 Customization and Deployment Handbook</i> SG24-6969.
	El desarrollador inicia la ejecución del conjunto básico de elementos	
	El equipo se familiariza con las funciones existentes en la tienda desde el punto de vista del cliente y del administrador	<i>WebSphere Commerce, Guía de tiendas de ejemplo</i>
Desarrollar los elementos de tienda	Modificar y ampliar los elementos del escaparate o crear nuevos elementos de escaparate	Parte 4, “Desarrollo del escaparate”, en la página 79
	Desarrollar funciones de servidor adicionales (escribiendo nuevos mandatos, EJB e integrándolas con los sistemas existentes)	<i>WebSphere Commerce, Guías de programación y aprendizaje</i>
	Modificar los datos y crear datos adicionales	Parte 6, “Desarrollo de los datos de la tienda”, en la página 117

Tabla 1. Escenario: Desarrollo y despliegue de una tienda de producción (continuación)

<p>Crear datos listos para producción</p>		<p>Parte 6, "Desarrollo de los datos de la tienda", en la página 117</p>
<p>Desplegar los elementos desarrollados en la producción</p>		<p><i>WebSphere Commerce, Guías de programación y aprendizaje</i> Puede encontrar información adicional en la publicación <i>WebSphere Commerce V5.5 Customization and Deployment Handbook</i>, SG24-6969 Redbook.</p>

Parte 2. Modelos de negocio soportados por WebSphere Commerce

Capítulo 2. Modelos de negocio soportados en WebSphere Commerce

Antes de comenzar a desarrollar su sitio o tienda con WebSphere Commerce, debe entender a qué modelos de negocio da soporte WebSphere Commerce. La mayoría de las tiendas creadas con WebSphere Commerce serán una instancia de uno de estos modelos de negocio.

Nota: También puede crear tiendas con WebSphere Commerce que no se ajusten a los modelos de negocio descritos en este capítulo.

Descripción de los modelos de negocio soportados en WebSphere Commerce

WebSphere Commerce proporciona la infraestructura de arquitectura para poner en línea negocios que se ajusten a uno de los modelos de negocio siguientes:

- Venta directa
- Alojamiento
- Cadena de valor

Venta directa

La Venta directa soporta transacciones comerciales relacionadas con productos, servicios o información directamente entre empresas y consumidores o entre dos empresas o partes. WebSphere Commerce da soporte a los siguientes tipos de modelos de negocio de venta directa:

- Directo al consumidor
- Directo a B2B

Directo al consumidor

Directo al consumidor soporta transacciones comerciales relacionadas con productos, servicios o información entre empresas y consumidores. Normalmente, los consumidores compran las mercancías o los servicios directamente a una empresa en un escenario directo al consumidor.

El siguiente diagrama ilustra un negocio directo al consumidor típico.



En un negocio directo al consumidor típico, los clientes compran directamente al negocio, que generalmente es un minorista, como se muestra en este diagrama. El negocio puede ser un minorista, un fabricante que vende sus artículos directamente a los consumidores mediante su propia tienda minorista o cualquier otro negocio que venda productos o proporcione servicios directamente al consumidor. Por ejemplo, un negocio que vende directamente a los consumidores mediante un catálogo se puede considerar un negocio directo al consumidor.

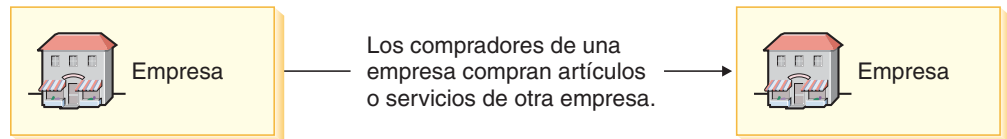
Las organizaciones que tradicionalmente no se consideran negocios, como por ejemplo los departamentos gubernamentales, también se pueden considerar

negocios Directo al consumidor. Los departamentos gubernamentales pueden proporcionar productos y servicios directamente a los clientes.

Directo a B2B

Directo a B2B soporta transacciones comerciales relacionadas con productos, servicios o información entre dos empresas o partes. Generalmente, las transacciones de tipo Directo a B2B se producen entre compradores, proveedores, fabricantes, revendedores, distribuidores y socios comerciales.

El diagrama siguiente ilustra un negocio directo a B2B típico.



En un negocio Directo a B2B típico, las empresas compran productos o servicios directamente a otra empresa. La empresa vendedora puede ser un mayorista, un distribuidor, un fabricante o un minorista que vende a compradores de otras empresas.

Las organizaciones que tradicionalmente no se consideran negocios, como por ejemplo, los departamentos gubernamentales y los medios de comunicación, también se pueden considerar negocios Directo a B2B. Los departamentos gubernamentales pueden proporcionar productos y servicios directamente a las empresas.

Alojamiento

El modelo de Alojamiento da soporte al alojamiento de comerciantes o de otros negocios mediante un ISP (Proveedor de servicios de Internet) u otro proveedor de alojamiento.

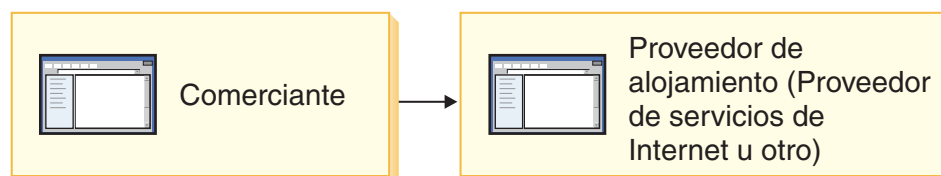
Los negocios de alojamiento pueden ser de dos tipos:

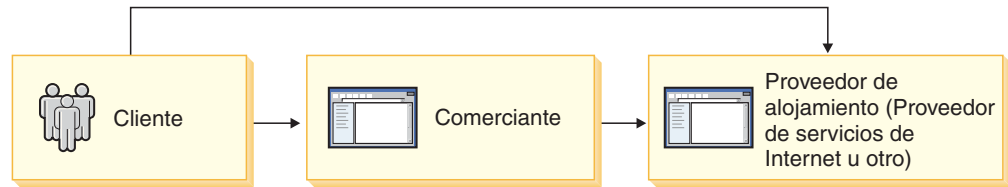
- tiendas alojadas
- (opcional) un sitio que permite a los clientes localizar las tiendas alojadas por el proveedor

Para gestionar las relaciones con las tiendas alojadas, los modelos de alojamiento incluyen normalmente un Centro (conocido en WebSphere Commerce como una tienda de centro.) Este Centro proporciona herramientas de autoservicio que permiten al comerciante crear y administrar una tienda, así como herramientas que permiten al proveedor de alojamiento gestionar todas las tiendas alojadas.

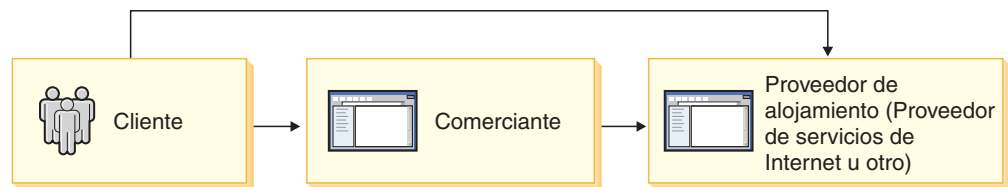
Los proveedores de alojamiento también suelen incluir una tienda en la que los clientes pueden encontrar y acceder a las tiendas que aloja el proveedor.

El diagrama siguiente muestra un ejemplo de alojamiento.





En este ejemplo, el comerciante entra en el sitio del proveedor de alojamiento y crea una tienda que se alojará en el sitio. Los proveedores de alojamiento suelen proporcionar a los comerciantes herramientas de autoservicio sencillas que les permiten administrar una tienda alojada. Cuando una tienda alojada se abre al público, los clientes pueden acceder a la tienda a través del sitio del proveedor de alojamiento o entrando directamente en la tienda alojada.



En este ejemplo, el cliente puede entrar en la tienda o negocio alojado directamente o puede navegar por el sitio del proveedor de alojamiento y luego se le puede transferir a la tienda o negocio alojado.

Las tiendas alojadas son muy similares a las tiendas Directo al consumidor. Para obtener información sobre las diferencias específicas entre las dos, como se implementan en las tiendas de ejemplo de WebSphere Commerce, consulte la publicación *WebSphere Commerce, Guía de tiendas de ejemplo*.

Cadena de valor

Una novedad de WebSphere Commerce Versión 5.5 es la posibilidad de permitir transacciones comerciales en línea en las que participan varias empresas. Las cadenas de valor soportan transacciones en las que participan varias empresas o partes. Los productos, mercancías, servicios o información se entregan a través de las partes integrantes de la cadena de valor, desde los fabricantes a los usuarios finales. Una cadena de valor también tiene aspectos de relación y administrativos, es decir, puede gestionar la relación de los asociados o empresas de su cadena de valor, así como ofrecer ciertos servicios administrativos a esas partes.

Como resultado las cadenas de valor deben gestionar los dos extremos de sus negocios: sus clientes y ventas directas y sus asociados del canal y proveedores. Cada uno de estos extremos requiere sus propios canales y métodos de gestión.

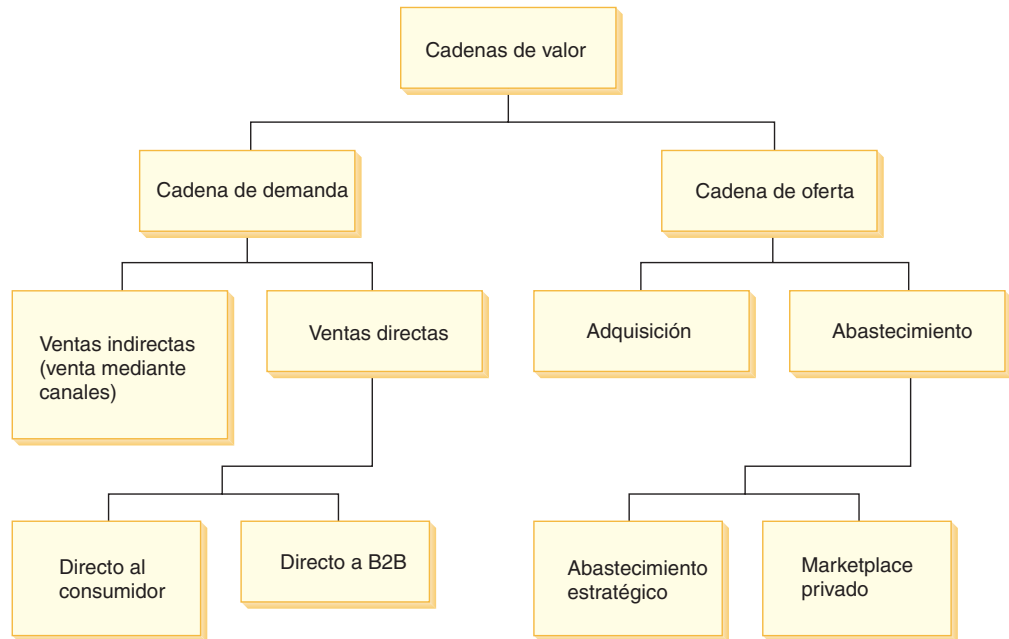
Para gestionar las relaciones con sus asociados o proveedores, los modelos de negocio de cadena de valor suelen incluir un Centro (conocido en WebSphere Commerce como una tienda de centro). Los administradores de la cadena de valor pueden administrar los aspectos operativos de la cadena de valor en la tienda de centro, incluido permitir que los asociados o proveedores puedan participar en la cadena de valor, es decir, registrarlos, configurarlos, crear colaboraciones. Los asociados y proveedores también puede acceder a la tienda de centro para realizar tareas administrativas como, por ejemplo, registrar usuarios.

Para poder vender directamente a los clientes (ventas directas), las cadenas de valor suelen incluir un escaparate, en el que los clientes pueden adquirir sus productos o servicios directamente.

WebSphere Commerce soporta las transacciones y la gestión de relaciones de estos dos tipos de cadenas de valor:

- Cadena de demanda
- Cadena de oferta

El diagrama siguiente proporciona una visión general de los asociados y las relaciones a los que se da soporte en las cadenas de valor.



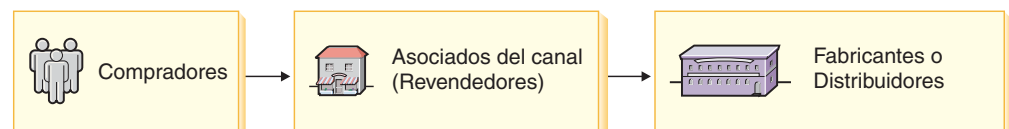
Cadena de demanda

Una cadena de demanda se compone de las empresas que venden los productos o servicios de un negocio. Por ejemplo, una cadena de demanda puede estar formada por los compradores que inician la transacción de ventas, los revendedores que venden los productos del fabricante y el fabricante que crea los productos. O una cadena de demanda puede estar compuesta por los revendedores que venden los productos de un fabricante, el fabricante que fabrica los productos y los distribuidores que suministran los productos del fabricante a los revendedores. Las cadenas de demanda también dan soporte a los canales de venta indirecta, en los que es el propietario de la cadena de demanda es quien vende directamente a los clientes o asociados. Para obtener más información sobre la venta directa, consulte el apartado “Venta directa” en la página 17.

Alojamiento de la cadena de demanda: El propietario de la cadena de demanda puede alojar tiendas para sus asociados del canal, por ejemplo revendedores o distribuidores.

Los diagramas siguientes ilustran ejemplos de algunas de las cadenas de demanda a las que da soporte WebSphere Commerce.

Compradores, asociados del canal (revendedores) y fabricantes:



En este ejemplo, los compradores adquieren productos a los revendedores de un fabricante (asociados del canal). Los revendedores, a su vez, obtienen los productos del fabricante, a través del Centro del fabricante.

Nota: Los revendedores pueden estar alojados por el fabricante o pueden ser remotos.

Revendedores, fabricantes y distribuidores:



En este ejemplo, el fabricante proporciona un centro para los asociados del canal, incluidos los revendedores. Los revendedores y los demás asociados del canal pueden realizar varias funciones en este centro, incluido localizar a los distribuidores de los productos del fabricante.

Para localizar los proveedores, el revendedor puede examinar un catálogo de productos del centro privado. Si los productos deseados están disponibles en más de un distribuidor, el revendedor puede consultar la disponibilidad de los productos, la ubicación de los distribuidores y los precios de diferentes distribuidores. A continuación, si el revendedor lo desea, puede dividir el pedido entre varios distribuidores. El pedido se envía entonces al distribuidor, que completa la transacción y entrega los artículos o servicios al revendedor. El revendedor vende posteriormente los artículos o servicios directamente al consumidor.

El sitio de ejemplo de cadena de demanda, Commerce Plaza, es un ejemplo de este escenario de revendedor, fabricante y distribuidor.

Nota: Los revendedores pueden estar alojados por el fabricante o pueden ser remotos.

Otros escenarios: Los ejemplos que se describen en este apartado no son más que algunas instancias de cadenas de demanda. Los detalles del escenario pueden variar según el tipo de negocio que dirija. Por ejemplo, si la empresa es un fabricante, la finalidad del centro puede ser la de ayudar a los revendedores del fabricante a localizar los artículos del fabricante de diferentes distribuidores. Si la empresa es un distribuidor, la finalidad del centro puede ser la de ayudar a los revendedores del distribuidor a encontrar artículos o servicios de diferentes proveedores.

Cadena de oferta

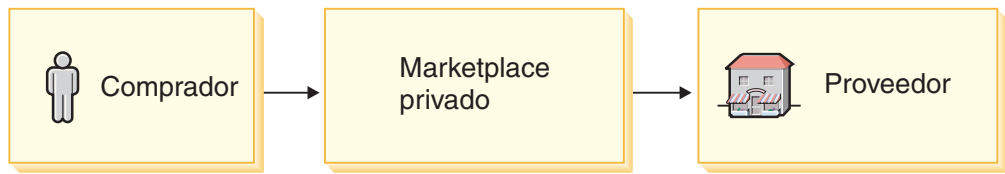
Una cadena de oferta se compone de las empresas que proporcionan servicios a un negocio. WebSphere Commerce proporciona la infraestructura de arquitectura para dar soporte a cadenas de oferta que adoptan la forma de un marketplace privado.

Un marketplace privado proporciona un foro para que los vendedores ofrezcan sus mercancías en venta. Los compradores entran en este foro y después de navegar por las opciones disponibles, seleccionan los artículos o servicios correspondientes.

Nota: El marketplace privado no da soporte a ofertas ni contraofertas competitivas ni a otros métodos competitivos.

Alojamiento de la cadena de oferta: El propietario de la cadena de oferta puede alojar una tienda para sus proveedores.

El diagrama siguiente muestra un ejemplo de negocio de proveedor.



En esta cadena de oferta de ejemplo, el comprador entra en el Centro del proveedor para interactuar y navega por el catálogo colectivo en el que se presentan productos y ofertas de varios proveedores. A continuación, el comprador puede seleccionar la oferta deseada o solicitar presupuestos de varios proveedores. El comprador también tiene la opción de realizar transacciones o compras directamente de proveedores en línea.

Tiendas de ejemplo en WebSphere Commerce

WebSphere Commerce proporciona varias tiendas de ejemplo que puede utilizar para familiarizarse con el modo en que WebSphere Commerce da soporte a los diferentes modelos de negocio que figuran en este capítulo. Los ejemplos disponibles (y los archivos archivadores de tienda correspondientes) son los siguientes:

Tabla 2.

Directo al consumidor	Directo a B2B	Alojamiento	Cadena de demanda	Cadena de oferta
FashionFlow (ConsumerDirect.sar)	ToolTech (B2BDirect.sar)	Nota: Todas las tiendas que se listan a continuación también están disponibles en el archivador de tienda compuesto Hosting.sar. Se le recomienda que publique Hosting.sar para ver el ejemplo de alojamiento completo.	Nota: Todas las tiendas que se listan a continuación también están disponibles en el archivador de tienda compuesto DemandChain.sar. Se le recomienda que publique DemandChain.sar para ver el ejemplo de cadena de demanda completo.	Nota: Todas las tiendas que se listan a continuación también están disponibles en el archivador de tienda compuesto SupplyChain.sar. Se le recomienda que publique SupplyChain.sar para ver el ejemplo de cadena de oferta completo.
		Centro de alojamiento de Commerce (Hosting Hub.sar)	Commerce Plaza (ChannelHub.sar)	Centro de proveedores de Commerce (SupplierHub.sar)
		Directorio de tiendas (Store Directory.sar)	Tienda con elementos de catálogo (CatalogAsset Store.sar)	Tienda con elementos de catálogo (CatalogAsset Store.sar)

Tabla 2. (continuación)

		Tienda con elementos de catálogo (CatalogAssetStore.sar)	Tienda con elementos de escaparate de revendedor (ResellerstorefrontAssetStore.sar)	Tienda con elementos de proveedor (SupplierAssetStore.sar)
		Tienda con elementos de escaparate alojada (HostedStoreFrontAssetStore.sar)	Tienda con elementos de distribuidor (DistributorAssetStore.sar)	Proveedores
		Tiendas alojadas	Tiendas de revendedor alojadas	
			Tiendas de distribuidor	

Para obtener más información acerca de los tipos de tiendas de estos ejemplos, consulte el apartado “Descripción del soporte de modelos de negocio en la arquitectura de tienda” en la página 70. Para obtener información detallada sobre las tiendas de ejemplo, consulte la publicación *WebSphere Commerce, Guía de tiendas de ejemplo*.

Nota: Cada ejemplo contiene también un archivador de tienda de componente que contiene la estructura de la organización para el modelo de negocio.

Tenga en cuenta que estos ejemplos son representativos de una instancia específica de tienda de cada modelo de negocio y no pretenden demostrar todas las posibles variaciones disponibles en el modelo de negocio. No obstante, incluso si su instancia específica del negocio es muy diferente del ejemplo proporcionado, es posible que pueda utilizar los ejemplos como un punto de partida para su propio sitio, o utilizar partes del mismo para crear el sitio. Para obtener información detallada acerca de los ejemplos que se proporcionan con WebSphere Commerce, consulte la publicación *WebSphere Commerce, Guía de tiendas de ejemplo*.

Parte 3. Arquitectura de WebSphere Commerce

En este apartado se proporciona una visión general de cómo la arquitectura de WebSphere Commerce permite abrir negocios en línea. En especial, este apartado describe cómo los componentes de la arquitectura de WebSphere Commerce permiten que las diferentes partes (por ejemplo, los clientes, los business partners o los distribuidores, revendedores y proveedores) del negocio puedan interactuar en línea.

Para permitir que las diferentes partes (por ejemplo, los clientes, business partners, proveedores, fabricantes, distribuidores y administradores) que colaboran con su negocio puedan interactuar en línea con su negocio y entre sí, la arquitectura de WebSphere Commerce incluye los siguientes componentes:

- Estructura de organización
- Modelo de control de acceso
- Infraestructura de política de negocio
- Arquitectura de instancia
- Arquitectura de tienda

Estos componentes juntos crean la arquitectura que permite que los diferentes asociados del negocio puedan interactuar entre sí.

Capítulo 3. Estructura organizativa de WebSphere Commerce

Para que los clientes o compradores puedan acceder al sitio, examinar el catálogo y formalizar pedidos; o para que los empleados puedan llevar a cabo la administración del sitio, incluyendo la actualización del catálogo, la creación de nuevas promociones o la gestión de pedidos; o para que los revendedores u otros business partners puedan realizar transacciones en el sitio, todos los participantes en el escenario de negocio deben tener asignada una posición en la estructura organizativa de WebSphere Commerce.

Descripción de la estructura organizativa de WebSphere Commerce

La estructura organizativa de WebSphere Commerce proporciona un marco de trabajo para las personas, o entidades, que participan en el escenario de negocio. Este marco de trabajo está organizado en una estructura jerárquica, que imita las jerarquías organizativas típicas, con entradas para organizaciones, unidades de organización y usuarios. Las organizaciones y las unidades de organización del marco de trabajo actúan como propietarios de las partes del negocio. Todas las partes del negocio, incluyendo clientes, administradores, tiendas, catálogos y distribuidores, deben ser propiedad de una organización o de una unidad de organización.

La estructura organizativa y el modelo de control de acceso, que se describen en el Capítulo 4, "Control de acceso en WebSphere Commerce", en la página 37, están estrechamente relacionados, en cuanto que el modelo de control de acceso aplica las políticas de control de acceso a las organizaciones en lugar de a las entidades individuales (tiendas, clientes, administradores, etcétera). Las políticas aplicables a una entidad (o recurso) se aplican a las organizaciones propietarias de la entidad o recurso.

El diagrama siguiente describe la estructura de organización básica de WebSphere Commerce. La estructura de organización básica se instala durante la creación de

la instancia, independientemente del modelo de negocio.



- **Organización raíz:** La organización raíz es la organización de nivel superior y es su propio padre. Todas las organizaciones de la estructura organizativa de WebSphere Commerce son descendientes de la organización raíz. La organización raíz es propietaria de los administradores de sitio.
- **Organización por omisión:** La organización raíz es la propietaria de la organización por omisión. Todos los clientes invitados y todos los clientes de un escenario Directo al consumidor pertenecen a la organización por omisión. Los clientes de un escenario Directo a B2B y de cadena de valor pueden pertenecer a la organización por omisión o a otras organizaciones.

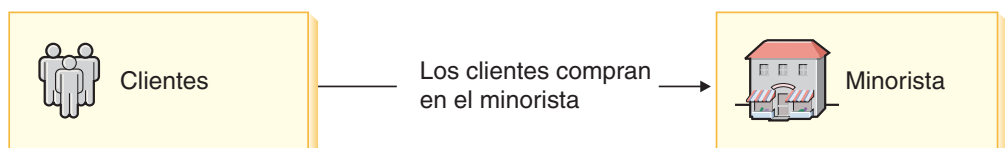
Pueden existir uno o más niveles diferentes de entidades de organización por debajo de las entidades de organización padre. Puede añadir tantas entidades de organización hijo como sean necesarias para dar soporte a su negocio.

Cómo la estructura organizativa da soporte a los modelos de negocio

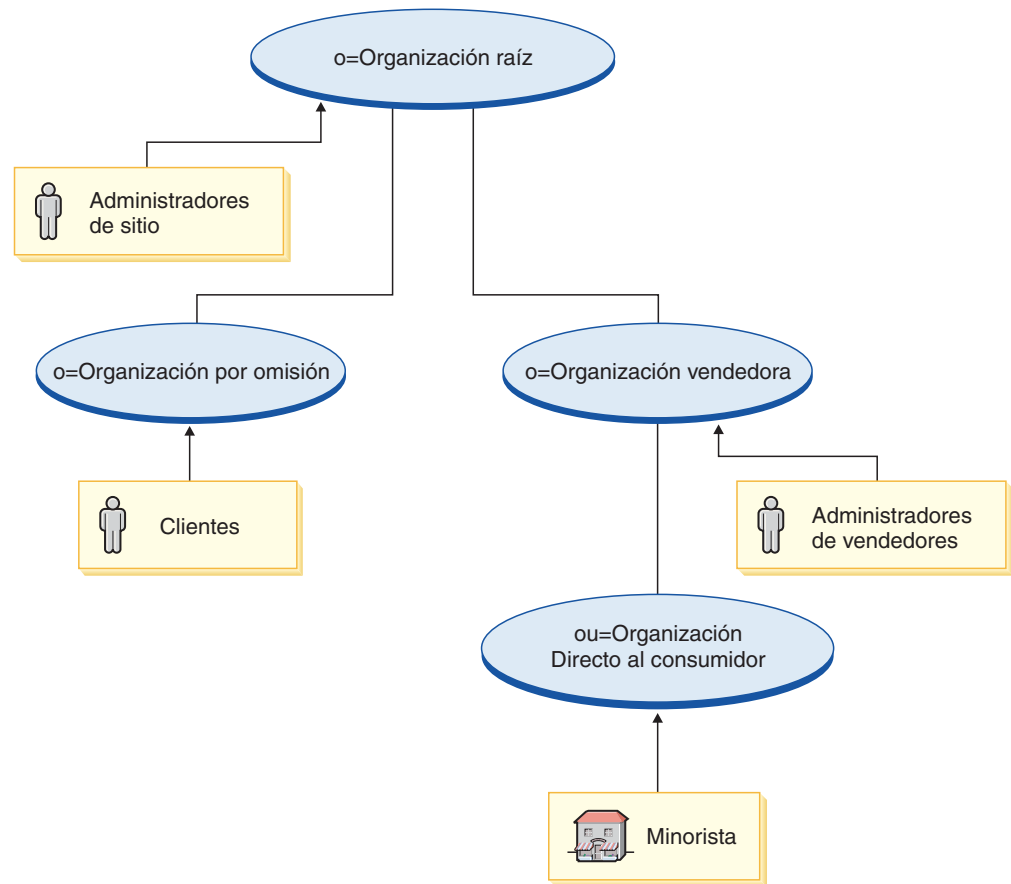
La estructura organizativa de WebSphere Commerce es lo suficientemente flexible como para dar soporte a todas las entidades de los modelos de negocio soportados. Los diagramas de los apartados siguientes muestran un ejemplo típico de cómo puede correlacionarse cada modelo de negocio con la estructura organizativa de WebSphere Commerce.

Directo al consumidor

El siguiente diagrama ilustra un negocio directo al consumidor típico.



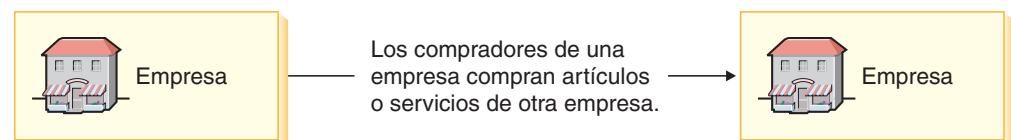
Para poner este negocio en línea con WebSphere Commerce, las entidades del diagrama anterior deben asignarse a las organizaciones siguientes:



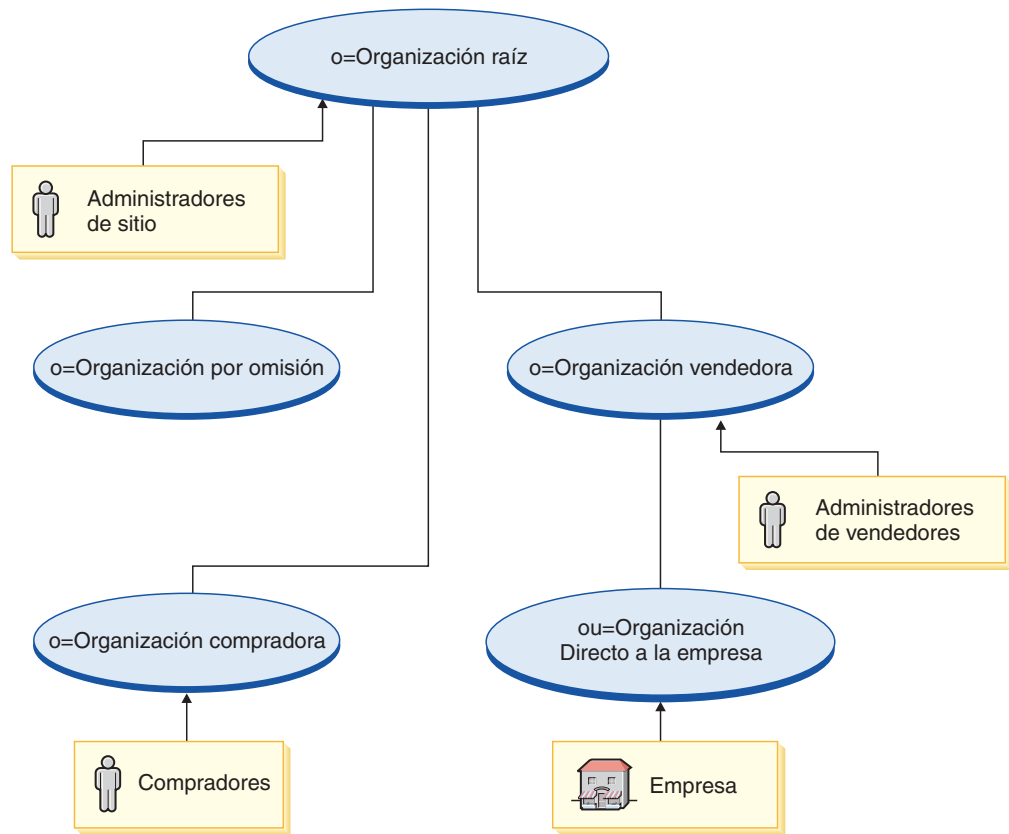
- **Organización raíz:** Todas las organizaciones del negocio se convierten en descendientes de la organización raíz. Los administradores de sitio, que mantienen el sitio en línea, son propiedad de la organización raíz.
 - **Organización por omisión:** Todos los clientes del negocio son propiedad de la organización por omisión.
 - **Organización vendedora:** Se crea una organización vendedora para que sea la propietaria de todas las organizaciones vendedoras (incluidas las tiendas y los administradores que mantienen las tiendas). Los administradores que mantienen las funciones de la tienda (por ejemplo, los representantes de servicio al cliente, los jefes de producto y los gestores de catálogos) se denominan administradores de vendedores y son propiedad de la organización vendedora.
 - Bajo la organización vendedora se crea una unidad de organización hijo (ou), organización directo al consumidor, para que sea propietaria de la tienda (Minorista).

Directo a B2B

El diagrama siguiente ilustra un negocio directo a B2B típico.



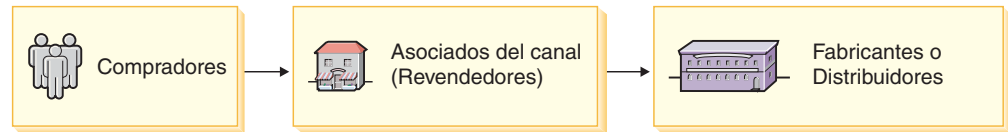
Para poner este negocio en línea, las entidades del diagrama anterior deben asignarse a las organizaciones siguientes:



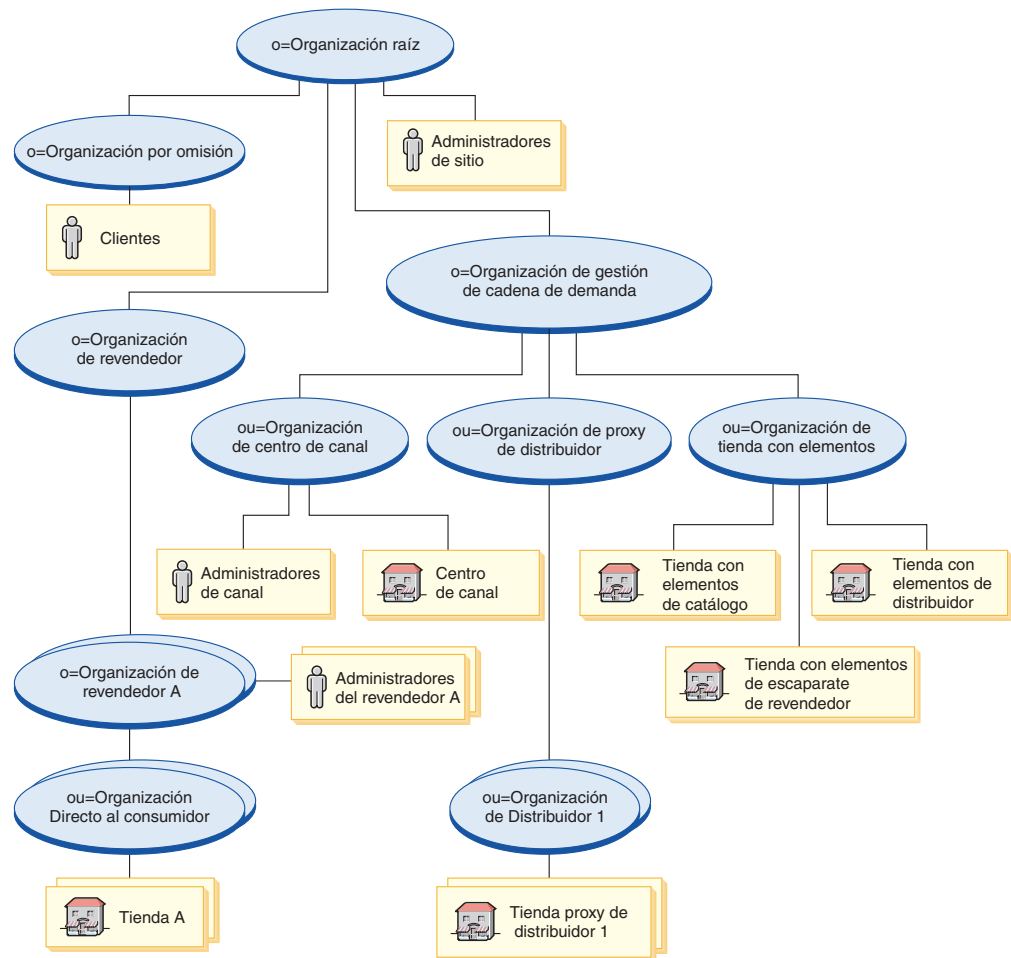
- **Organización raíz:** Todas las organizaciones del negocio se convierten en descendientes de la organización raíz. Los administradores de sitio, que mantienen el sitio en línea, son propiedad de la organización raíz.
 - **Organización por omisión:** A diferencia de lo que ocurre en la estructura de organización directo al consumidor, los clientes no son propiedad de la organización por omisión. En su lugar, los clientes son compradores propiedad de la organización compradora.
 - **Organización compradora:** A los clientes, que en las organizaciones Directo a B2B se les considera compradores, se les asigna su propia organización en la estructura de la organización Directo a B2B.
 - **Organización vendedora:** Se crea una organización vendedora para que sea la propietaria de todas las organizaciones que poseen tiendas. Los administradores que mantienen las funciones de la tienda (por ejemplo, los representantes de servicio al cliente, los jefes de producto y los gestores de catálogos) se denominan administradores de vendedores y son propiedad directa de la organización vendedora.
 - Bajo la organización vendedora se crea una unidad de organización hijo (ou), organización Directo a B2B, para que sea propietaria de la tienda (Negocio).

Cadena de demanda

El diagrama siguiente muestra un ejemplo de negocio de cadena de demanda.



Para poner este negocio en línea, las entidades del diagrama anterior deben asignarse a las organizaciones siguientes:

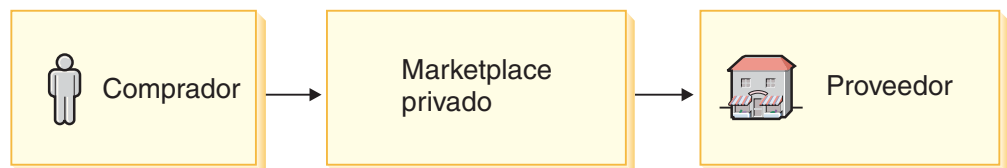


- **Organización raíz:** Todas las organizaciones del negocio se convierten en descendientes de la organización raíz. Asimismo, los administradores que mantienen el sitio en línea, los Administradores de sitio, se añaden directamente debajo de la organización raíz.
 - **Organización por omisión:** Por omisión, bajo la organización por omisión no se coloca nada. Los clientes de las tiendas de revendedor pueden colocarse bajo esta organización.
 - **Organización de gestión de la cadena de demanda:** La organización de gestión de la cadena de demanda se crea para que sea la propietaria de todas las organizaciones relacionadas con el canal (con la exclusión de la organización propietaria de los revendedores). La organización de gestión de la cadena de demanda es propietaria de las siguientes unidades de organización hijo:
 - **ou=Organización de centro de canal:** Incluye a los administradores de canal y al centro de canal.
 - **ou=Organización de proxy de distribuidor:** Incluye a las tiendas con elementos de catálogo, tiendas con elementos de distribuidor y tiendas con elementos de escaparate de revendedor.
 - **ou=Organización de tienda con elementos:** Incluye a las tiendas con elementos de escaparate de revendedor.
- **o=Organización de revendedor:** Incluye a los administradores del revendedor A.
- **o=Organización de revendedor A:** Incluye a la organización Directo al consumidor (que contiene a la tienda A) y a la organización de Distribuidor 1 (que contiene a la tienda proxy de distribuidor 1).

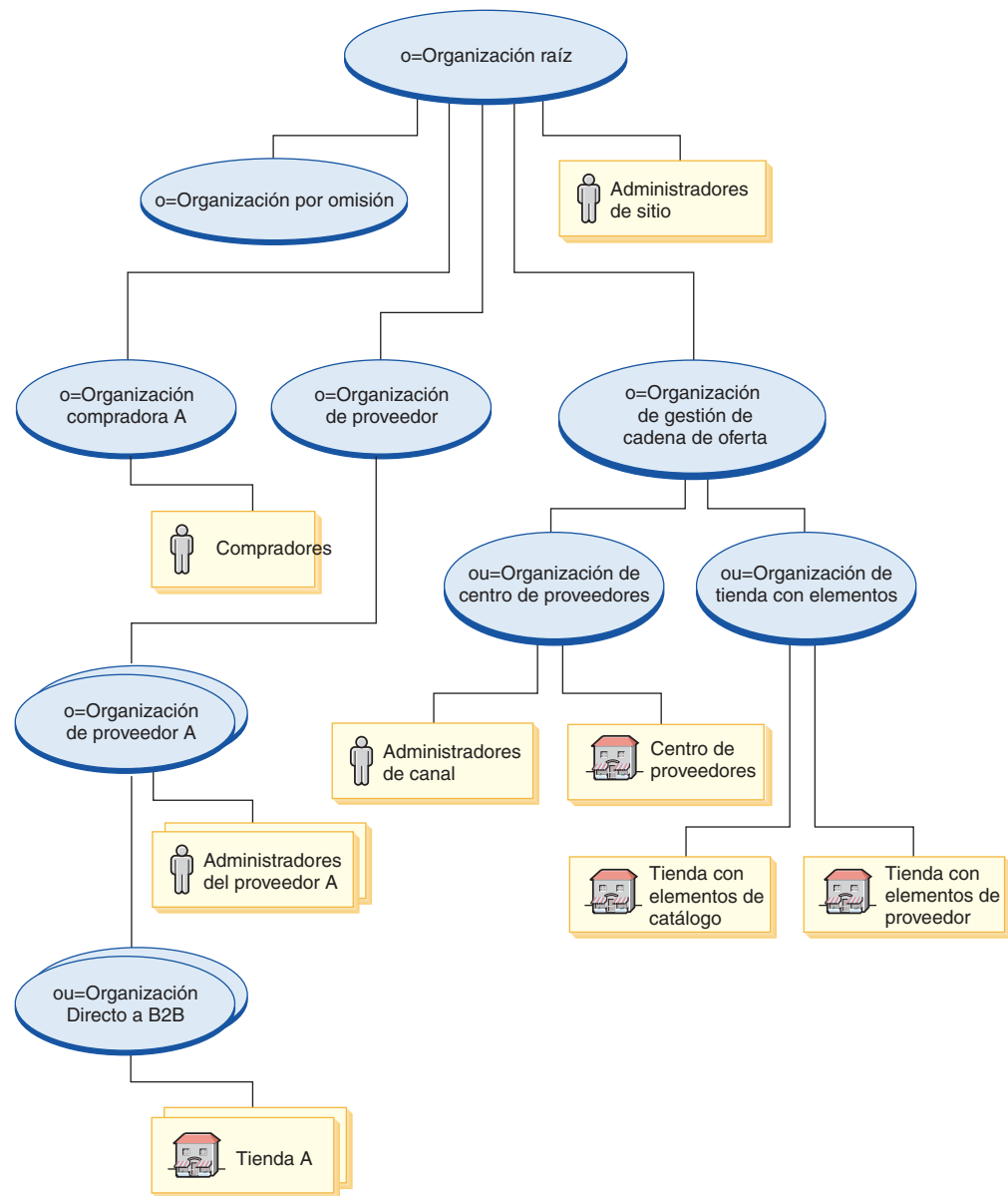
- **Organización de Centro de canal:** La organización de Centro de canal se crea para que sea la propietaria del Centro de canal. Los administradores que mantienen las funciones de los Centros de canal y que también administran la organización de revendedor, se denominan administradores de canal y son propiedad directa de la organización de Centro de canal.
- **Organización de proxy de distribuidor:** La organización de proxy de distribuidor se crea para que sea la propietaria de todas las conexiones con los distribuidores. Se crea una unidad de organización hijo para cada proxy de distribuidor de la organización.
 - **Organización de distribuidor:** Se crea una nueva unidad de organización de distribuidor para cada proxy de distribuidor del sitio.
- **Organización de tienda con elementos:** La organización de tienda con elementos se crea para que sea la propietaria de todos los elementos que se utilizan para crear las tiendas de los asociados del canal (revendedores y distribuidores).
- **Organización de revendedor:** La organización de revendedor se crea para que sea la propietaria de los revendedores de la cadena de demanda. Para cada revendedor se crea una organización hijo.
 - **Organización de revendedor A, B, C:** Se crea una organización de revendedor nueva bajo la organización de revendedor padre para cada tienda de revendedor. Los administradores que mantienen las funciones de la tienda (por ejemplo, los representantes de servicio al cliente, los jefes de producto y los gestores de catálogos) se denominan administradores de revendedores y son propiedad directa de la organización de revendedor correspondiente.

Cadena de oferta

El siguiente diagrama ilustra un negocio de cadena de oferta típico.



Para poner este negocio en línea, las entidades del diagrama anterior deben asignarse a las organizaciones siguientes:



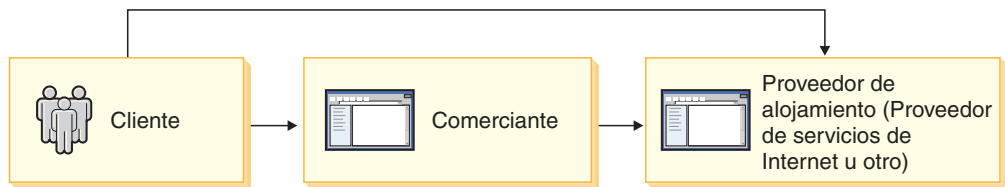
- **Organización raíz:** Todas las organizaciones del negocio se convierten en descendientes de la organización raíz. Asimismo, los administradores que mantienen el sitio en línea, los Administradores de sitio, se añaden directamente debajo de la organización raíz.
 - **Organización por omisión:** Por omisión, bajo la organización por omisión no se coloca nada.
 - **Organización de gestión de la cadena de oferta:** La organización de gestión de la cadena de demanda se crea para que sea la propietaria de todas las organizaciones relacionadas con la cadena de oferta (con la exclusión de la organización propietaria de los proveedores). La organización de gestión de la cadena de oferta es propietaria de las siguientes unidades de organización hijo:
 - **Organización de Centro de proveedores:** La organización de Centro de proveedores se crea para que sea la propietaria del centro de proveedores.

Los administradores que mantienen las funciones de los centros de proveedores, y que también administran la organización de proveedor, se denominan administradores de canal y son propiedad directa de la organización de Centro de proveedores.

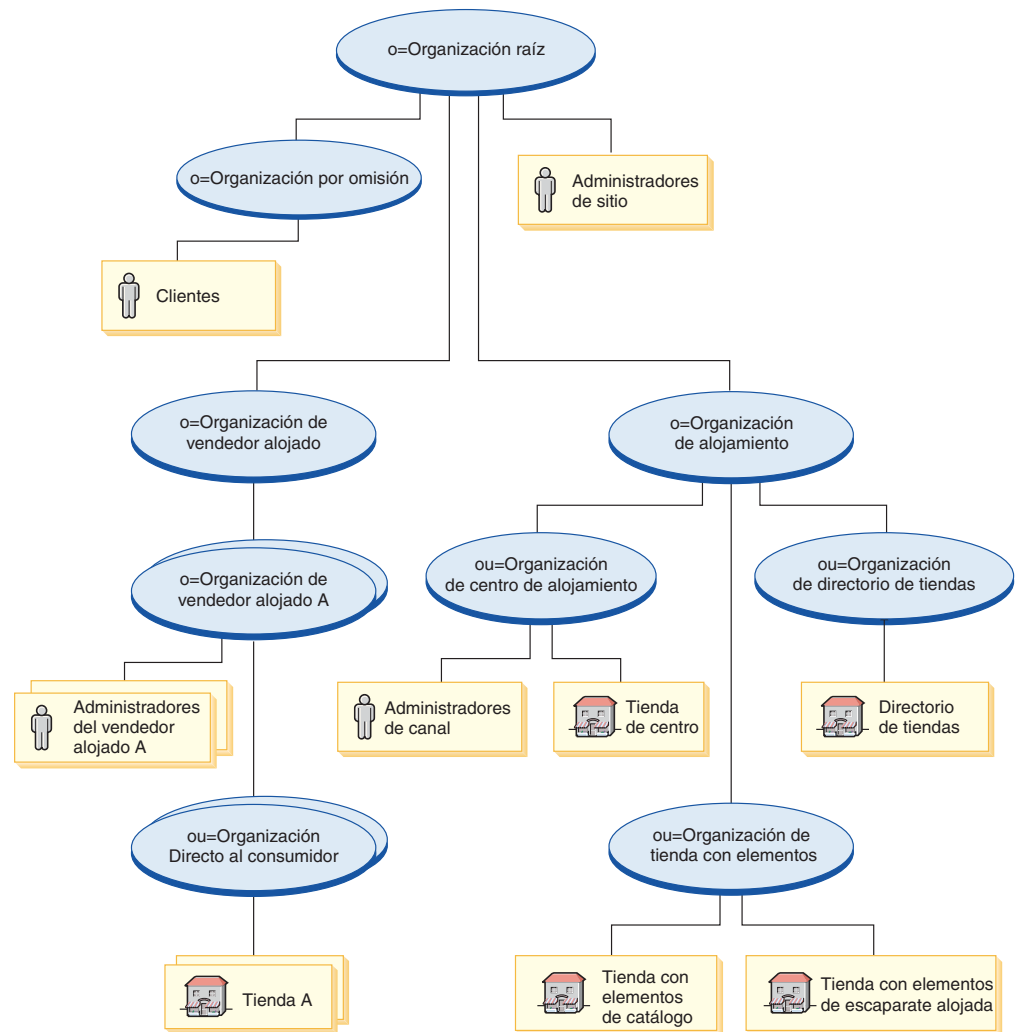
- **Organización de tienda con elementos:** La organización de tienda con elementos se crea para que sea la propietaria de todos los elementos que se utilizan para crear tiendas para proveedores.
- **Organización de proveedor:** La organización de proveedor se crea para que sea la propietaria de todos los proveedores de la cadena de oferta. Para cada revendedor se crea una organización hijo.
- **Organización de proveedor A, B, C:** Se crea una organización de proveedor nueva bajo la organización de proveedor padre para cada tienda de proveedor. Los administradores que mantienen las funciones de la tienda se denominan administradores de proveedores y son propiedad directa de la organización de proveedor correspondiente.
- **Organización compradora:** A los compradores se les proporciona su propia organización bajo la organización raíz. Todos los compradores son propiedad de la organización compradora correspondiente.

Alojamiento

El diagrama siguiente ilustra un negocio de alojamiento típico.



Para poner este negocio en línea, las entidades del diagrama anterior deben asignarse a las organizaciones siguientes:



- **Organización raíz:** Todas las organizaciones del negocio se convierten en descendientes de la organización raíz. Asimismo, los administradores que mantienen el sitio en línea, los Administradores de sitio, se añaden directamente debajo de la organización raíz.
 - **Organización por omisión:** Todos los clientes del negocio son propiedad de la organización por omisión.
 - **Organización de alojamiento:** La organización de alojamiento se crea para que sea la propietaria de todas las organizaciones relacionadas con alojamiento (con la exclusión de la organización propietaria de las tiendas alojadas). La organización de alojamiento es la propietaria de las siguientes unidades de organización hijo:
 - **Organización de Centro de alojamiento:** La organización de Centro de alojamiento se crea para que sea la propietaria del Centro de alojamiento. Los administradores que mantienen las funciones del centro de alojamiento, y que también administran la organización de alojamiento, se denominan administradores de canal y son propiedad directa de la organización de Centro de alojamiento.
 - **Organización de directorio de tiendas:** La organización de directorio de tiendas se crea para que sea la propietaria del directorio de tiendas.

- **Organización de tienda con elementos:** La organización de tienda con elementos se crea para que sea la propietaria de todos los elementos que se utilizan para crear tiendas alojadas.
- **Organización de vendedor alojado:** La organización de vendedor alojado se crea para que sea la propietaria de todas las tiendas alojadas. Se crea una unidad de organización hijo para cada tienda alojada.
- **Organización de tienda alojada A, B, C:** Se crea una organización de tienda alojada nueva bajo la organización de alojamiento padre para cada tienda alojada. Los administradores que mantienen las funciones de la tienda se denominan administradores de vendedores alojados y son propiedad directa de la organización de tienda alojada correspondiente.

Estructuras de organización de ejemplo

WebSphere Commerce proporciona estructuras de organización de ejemplo para cada modelo de negocio soportado. Estas estructuras de organización de ejemplo están disponibles por separado (como archivadores de tienda de componente), lo que le permite utilizar la estructura de organización de ejemplo como punto de partida para su sitio o como parte de los negocios de ejemplo. Para obtener más información sobre las estructuras de organización de ejemplo, consulte la publicación *WebSphere Commerce, Guía de tiendas de ejemplo*.

Creación de estructuras de organización

En lugar de crear nuevas estructuras de organización para el sitio, se le recomienda que comience publicando una de las estructuras de organización de ejemplo que se proporcionan con WebSphere Commerce, y que luego realice cambios en dicha estructura de organización según sea necesario. Para obtener más información sobre cómo editar datos de organización, consulte el apartado “Los elementos de miembro en WebSphere Commerce” en la página 125.

Capítulo 4. Control de acceso en WebSphere Commerce

WebSphere Commerce le permite determinar, mediante el control de acceso, qué tareas puede realizar un usuario determinado, ya sean clientes, compradores, administradores, distribuidores, fabricantes o proveedores en su negocio.

El modelo de control de acceso de WebSphere Commerce se describe detalladamente en la publicación *WebSphere Commerce, Guía de seguridad*. No obstante, para comprender cómo el control de acceso afecta al desarrollo de sitios y tiendas se proporciona un breve resumen.



Descripción del control de acceso en WebSphere Commerce

El control de acceso en WebSphere Commerce consta de los elementos siguientes: usuarios, acciones, recursos y relaciones.

- Los usuarios son aquellos que utilizan el sistema. Para fines de control de acceso, los usuarios se deben agrupar en grupos de acceso relevantes. Los roles son uno de los atributos comunes que se utilizan para determinar los miembros de un grupo de acceso. Los roles se asignan a los usuarios según la organización. Para obtener más información acerca de los roles, consulte el apartado “Roles” en la página 127. Algunos ejemplos de grupos de acceso son clientes registrados, clientes invitados o grupos administrativos como, por ejemplo, representantes de servicio al cliente.
- Las acciones son las actividades que los usuarios pueden realizar en el recurso. Para fines de control de acceso, las acciones también se deben agrupar en grupos de acciones relevantes. Por ejemplo, una acción que se utiliza comúnmente en una tienda es una vista. Las vistas se invocan para mostrar una página de la tienda a los clientes. Para poder acceder a las vistas que se utilizan en la tienda, deben declararse como acciones y asignarse a un grupo de acciones.
- Los recursos son entidades protegidas. Por ejemplo, si la acción es una vista, el recurso que se ha de proteger es el mandato que ha invocado la vista como, por ejemplo, `com.ibm.commerce.command.ViewCommand`. Para fines de control de acceso, los recursos se agrupan en grupos de recursos.
- Las relaciones son la relación que existe entre el usuario y el recurso. Es posible que las políticas de control de acceso requieran que se cumpla una relación entre el usuario y el recurso. Por ejemplo, es posible que a los usuarios sólo se les permita visualizar los pedidos que han creado.

Políticas de control de acceso

Las políticas de control de acceso autorizan a los grupos de acceso a realizar acciones determinadas en los recursos de WebSphere Commerce, siempre que los usuarios del grupo de acceso satisfagan una relación determinada con respecto al recurso.

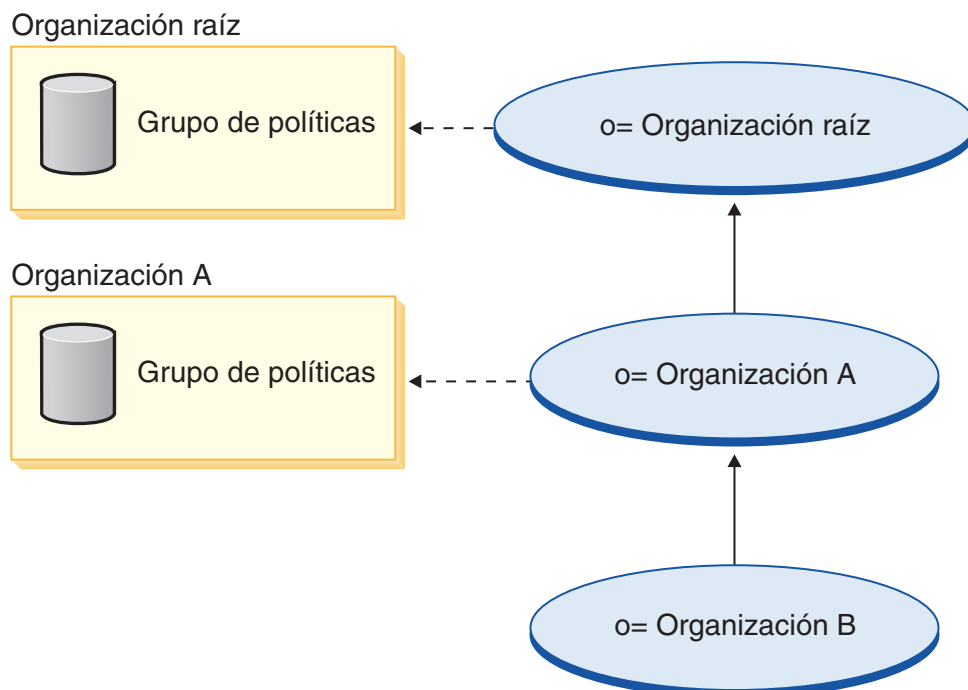
WebSphere Commerce proporciona más de trescientas políticas de control de acceso por omisión que se cargan durante la creación de la instancia. Estas políticas cubren una amplia gama de actividades comerciales comunes, incluyendo la creación y el proceso de pedidos y el intercambio, como por ejemplo las  solicitudes de presupuesto y  los contratos. Las políticas por omisión se describen en la publicación *WebSphere Commerce, Guía de seguridad*.

Grupos de políticas de control de acceso

Para que se aplique una política de control de acceso en la tienda o sitio, dicha política debe pertenecer a un grupo de políticas de control de acceso y la organización propietaria del recurso debe suscribirse al grupo de políticas. Por omisión, todas las políticas de control de acceso que se proporcionan con WebSphere Commerce están asignadas a grupos de políticas. Para obtener más información acerca de las políticas por omisión que se proporcionan con WebSphere Commerce, consulte la publicación *WebSphere Commerce, Guía de seguridad*.

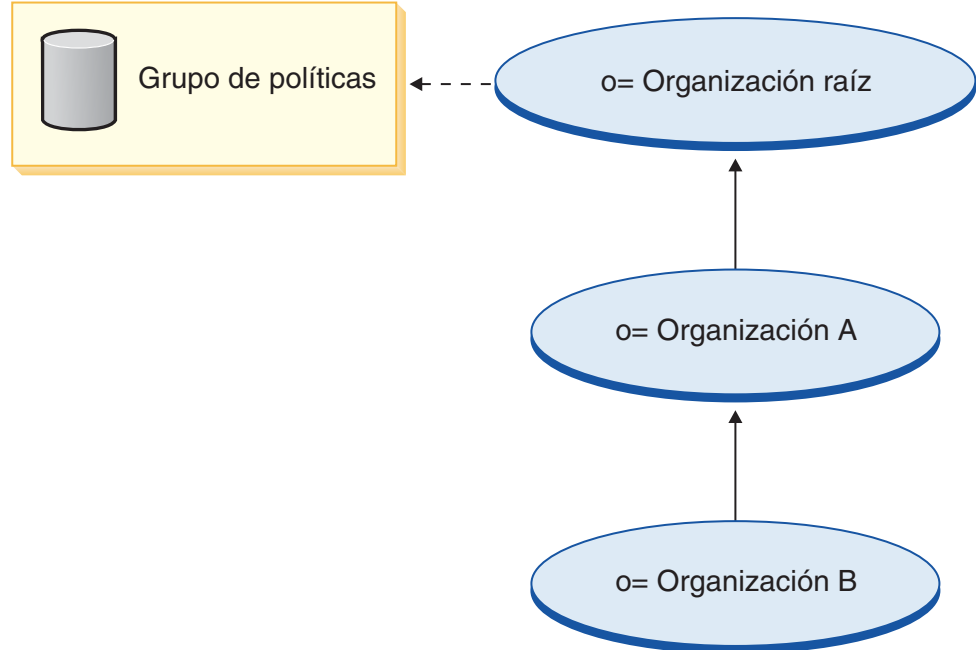
Aunque los grupos de políticas de control de acceso son propiedad de las organizaciones, no se aplican automáticamente a la organización. Una organización debe suscribirse a un grupo de políticas para que las políticas de control de acceso se apliquen a la organización. Si la organización tiene organizaciones hijo, todos los grupos de políticas a los que se suscriba la organización padre se aplicarán automáticamente a las organizaciones hijo. No obstante, si la organización hijo se suscribe directamente a un grupo de políticas, los grupos de políticas a los que se suscriba la organización padre ya no se aplicarán a la organización hijo.

En las versiones anteriores de WebSphere Commerce, una política se aplicaba a todos los recursos propiedad de los descendientes de la organización propietaria de dicha política. Por ejemplo, si la Organización A tenía una política determinada y era el padre de la Organización B, entonces la Organización B también tenía implícitamente dicha política. En WebSphere Commerce 5.5, las organizaciones se pueden suscribir a grupos de políticas. En WebSphere Commerce 5.5, si la Organización B no se suscribe a ningún grupo de políticas, la infraestructura de control de acceso comenzará a buscar en la jerarquía de organizaciones hasta que encuentre una organización que se suscribe al menos a un grupo de políticas. Si la organización padre inmediata de la Organización B, la Organización A, se suscribe a un grupo de políticas, la búsqueda se detiene y las políticas del grupo de políticas de la Organización A se aplican a las Organizaciones A y B. Esto puede observarse en el diagrama siguiente.



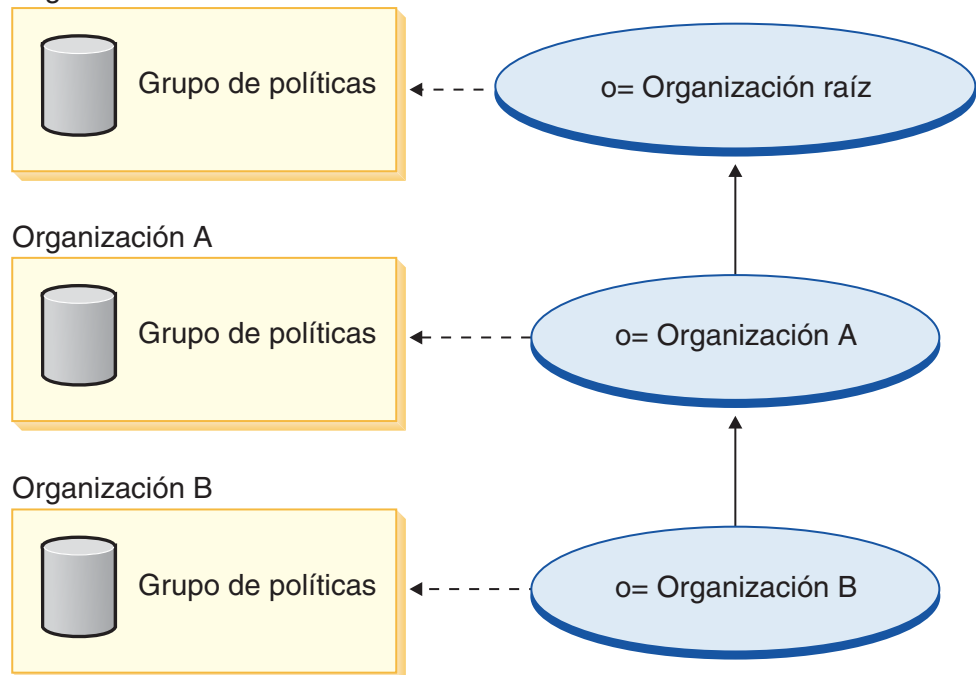
Si la Organización A no se suscribe a ningún grupo de políticas, la búsqueda continúa por la jerarquía de organizaciones hasta que se encuentra una organización con una suscripción. Esto puede observarse en el diagrama siguiente en el que la Organización raíz se suscribe a un grupo de políticas. La Organización B y la Organización A heredan las políticas de dicho grupo.

Organización raíz



Si la Organización B se suscribe a un grupo de políticas, la búsqueda se detiene en la Organización B, y la Organización B solamente puede aplicar las políticas a las que se ha suscrito, como se muestra en el diagrama siguiente.

Organización raíz



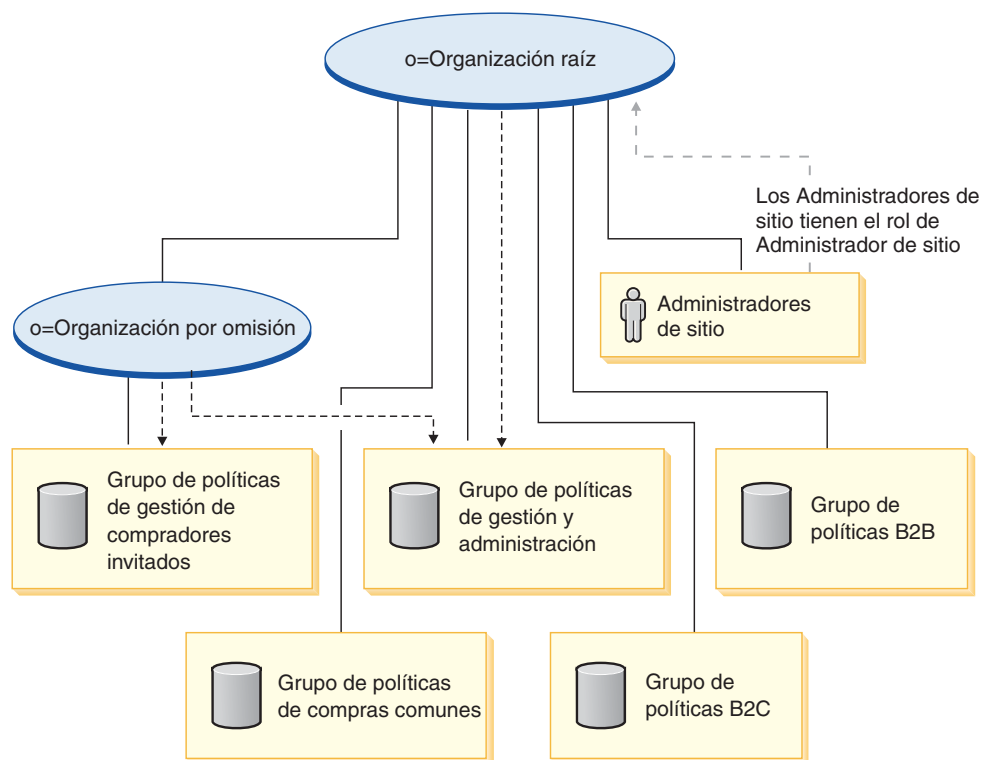
Nota: En relación con el control de acceso, la propiedad de los recursos tiene un significado especial. Todos los recursos deben implementar la interfaz `com.ibm.commerce.security.Protectable`. Uno de los métodos de esta interfaz es `getOwner()`, que devuelve el ID de miembro del propietario del recurso. Por ejemplo, el bean de entidad de pedido es un recurso protegido ya que su interfaz remota amplía la interfaz `Protectable`. La implementación del pedido de `getOwner()` es tal que un recurso de pedido específico devuelve el propietario de la tienda en la que se ha formalizado el pedido. En el caso de las políticas en las que el recurso es un mandato, por ejemplo, `com.ibm.commerce.command.ViewCommand`, la implementación por omisión de `getOwner()` devuelve el propietario de la tienda que actualmente está en el contexto del mandato. Si no hay ninguna tienda en el contexto del mandato, entonces se utiliza la organización raíz como propietario. Para obtener más información, consulte la publicación *WebSphere Commerce, Guías de programación y aprendizaje*.

Descripción del control de acceso en los modelos de negocio

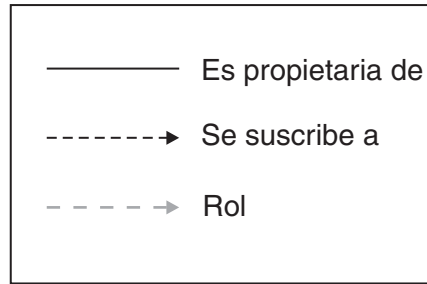
La estructura de control de acceso de WebSphere Commerce es lo suficientemente flexible como para dar soporte a todas las entidades de los modelos de negocio soportados. Los diagramas de los apartados siguientes muestran cómo se aplica el control de acceso a un ejemplo típico de cada modelo de negocio.

Estructura de control de acceso básica

La estructura de control de acceso básica se instala durante la creación de la instancia, independientemente del modelo de negocio.



Leyenda



La organización raíz es propietaria de los grupos de políticas por omisión siguientes:

- Gestión y administración
- Compras comunes
- B2C
- B2B

No obstante, la organización raíz sólo se suscribe al grupo de políticas de gestión y administración. Como resultado, estas políticas se aplican a los administradores de sitio, que están directamente bajo la organización raíz.

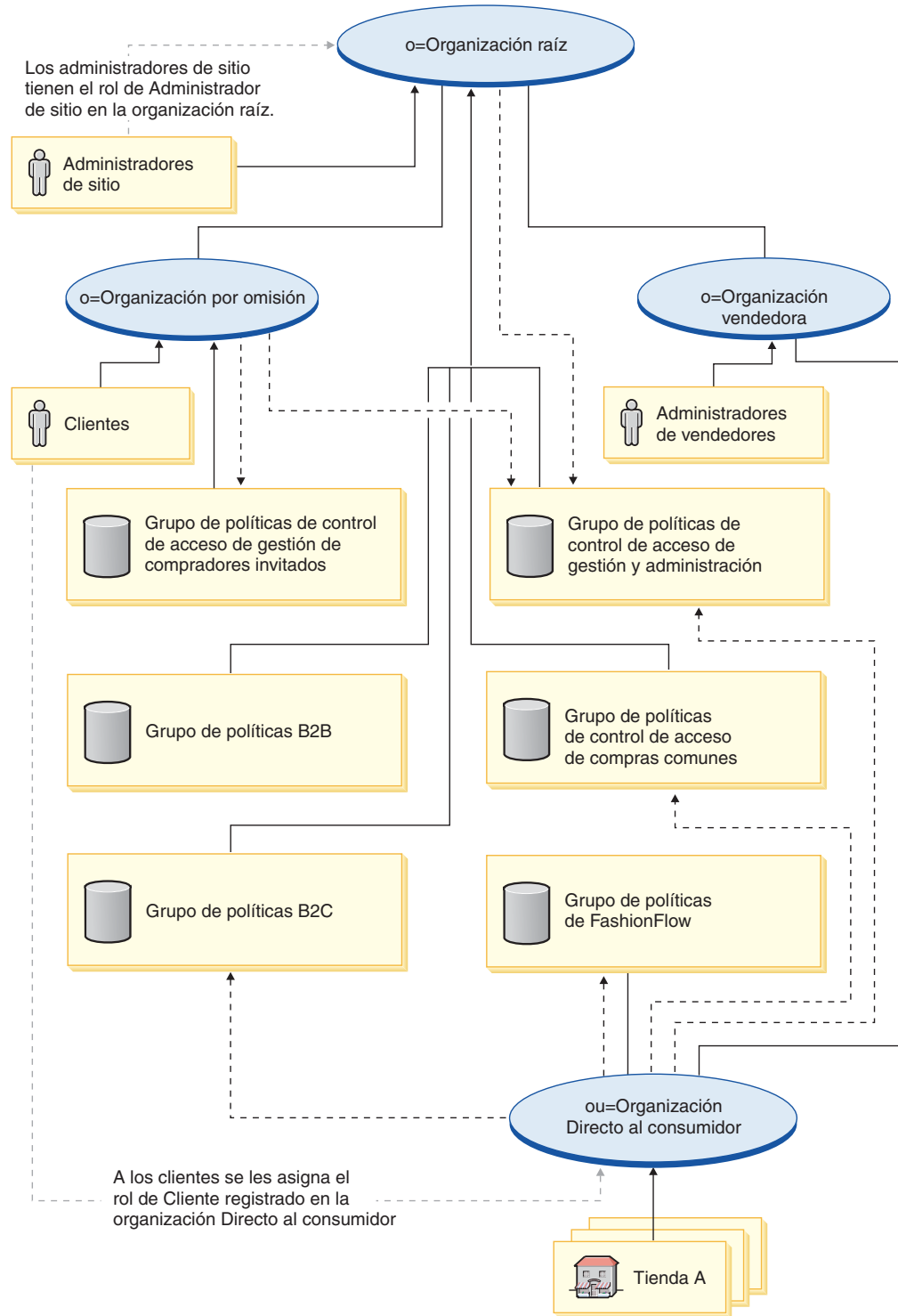
Las políticas del grupo de políticas de gestión y administración no se aplican a la organización por omisión mediante herencia, ya que la organización por omisión se suscribe al grupo de políticas de gestión de compradores invitados. Para que se puedan aplicar las políticas de gestión y administración, la organización por omisión debe suscribirse explícitamente al grupo de políticas de gestión y administración.

La organización por omisión es propietaria del grupo de políticas de gestión de compradores invitados.

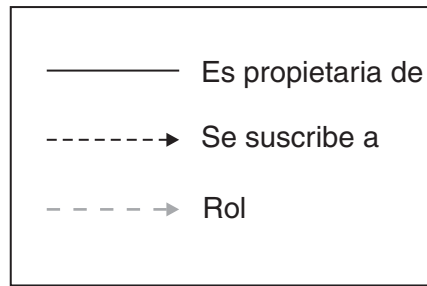
Nota: Para obtener información detallada acerca de los grupos de políticas por omisión, consulte el apéndice de la publicación *WebSphere Commerce, Guía de seguridad*.

Directo al consumidor

El diagrama siguiente describe una organización Directo al consumidor y una estructura de control de acceso básicas.



Leyenda



En este diagrama que describe la organización Directo al consumidor básica, la organización raíz es propietaria y se suscribe a los grupos de políticas por omisión, como se describe en el apartado “Estructura de control de acceso básica” en la página 40.

La organización Directo al consumidor se suscribe directamente a las políticas de control de acceso B2C, al grupo de políticas de gestión y administración y al grupo de políticas de compras comunes.

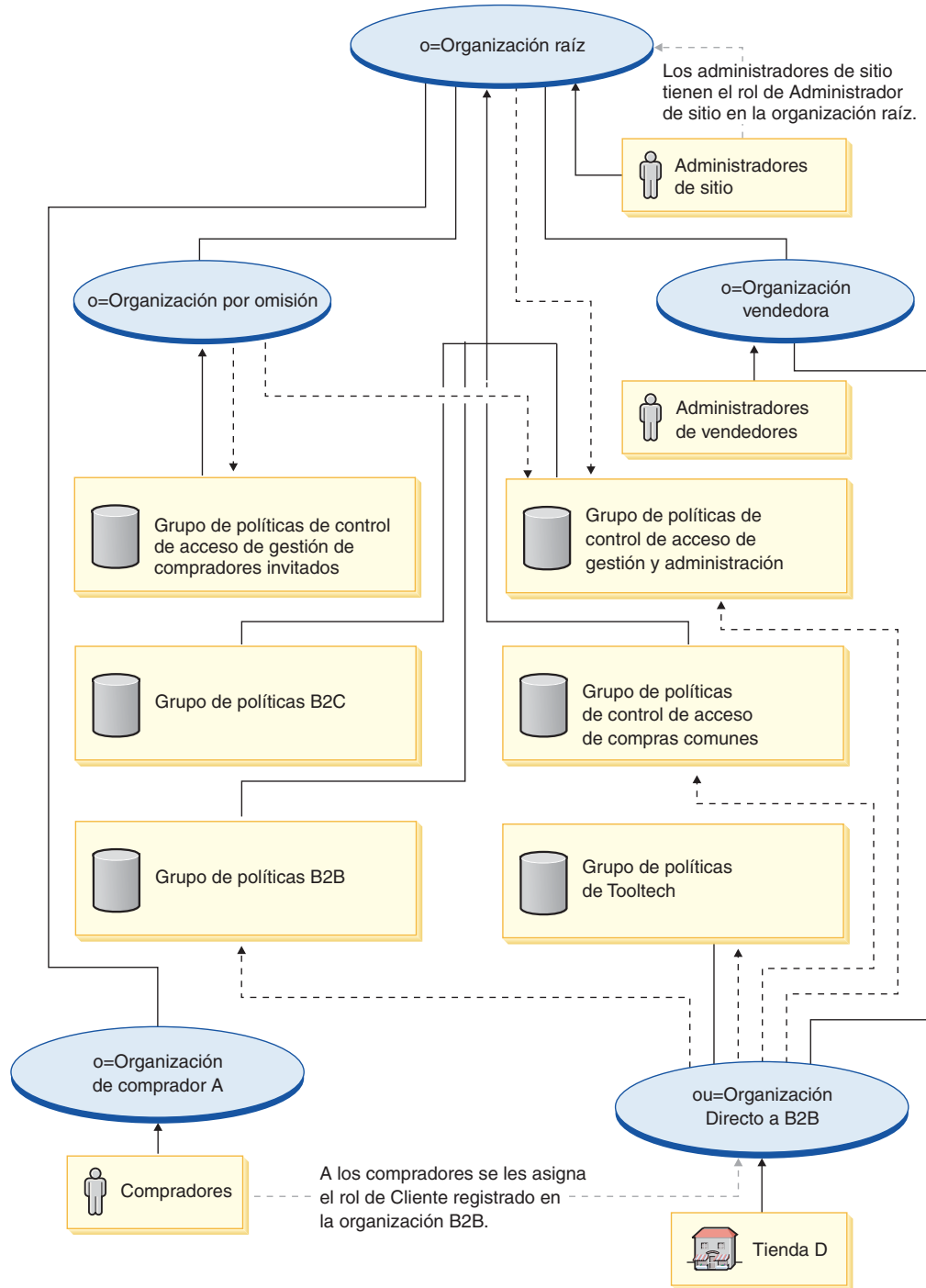
La organización Directo al consumidor también es propietaria y se suscribe al grupo de políticas de FashionFlow. El grupo de políticas de FashionFlow contiene la política siguiente:

AllUsersExecuteFashionAllUsersViews

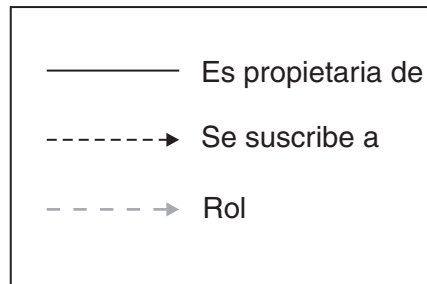
Puesto que las entidades de organización se suscriben a los grupos de políticas de control de acceso, si va a crear varias tiendas en el sitio, y desea aplicar grupos de políticas de control de acceso diferentes a tiendas individuales, debe crear organizaciones independientes para que sean propietarias de cada tienda.

Directo a B2B

El diagrama siguiente describe una organización Directo a B2B y una estructura de control de acceso básicas.



Leyenda



En este diagrama, que describe una estructura de organización Directo a B2B básica, la organización raíz es propietaria y se suscribe a los grupos de políticas por omisión, como se describe en el apartado “Estructura de control de acceso básica” en la página 40.

La organización Directo a B2B se suscribe directamente a los grupos de políticas B2B, Gestión y administración y Compras comunes.

La organización Directo a B2B también es propietaria y se suscribe al grupo de políticas de ToolTech. El grupo de políticas de ToolTech contiene las políticas siguientes:

- AllUsersForToolTechExecuteToolTechAllUsersViews
- RegisteredCustomersForOrgForToolTechExecuteToolTechRegisteredCustomerViews

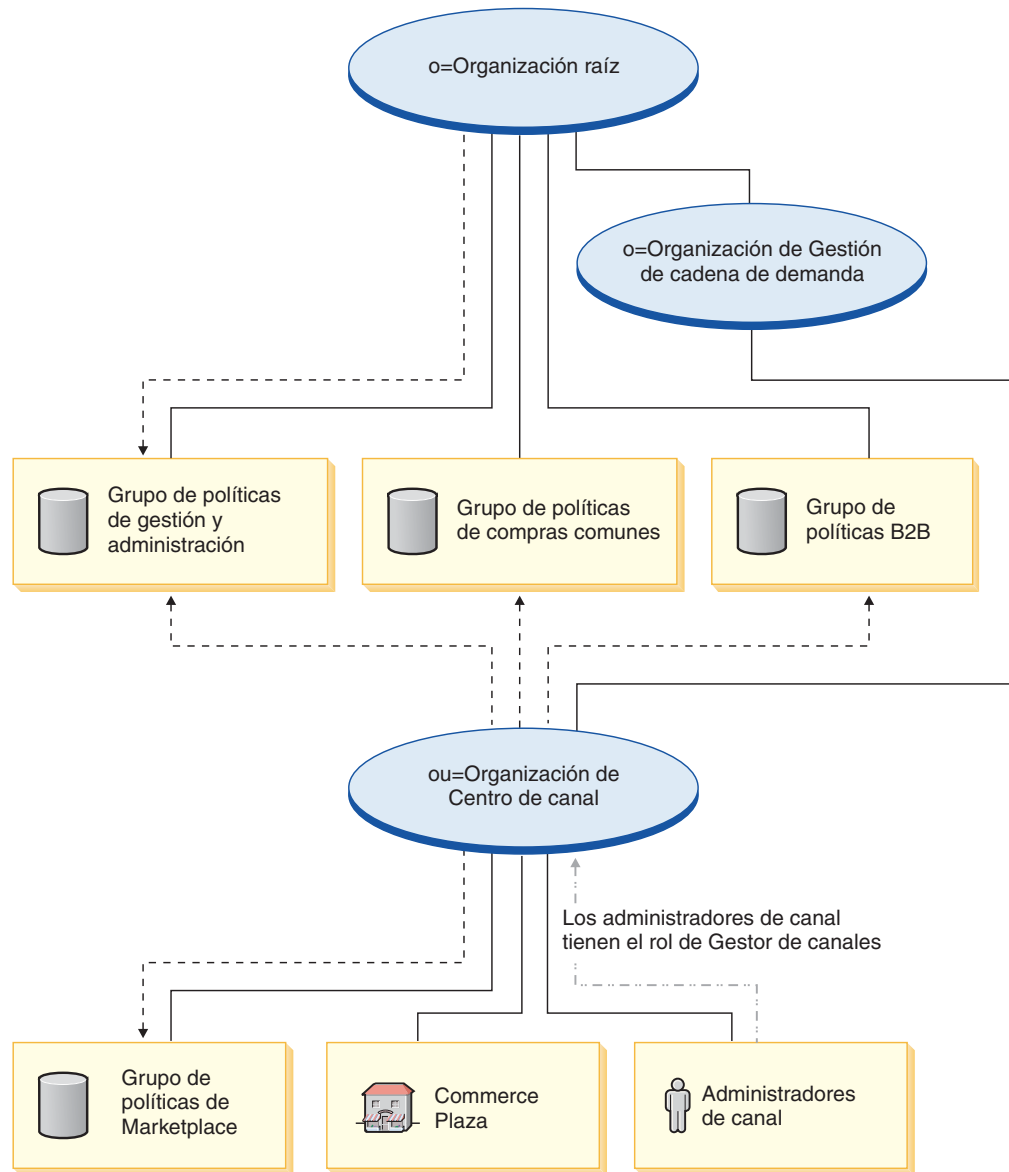
Los compradores son clientes que formalizan pedidos en una tienda Directo a B2B. Todos los compradores deben pertenecer a una organización compradora. Normalmente, las organizaciones compradoras no se suscriben a ningún grupo de políticas, ya que las políticas de gestión y administración heredadas de la organización raíz son suficientes.

Puesto que las entidades de organización se suscriben a los grupos de políticas de control de acceso, si va a crear varias tiendas en el sitio, y desea aplicar grupos de políticas de control de acceso diferentes a tiendas individuales, debe crear organizaciones independientes para que sean propietarias de cada tienda.

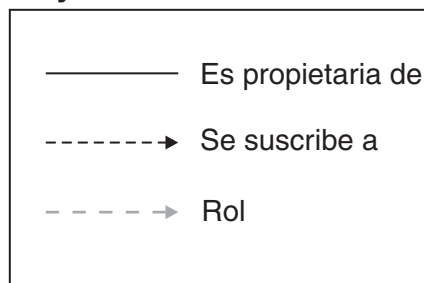
Cadena de demanda

En este diagrama, que describe una estructura de organización de cadena de demanda, la organización raíz es la propietaria y se suscribe a los grupos de políticas por omisión, como se describe en el apartado “Estructura de control de

acceso básica” en la página 40.



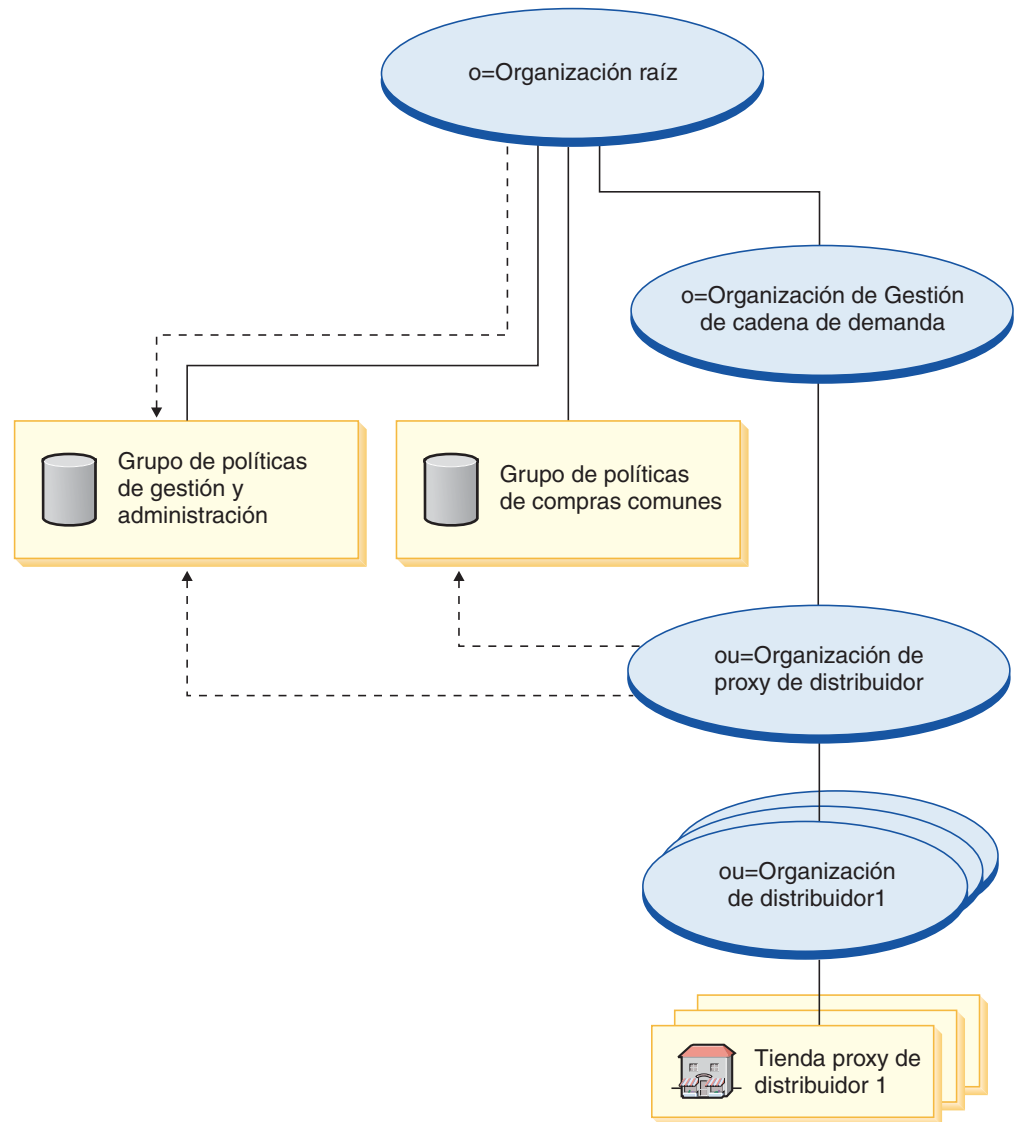
Leyenda



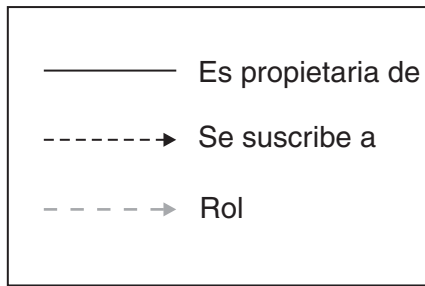
La organización de centro de canal se suscribe directamente al grupo de políticas de gestión y administración, al grupo de políticas de compras comunes, al grupo de políticas B2B y es propietaria y se suscribe al grupo de políticas de Marketplace. Como resultado, estas políticas se aplican a los administradores de canal, que están directamente bajo la organización de centro de canal, así como al centro de canal (Commerce Plaza).

El grupo de políticas de Marketplace contiene las políticas siguientes:

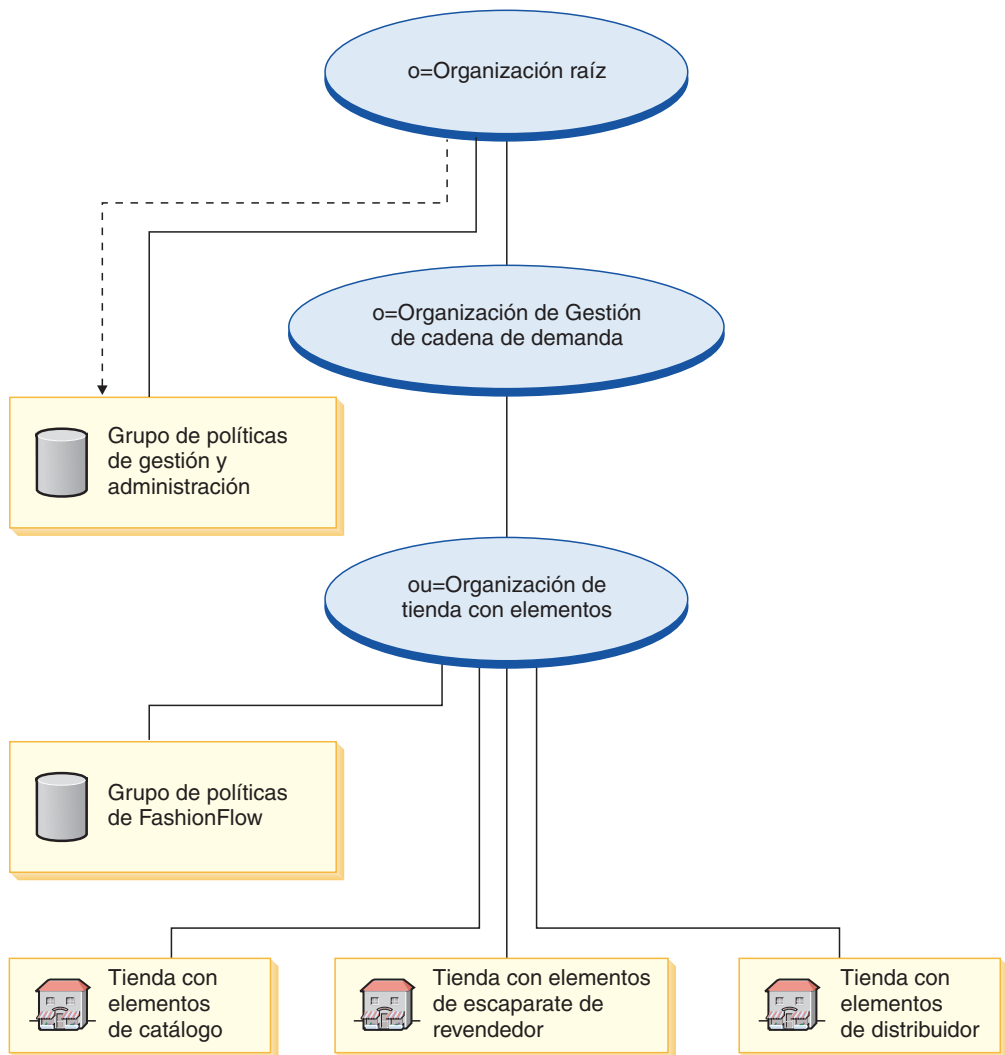
- AllUsersExecuteMarketplaceAllUserViews
- RegisteredCustomersForOrgExecuteMarketplaceRegisteredCustomerViews
- ContractAdministratorsForChannelOrgExecuteCreateCommandsOnMemberResource
- ContractAdministratorsForChannelOrgExecuteContractDeployCommandsOnContractResource
- ContractAdministratorsForChannelOrgDisplayContractDatabaseResourceGroup



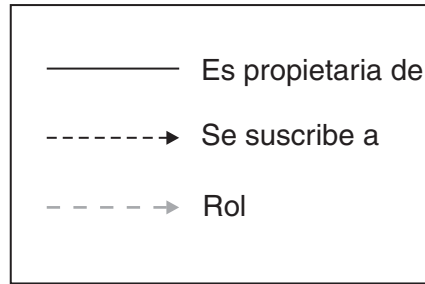
Leyenda



La organización de proxy de distribuidor se suscribe al grupo de políticas de gestión y administración y al grupo de políticas de compras comunes. Como resultado, estas políticas se aplican a las organizaciones de distribuidor, que están directamente bajo la organización de proxy de distribuidor.

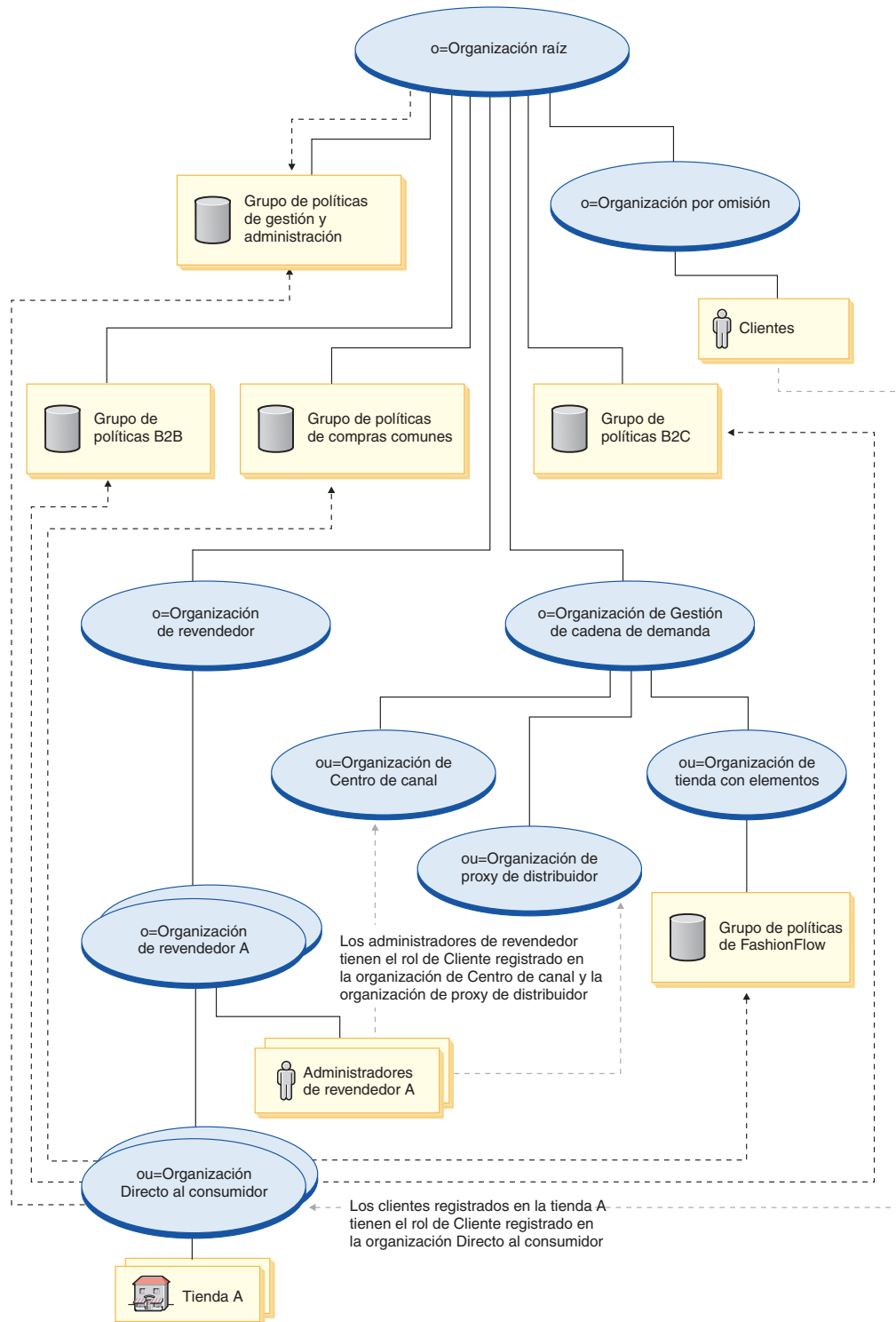


Leyenda

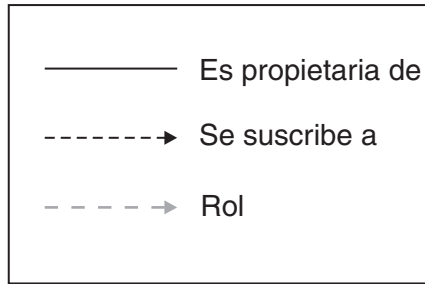


La organización de tienda con elementos no se suscribe directamente a ningún grupo de políticas. Como consecuencia, hereda el grupo de políticas de gestión y administración de la organización raíz. Estas políticas se aplican a la organización de tienda con elementos y a las tiendas con elementos de las que es propietaria. La organización de tienda con elementos es propietaria del grupo de políticas de FashionFlow, pero no se suscribe a él.

Nota: Las organizaciones Directo al consumidor de revendedor individuales se suscribirán al grupo de políticas de FashionFlow cuando se cree la tienda de revendedor.



Leyenda

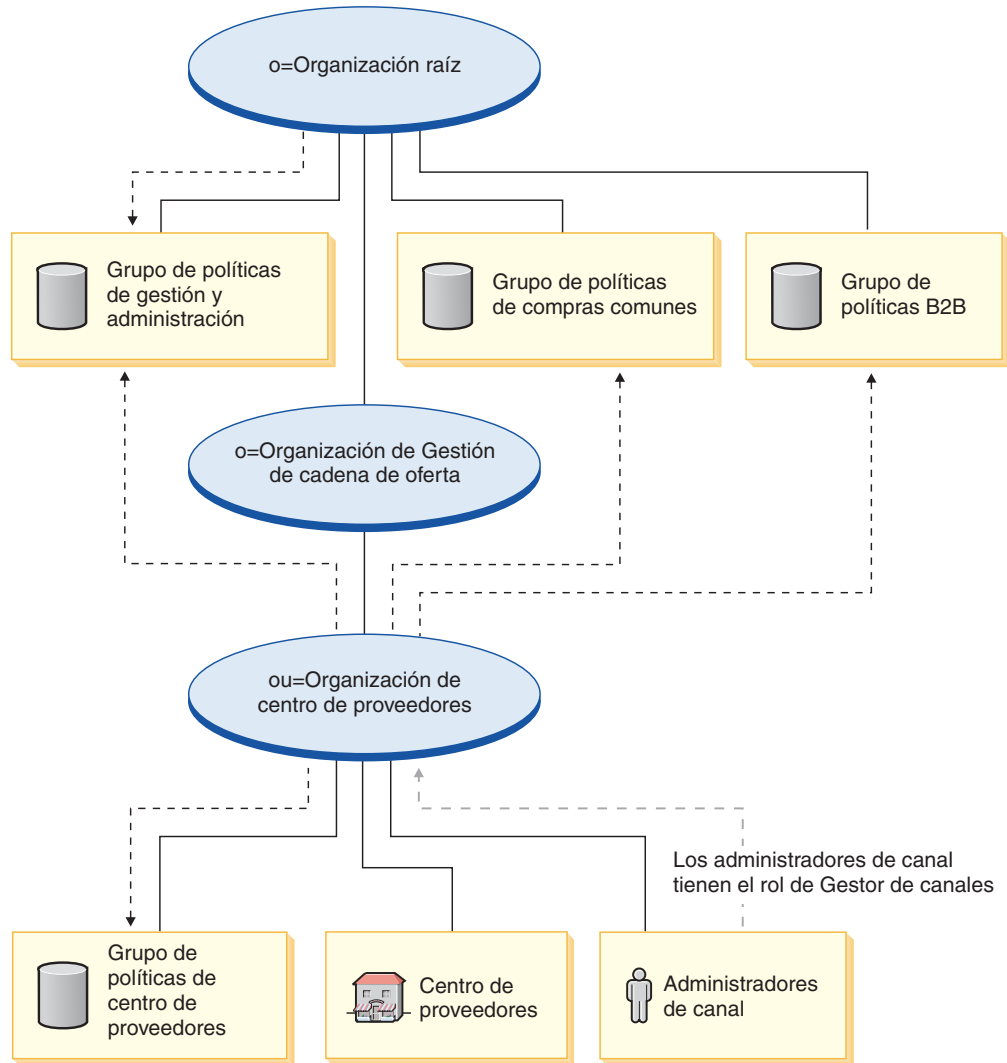


La organización de revendedor no se suscribe directamente a ningún grupo de políticas. Como consecuencia, hereda el grupo de políticas de gestión y administración de la organización raíz. Estas políticas se aplican a la organización de revendedor y a las organizaciones del revendedor A de las que es propietaria, así como a los administradores del revendedor A.

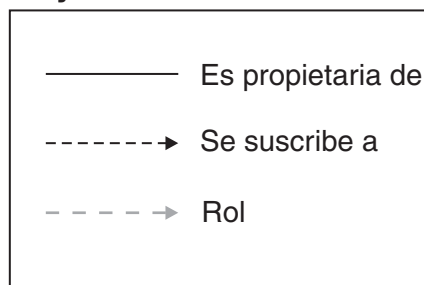
La organización Directo al consumidor se suscribe directamente al grupo de políticas de gestión y administración, al grupo de políticas de compras comunes, a los grupos de políticas B2B y B2C, así como al grupo de políticas de FashionFlow. Estas políticas se aplican a todas las tiendas que son propiedad de la organización Directo al consumidor.

Cadena de oferta

En estos diagramas, que describen una estructura de organización de cadena de oferta básica, la organización raíz es propietaria y se suscribe a los grupos de políticas por omisión como se describe en el apartado “Estructura de control de acceso básica” en la página 40.



Leyenda

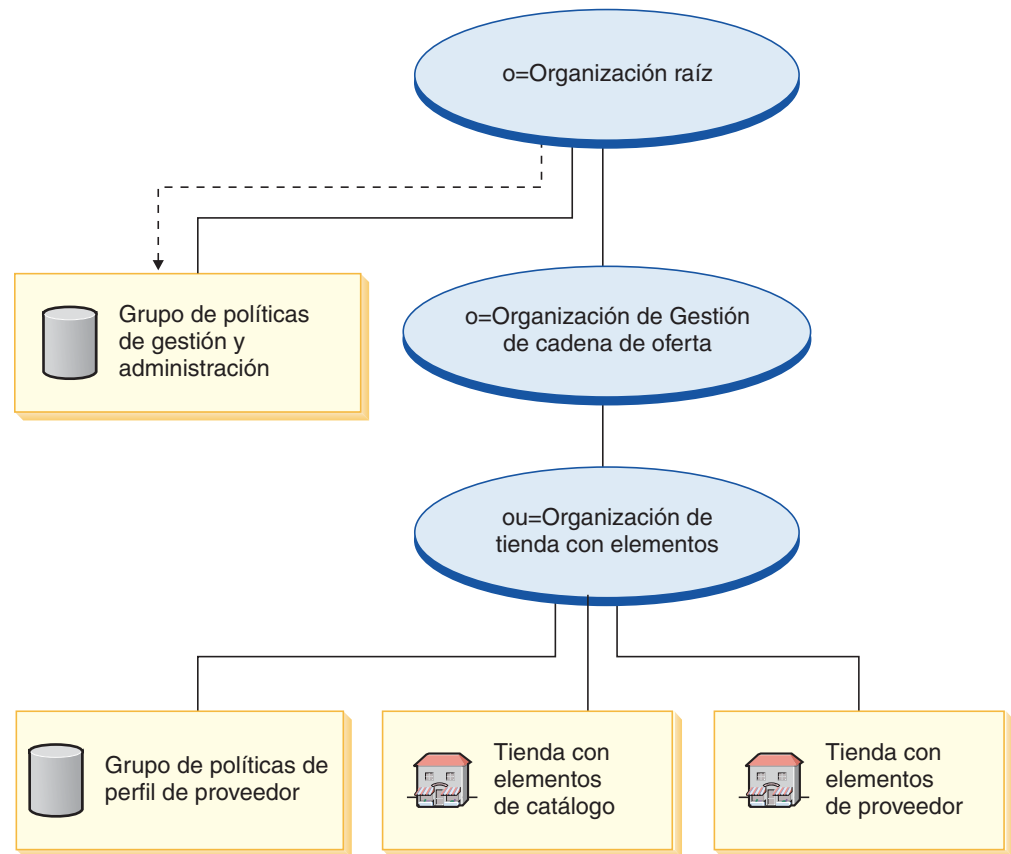


La organización de centro de proveedores se suscribe directamente al grupo de políticas de gestión y administración, al grupo de políticas de compras comunes, al grupo de políticas B2B y es propietaria y se suscribe al grupo de políticas de Centro de proveedores. Como resultado, estas políticas se aplican a los

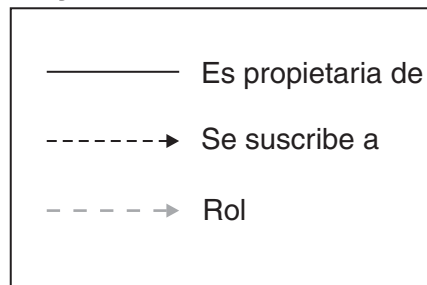
administradores de canal, que están directamente bajo la organización de centro de proveedores, así como al centro de proveedores.

El grupo de políticas de Centro de proveedores contiene las políticas siguientes:

- AllUsersForSupplierHubExecuteSupplierHubAllUsersViews
- RegisteredCustomersForOrgForSupplierHubExecuteSupplierHubRegisteredCustomerViews
- ContractAdministratorsForChannelOrgExecuteCreateCommandsOnMemberResource
- ContractAdministratorsForChannelOrgExecuteContractDeployCommandsOnContractResource
- ContractAdministratorsForChannelOrgDisplayContractDatabaseanResourceGroup



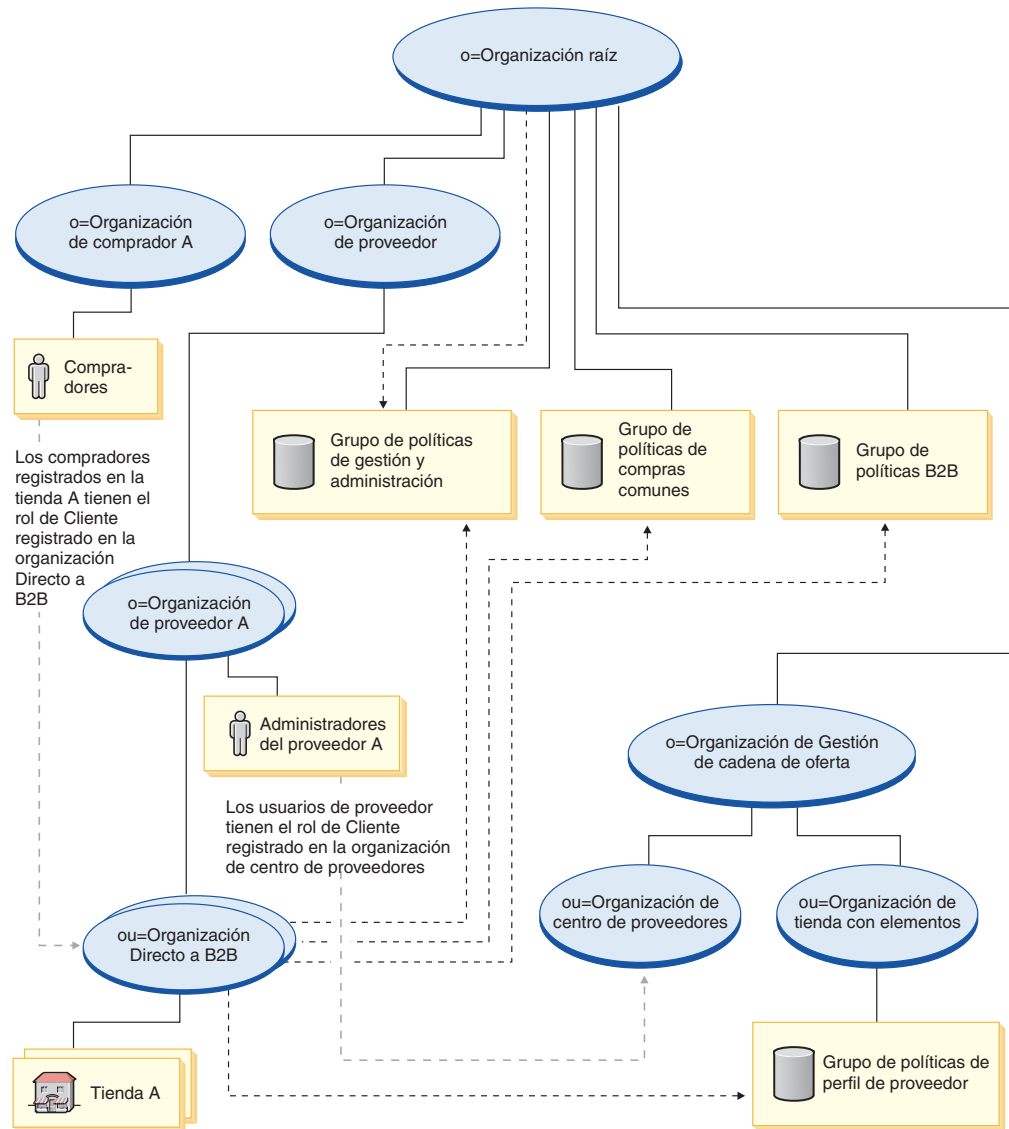
Leyenda



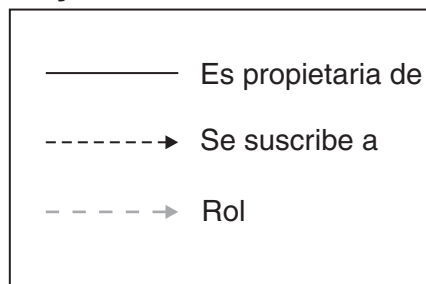
La organización de tienda con elementos no se suscribe directamente a ningún grupo de políticas. Como consecuencia, hereda el grupo de políticas de gestión y

administración de la organización raíz. Estas políticas se aplican a la organización de tienda con elementos y a las tiendas con elementos de las que es propietaria. La organización de tienda con elementos es propietaria del grupo de políticas de perfil de proveedor, pero no se suscribe a él.

Nota: La organización Directo a B2B del proveedor individual se suscribirá al grupo de políticas de perfil de proveedor cuando se cree la tienda de proveedor.



Leyenda



La organización de proveedor no se suscribe directamente a ningún grupo de políticas. Como consecuencia, hereda el grupo de políticas de gestión y administración de la organización raíz. Estas políticas se aplican a la organización de proveedor, a las organizaciones del proveedor A de las que es propietaria, así como a los administradores del proveedor A.

La organización Directo a B2B se suscribe directamente a los grupos de políticas de gestión y administración, compras comunes, B2B y de perfil de proveedor. Estas políticas se aplican a todas las tiendas que son propiedad de la organización Directo a B2B.

El grupo de políticas de Perfil de proveedor contiene las políticas siguientes:

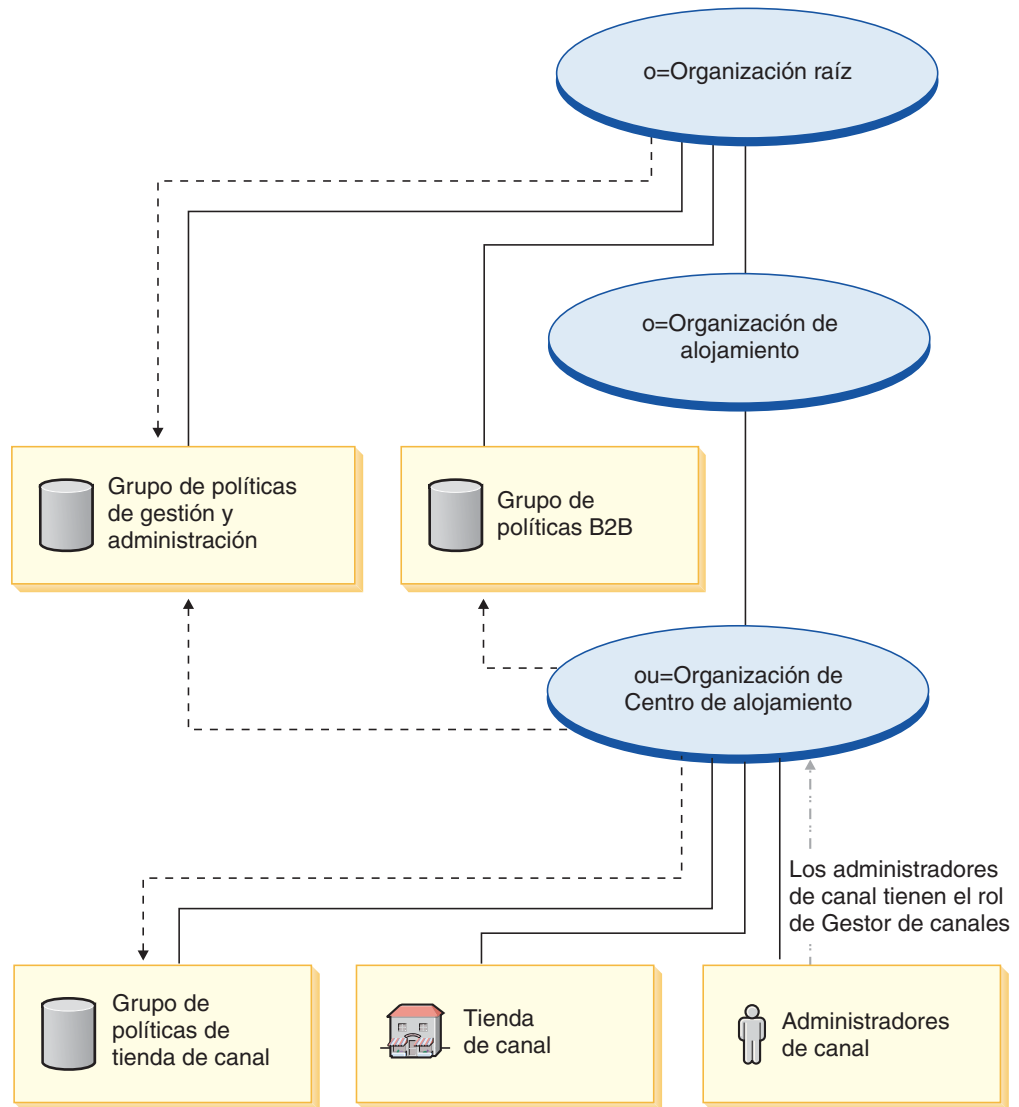
- AllUsersForSupplierExecuteSupplierAllUsersViews
- RegisteredCustomersForOrgForSupplierExecuteSupplierRegisteredCustomerViews

Los compradores son clientes que formalizan pedidos en una tienda B2B. Todos los compradores deben pertenecer a una organización compradora. Normalmente, las organizaciones compradoras no se suscriben a ningún grupo de políticas, ya que las políticas de gestión y administración heredadas de la organización raíz son suficientes.

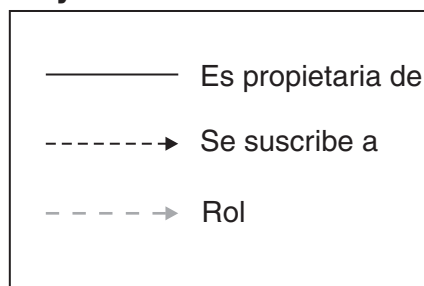
Alojamiento

En este diagrama, que describe una estructura de organización de alojamiento básica, la organización raíz es la propietaria y se suscribe a los grupos de políticas por omisión, como se describe en el apartado "Estructura de control de acceso

básica” en la página 40.



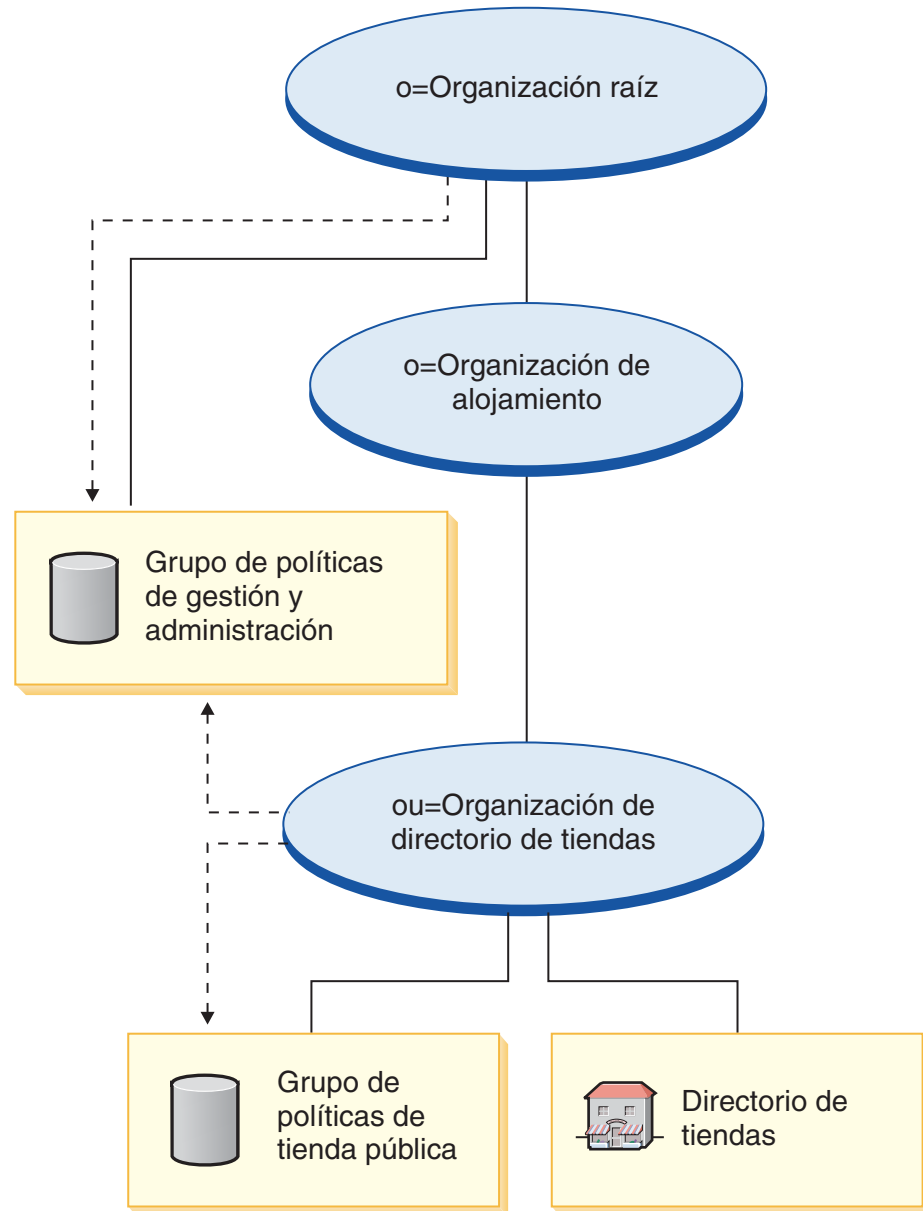
Leyenda



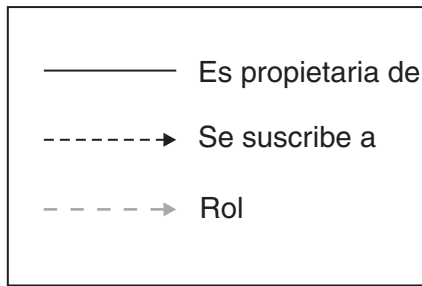
La organización de centro de alojamiento se suscribe directamente al grupo de políticas de gestión y administración, al grupo de políticas B2B y es propietaria y se suscribe al grupo de políticas de tienda de canal. Como resultado, estas políticas se aplican a los administradores de canal, que están directamente bajo la organización de centro de alojamiento, así como a la tienda de canal (centro de alojamiento).

El grupo de políticas de centro de alojamiento contiene las políticas siguientes:

- AllUsersExecuteChannelStoreAllUsersViews
- ContractAdministratorsForChannelOrgExecuteCreate
CommandsOnMemberResource
- ContractAdministratorsForChannelOrgExecuteContract
DeployCommandsOnContractResource
- ContractAdministratorsForChannelOrgDisplayContract
DatabaseanResourceGroup



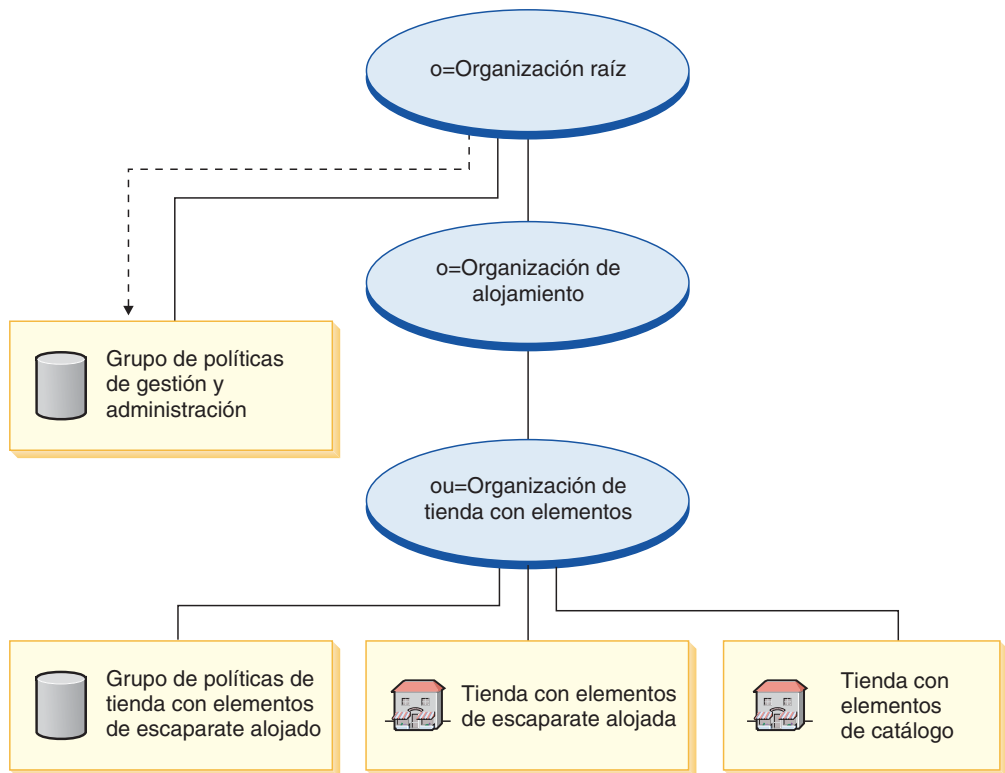
Leyenda



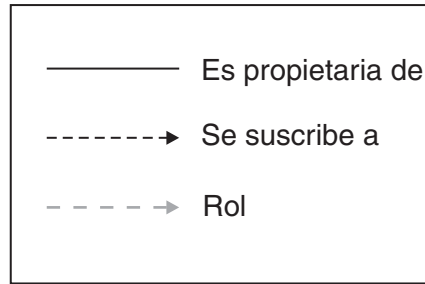
La organización de directorio de tiendas se suscribe directamente al grupo de políticas de gestión y administración y es propietaria y se suscribe al grupo de políticas de directorio de tiendas. Como resultado, estas políticas se aplican al directorio de tiendas, que está directamente bajo la organización de directorio de tiendas.

El grupo de políticas de directorio de tiendas contiene la política siguiente:

- AllUsersExecutePublicStoreAllUsersViews

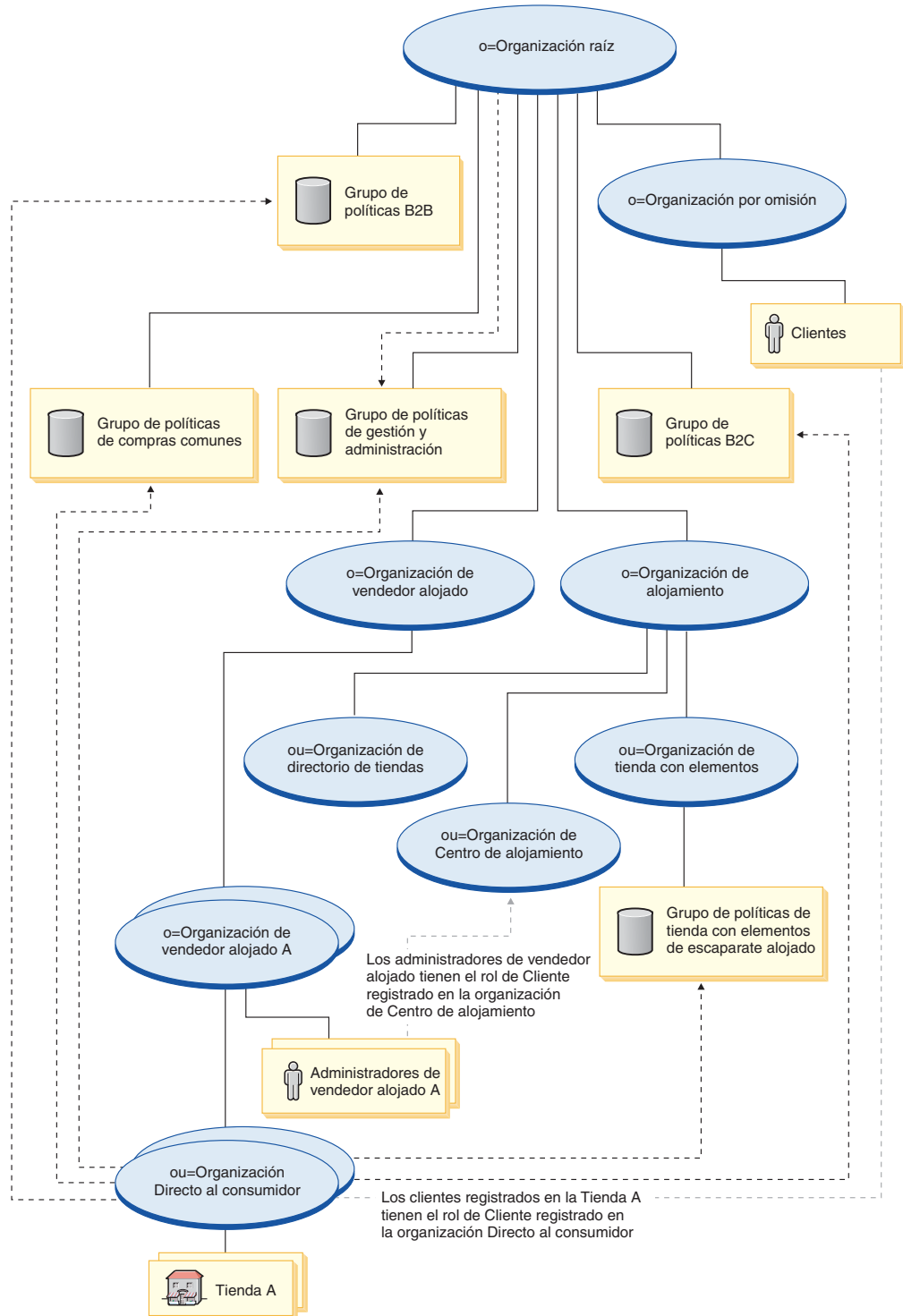


Leyenda

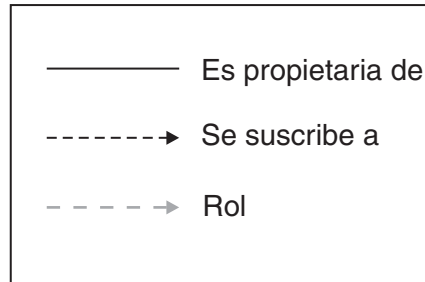


La organización de tienda con elementos no se suscribe directamente a ningún grupo de políticas. Como consecuencia, hereda el grupo de políticas de gestión y administración de la organización raíz. Estas políticas se aplican a la organización de tienda con elementos y a las tiendas con elementos de las que es propietaria. La organización de tienda con elementos es propietaria del grupo de políticas de tienda con elementos de escaparate alojada, pero no se suscribe a él.

Nota: Las organizaciones de vendedor alojado individuales se suscribirán al grupo de políticas de tienda con elementos de escaparate alojada cuando se cree la tienda alojada.



Leyenda



La organización de vendedor alojado no se suscribe directamente a ningún grupo de políticas. Como consecuencia, hereda el grupo de políticas de gestión y administración de la organización raíz. Estas políticas se aplican a la organización de vendedor alojado y a las organizaciones del vendedor alojado A de las que es propietaria, así como a los administradores del vendedor alojado A.

La organización Directo al consumidor se suscribe directamente a los grupos de políticas de gestión y administración, compras comunes, B2B y B2C, así como al grupo de políticas de tienda con elementos de escaparate alojada. Estas políticas se aplican a todas las tiendas que son propiedad de la organización Directo al consumidor.

Control de acceso en negocios de ejemplo

Cada uno de los negocios de ejemplo de WebSphere Commerce incluye la infraestructura de control de acceso. Para obtener información detallada acerca de cómo se implementa la infraestructura de control de acceso en estos negocios, consulte la publicación *WebSphere Commerce, Guía de tiendas de ejemplo*.

Adición de control de acceso a las tiendas

Para obtener más información sobre cómo añadir control de acceso a las tiendas, consulte el Capítulo 33, "Control de acceso en la tienda", en la página 311.

Capítulo 5. Infraestructura de políticas de negocio de WebSphere Commerce

Las políticas de negocio son conjuntos de normas seguidas por una tienda o grupo de tiendas que definen los procesos de negocio, las prácticas del sector y el ámbito y las características de las ofertas de una tienda o grupo de tiendas y el modo en que la tienda o sitio interactúa con los clientes u otros business partners. Por ejemplo, es posible que el sitio tenga políticas de negocio que determinen cuándo y cómo pueden los clientes devolver productos a una tienda, o políticas de negocio que determinen los métodos de pago que acepta la tienda.

Descripción de la infraestructura de políticas de negocio de WebSphere Commerce

WebSphere Commerce proporciona una infraestructura que le permite implementar las políticas de negocio del sitio o tienda en línea. La infraestructura de políticas de negocio consta de los siguientes componentes:

- Políticas de negocio
- Contratos y acuerdos de servicio
- Términos y condiciones
- Cuentas de negocio

Políticas de negocio

En la mayor parte de los casos, tendrá políticas de negocio predefinidas para su negocio que debe implementar en su tienda en línea o sitio. WebSphere Commerce proporciona un conjunto de políticas de negocio que puede utilizar como están, o puede cambiarlas para que se ajusten a sus necesidades. Para obtener más información acerca de las políticas de negocio por omisión que se proporcionan con WebSphere Commerce, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce. Para obtener información sobre cómo editar estas políticas de negocio, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

Cuentas de negocio

Las cuentas de negocio definen la relación entre un cliente y el negocio. Las cuentas de negocio realizan un seguimiento de los contratos y pedidos de las organizaciones de cliente y configuran el modo en que las organizaciones cliente compran en una tienda.

Contratos y acuerdos de servicio

Antes de que un cliente o business partner (por ejemplo, revendedores o distribuidores) puedan acceder a la tienda, debe crear un contrato o acuerdo de servicio. En la infraestructura de políticas de negocio de WebSphere Commerce, los contratos se crean para los clientes y los acuerdos de servicio para otros tipos de business partners.

- **Contratos:** Un contrato con un cliente define a qué áreas de la tienda puede acceder el cliente, los precios que verá el cliente y durante cuánto tiempo puede acceder el cliente al sitio y los precios. Todas las tiendas deben contener como mínimo un contrato, ya que sin contrato nadie que no sea el administrador

interno puede acceder a la tienda. WebSphere Commerce proporciona un contrato por omisión que se aplica a todos los clientes que compran en una tienda. En WebSphere Commerce Professional Edition, el contrato por omisión es el único contrato soportado.

- **Acuerdos de servicio:** Un acuerdo de servicio con un business partner (los business partners pueden ser revendedores, distribuidores, fabricantes, proveedores u otros socios comerciales) define el acuerdo alcanzado con el business partner. Por ejemplo, un acuerdo de servicio con un revendedor puede definir qué acceso tiene el revendedor a su sitio, si puede compartir el catálogo o si aloja una tienda para el mismo. Un acuerdo de servicio con un distribuidor puede definir el modo en que los clientes del sitio pueden recibir presupuestos de un distribuidor o el modo en que pueden acceder al sitio del distribuidor desde el suyo.

Términos y condiciones

Los términos y condiciones definen el modo en que se implementan los contratos y los acuerdos de servicio de un cliente o business partner determinado. En los contratos, los términos y condiciones pueden definir lo que se está vendiendo bajo el contrato, el precio de los artículos vendidos, el modo en que se envían los artículos al cliente o cómo pagará el pedido el cliente. En los acuerdos de servicio con business partners, los términos y condiciones pueden limitar los productos que puede vender el business partner.

Los términos y condiciones suelen hacer referencia a políticas de negocio ya que muchos de los aspectos de un sitio o muchas de las operaciones de las tiendas se definen mediante políticas de negocio. Los términos y condiciones proporcionan parámetros para las políticas de negocio a las que hacen referencia. El suministro de parámetros a las políticas de negocio le permite modificar el comportamiento de esas políticas para cada contrato.

Políticas de negocio en negocios de ejemplo

Cada uno de los negocios de ejemplo de WebSphere Commerce incluye la infraestructura de políticas de negocio. Para obtener información detallada acerca de cómo se implementa la infraestructura de políticas de negocio, consulte la publicación *WebSphere Commerce, Guía de tiendas de ejemplo*

Adición de políticas de negocio al sitio

Para obtener más información acerca de cómo implementar la infraestructura de políticas de negocio en el sitio, consulte el Capítulo 18, "Elementos de contrato", en la página 195.

Capítulo 6. Arquitectura de instancias

Este capítulo proporciona una introducción a la arquitectura de instancias de WebSphere Commerce Server.

WebSphere Commerce Server

WebSphere Commerce Server es una aplicación de WebSphere Application Server que maneja las funciones relacionadas con la tienda y el comercio de una solución de comercio electrónico. Los elementos de escaparate y la lógica de negocio residen en una aplicación Web incluida en WebSphere Commerce Server. WebSphere Commerce proporciona una aplicación Web por omisión (`Stores.war`), o puede crear la suya propia.

Una aplicación Web puede contener los elementos para una tienda o para varias tiendas. Cuando una aplicación Web contiene varios escaparates y lógica de negocio, los elementos de cada tienda se separan en directorios de tienda (`dirienda`).

Instancia de WebSphere Commerce Server

Una instancia de WebSphere Commerce Server es una aplicación WebSphere Application Server desplegada con una base de datos asociada. Una instancia puede dar soporte a varias tiendas. Todas las tiendas de una instancia comparten la misma base de datos y pueden compartir algunos tipos de datos, por ejemplo, el catálogo, el despacho de pedidos o las recepciones. Todas las tiendas en una instancia también comparten el mismo contenedor EJB.

Puede crear una sola tienda en una instancia o puede crear varias tiendas en una instancia. Para obtener más información sobre varias tiendas de una instancia, consulte el apartado "Varias tiendas en una sola instancia" en la página 68.

Capítulo 7. Arquitectura de la tienda

Para dar soporte a la creación de tiendas en línea, WebSphere Commerce proporciona una arquitectura de tienda. En este capítulo se describe esta arquitectura junto con algunos de los ejemplos de tienda que se pueden implementar utilizándola.

Descripción de la arquitectura de tienda de WebSphere Commerce

Para dar soporte a las tiendas de su sitio, WebSphere Commerce proporciona un archivador de tienda que le permite crear tiendas en línea. La arquitectura de la tienda consta de los siguientes componentes:

- Elementos de tienda
- Soporte de varias tiendas en una sola instancia
- Relaciones entre tiendas

Elementos de tienda

En WebSphere Commerce, una tienda en línea es el lugar donde se llevan a cabo todas las transacciones de su negocio en línea. Todas las tiendas en línea creadas con WebSphere Commerce incluyen como mínimo uno de los siguientes tipos de elementos:

- Escaparate: La parte externa de la tienda, o la parte que se muestra a los clientes, se conoce como escaparate. El escaparate consta de elementos Web tales como páginas HTML, archivos JSP, hojas de estilo, imágenes, gráficos y otros tipos de archivos multimedia. Esta guía describe los conceptos y las tareas necesarias para crear los archivos JSP que generan las páginas de la tienda. Para obtener más información, consulte la Parte 4, “Desarrollo del escaparate”, en la página 79.
- Lógica de negocio: La parte de la tienda que procesa peticiones de cliente, incluidos los mandatos y el código personalizado se conoce como lógica de negocio. Para obtener información más detallada sobre la creación de lógica de negocio o código personalizado, consulte la publicación *WebSphere Commerce, Guías de programación y aprendizaje*.
- Datos de tienda: Los elementos de datos que forman la tienda. Para que una tienda funcione correctamente debe tener los datos en una ubicación que dé soporte a todas las actividades de los clientes. Por ejemplo, para que un cliente efectúe una compra, la tienda debe contener un catálogo de los productos en venta y debe incluir un proceso para el manejo de pedidos, el inventario para satisfacer el pedido y un proceso de envío. También debe tener métodos para procesar y recoger los pagos. Los conceptos y las tareas involucradas en la creación de datos de tienda se tratan en la Parte 6, “Desarrollo de los datos de la tienda”, en la página 117.

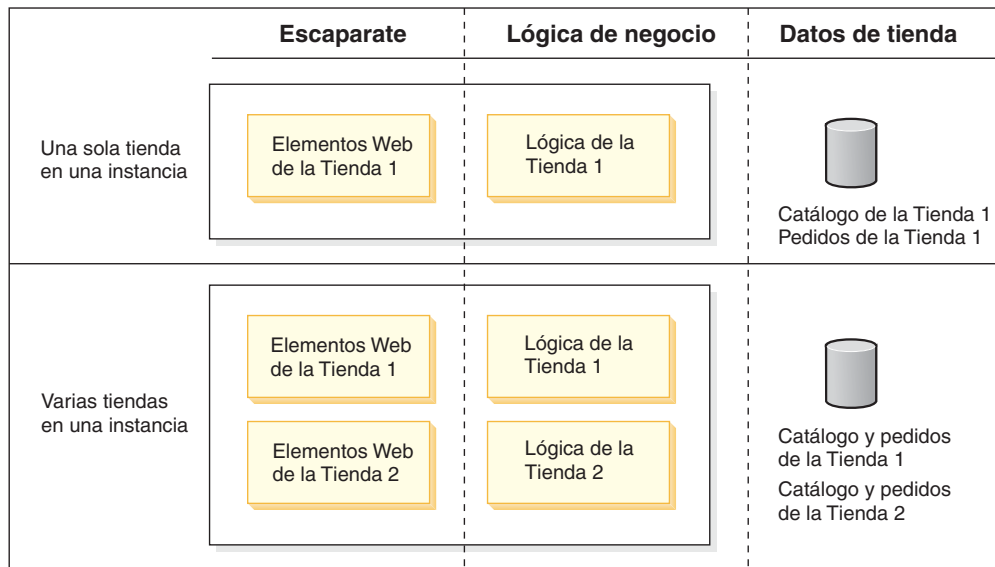
Si una tienda contiene los tres tipos de elementos, esto es, los elementos de escaparate, la lógica de negocio y los datos de tienda, es una tienda totalmente operativa. En WebSphere Commerce si una tienda contiene únicamente un subconjunto de los elementos, es decir, contiene los elementos de escaparate y la lógica de negocio, los datos de tienda y la lógica de negocio o simplemente los datos de tienda, se denomina una *tienda con elementos*.

Tiendas con elementos

Las tiendas con elementos son grupos de recursos que se pueden compartir (elementos de negocio, procesos de negocio y elementos de escaparate) de los que pueden beneficiarse otras tiendas. Por ejemplo, en lugar de crear un catálogo como parte de una tienda de centro, ésta puede beneficiarse de una tienda con elementos de catálogo que otros canales del centro o asociados pueden compartir también. Una tienda con elementos suele constar de los elementos que se pueden utilizar en varias tiendas. Para obtener más información, consulte el apartado “Relaciones entre tiendas” en la página 70.

Varias tiendas en una sola instancia

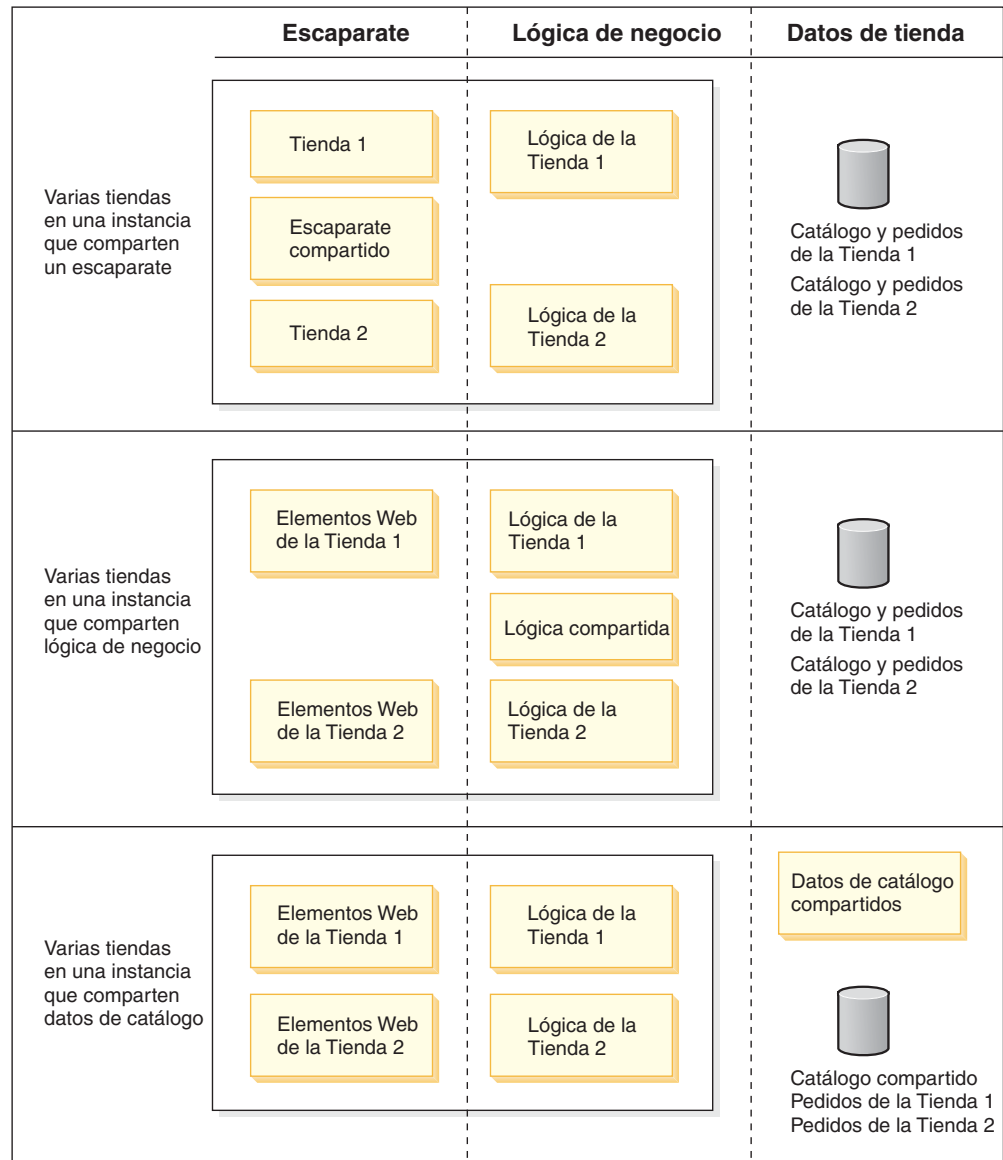
WebSphere Commerce le permite dar soporte a varias tiendas en línea en su instancia de WebSphere Commerce Server. El siguiente diagrama ilustra algunas configuraciones de tienda posibles:



Las tiendas descritas en el diagrama anterior son tiendas autónomas. Es decir, aunque están en la misma instancia, no comparten los datos ni tienen relaciones entre sí. Tienen escaparates, lógica de negocio y datos de tienda diferentes.

Asimismo, en una instancia puede crear varias tiendas que compartan el mismo escaparate, la misma lógica de negocio o los mismos datos de tienda, incluidos los catálogos, o cualquier combinación de los tres. El siguiente diagrama muestra algunas de las configuraciones posibles en los que las tiendas comparten los

elementos:



Nota: El diagrama anterior solamente lista algunas de las configuraciones posibles entre varias tiendas de una instancia. Las tiendas pueden compartir más de un tipo de elemento, por ejemplo, varias tiendas de un sitio pueden compartir los escaparates, la lógica de negocio o los datos de tienda, o cualquier combinación de los tres.

Para obtener más información acerca de cómo varias tiendas de una instancia comparten elementos de tienda comunes, consulte el apartado “Relaciones entre tiendas” en la página 70.

Pueden existir varias tiendas en un sólo módulo Web Tiendas. Si es así, los elementos de tienda se separan utilizando los métodos siguientes:

- Elementos de escaparate: Los elementos de escaparate de cada tienda del módulo Web Tienda se almacenan en un directorio de tienda diferente (*dir_tienda*). Por ejemplo, todos los elementos de escaparate de la tienda MiTienda están en el directorio MiTienda.

- Lógica de negocio: El ID de tienda se utiliza para seleccionar la implementación del mandato para cada tienda, como se ha especificado en el registro de mandatos.
- Datos de tienda: Los elementos de tienda se identifican para cada tienda mediante un índice exclusivo.

Relaciones entre tiendas

Para poder dar soporte a que varias tiendas de un sitio tengan el mismo escaparate, lógica de negocio o datos de tienda o cualquier combinación de los elementos compartidos, y también para dar soporte a otros tipos de relaciones entre las tiendas de un sitio como, por ejemplo, que una tienda aloje a otra o transferir carros de la compra de una tienda a otra, WebSphere Commerce proporciona ahora la arquitectura necesaria para diferentes relaciones entre tiendas.

Las relaciones entre tiendas permiten que una tienda proporcione un servicio a otra tienda. Por ejemplo, la tienda A puede alojar a la tienda B, o la tienda C puede utilizar los datos de catálogo de la tienda D.

Para poder implementar estas relaciones entre tiendas, es necesario código que dé soporte a cada relación entre tiendas. WebSphere Commerce incluye muchas relaciones entre tiendas y el código de soporte. Estas relaciones entre tiendas se pueden agrupar aproximadamente en las categorías siguientes:

- Relaciones en las que una tienda proporciona elementos a otra tienda. Estos tipos de relaciones de tienda incluyen una tienda que proporciona los URL, mandatos, políticas de negocio, archivos de propiedades y monedas a otra.
- Relaciones en las que una tienda tiene una "relación de negocio" con otra tienda. Estos tipos de relaciones entre tiendas incluyen una relación en la que una tienda aloja a otra o una tienda envía los pedidos a otra tienda.

Nota: Para obtener una lista detallada de las relaciones entre tiendas por omisión que se proporcionan con WebSphere Commerce, consulte el Capítulo 14, "Relaciones entre tiendas", en la página 143.

Descripción del soporte de modelos de negocio en la arquitectura de tienda

Para dar soporte a las tiendas necesarias para los modelos de negocio, WebSphere Commerce utiliza la arquitectura de tienda para crear los tipos de tiendas siguientes:

- Tiendas cara al público
- Tiendas proxy
- Tiendas con elementos

Nota: Estas tiendas en concreto se recomiendan para implementar los modelos de negocio a los que da soporte WebSphere Commerce. Asimismo puede crear sus propios tipos de tiendas utilizando la arquitectura de tienda.

Tiendas cara al público

Las tiendas cara al público son las tiendas a las que los clientes pueden acceder directamente. Estas tiendas son los componentes principales del sitio. WebSphere Commerce da soporte a los siguientes tipos de tiendas cara al público:

- **Tienda de venta directa:** Una tienda que da soporte a transacciones comerciales de productos, servicios o información directamente entre empresas y clientes o entre dos empresas o partes. WebSphere Commerce da soporte a los siguientes tipos de tiendas de venta directa:
 - Directo al consumidor
 - Directo a B2B
- **Tienda de Centro:** Una tienda que permite a sus clientes o asociados acceder a productos o servicios disponibles de uno o varios asociados o clientes del propietario del Centro, a través de las demás tiendas del sitio.
- **Tienda alojada:** Tienda que aloja el operador del sitio para el propietario de la tienda. El propietario de la tienda puede tener la opción de administrar la tienda.

Creación de tiendas de centro y venta directa

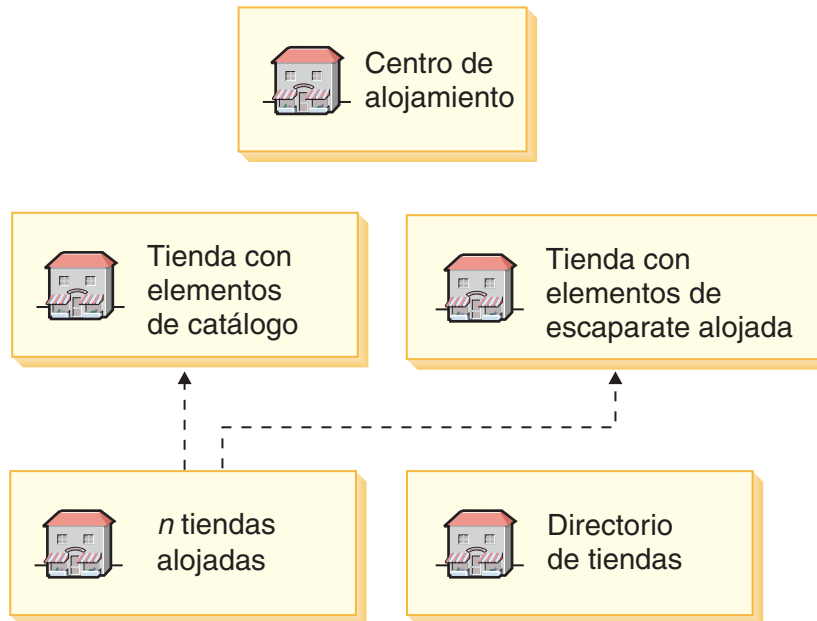
Las tiendas de centro y venta directa son las tiendas más tradicionales en WebSphere Commerce en lo que se refiere al proceso de creación de tiendas. Es decir, debe crear los elementos de escaparate, la lógica de negocio y los datos de tienda de cada tienda. Tiene la opción de crear estos elementos de forma tradicional, esto es, puede crear los elementos únicamente para dicha tienda. No obstante, también puede crear elementos para que los utilicen otras tiendas, esto es, puede crear los elementos de escaparate y lógica de negocio en una tienda con elementos o datos que pueden compartir las diferentes tiendas. O es posible que desee utilizar elementos de otras tiendas para crear partes de la tienda de centro o venta directa.

Para obtener más información sobre cómo crear elementos de escaparate, consulte la Parte 4, “Desarrollo del escaparate”, en la página 79. Para obtener más información sobre la creación de lógica de negocio o código personalizado, consulte la publicación *WebSphere Commerce, Guías de programación y aprendizaje*. Para obtener más información sobre cómo crear datos de tienda, consulte la Parte 6, “Desarrollo de los datos de la tienda”, en la página 117. Para obtener más información sobre cómo compartir elementos entre tiendas, consulte el Capítulo 14, “Relaciones entre tiendas”, en la página 143.

Creación de tiendas alojadas

En los ejemplos que se proporcionan con WebSphere Commerce, la mayor parte de la tienda alojada se crea compartiendo elementos de las tiendas con elementos existentes. Por ejemplo, en lugar de crear los elementos de escaparate o de catálogo para cada tienda que esté alojando, utilice el escaparate y, dependiendo del negocio, el catálogo de otra tienda. Para facilitar la creación de tiendas alojadas, WebSphere Commerce utiliza las tiendas con elementos. El diagrama siguiente ilustra cómo utilizan las tiendas alojadas los elementos de la tienda con elementos

de escaparate alojada y de la tienda con elementos de catálogo.



Los administradores de negocios alojados pueden luego realizar cambios externos (por ejemplo, cambios de diseño, incluir su propio logotipo y especificar su propio texto) para personalizar la tienda y también pueden cambiar determinados datos (filtrar el catálogo, cambiar precios, etcétera).

Asimismo puede crear tiendas alojadas de forma tradicional, es decir, puede crear los elementos de escaparate, la lógica de negocio y los datos de tienda por separado para cada tienda alojada. Para obtener más información sobre cómo crear elementos de escaparate, consulte la Parte 4, "Desarrollo del escaparate", en la página 79. Para obtener más información sobre la creación de lógica de negocio o código personalizado, consulte la publicación *WebSphere Commerce, Guías de programación y aprendizaje*. Para obtener más información sobre cómo crear datos de tienda, consulte la Parte 6, "Desarrollo de los datos de la tienda", en la página 117.

Asistente de creación de tiendas: El Asistente de creación de tiendas que se proporciona con WebSphere Commerce le permite crear tiendas alojadas rápida y fácilmente. El asistente le pide al cliente que proporcione algunos datos básicos acerca de la tienda (el nombre, la descripción, etc.), permite al cliente seleccionar el escaparate o el catálogo que desea utilizar y luego crea la tienda para él. La tienda resultante tiene algunos datos exclusivos (datos básicos de tienda que la convierten en una tienda exclusiva) pero utiliza los datos de escaparate y de catálogo de las tiendas con elementos existentes.

El comportamiento del Asistente de creación de tiendas se rige mediante una plantilla que determina las opciones que están disponibles para crear la tienda alojada, incluidas las relaciones entre tiendas, las modalidades de envío, los mensajes y el centro de formalización compartido. WebSphere Commerce proporciona varias plantillas para el Asistente de creación de tiendas, uno para cada modelo de negocio soportado. Estas plantillas se encuentran en el directorio siguiente:

```
dir_instalación_WC/xml/trading/xml
```

Se asocia una plantilla al Asistente de creación de tiendas basándose en el tipo de tienda con elementos de escaparate seleccionado en el asistente. Por ejemplo, si selecciona que se utilicen los elementos de la tienda con elementos de escaparate de revendedor, (que se identifica como RPS en el campo STORETYPE de la tabla STORE), el Asistente de creación de tiendas utiliza el archivo TemplateHostingContractRPS.xml.

Para obtener información sobre cómo crear una tienda alojada utilizando el Asistente de creación de tiendas, consulte la Ayuda en línea a la producción de WebSphere Commerce.

Nota: Si prefiere no utilizar el Asistente de creación de tiendas para crear tiendas alojadas, puede crear un acuerdo de servicio basado en una de las plantillas y luego importarlo a WebSphere Commerce. Para obtener información, consulte la Ayuda en línea a la producción de WebSphere Commerce.

Para cambiar los elementos que comparte la tienda alojada, debe cambiar la tienda con elementos. Para obtener más información, consulte el apartado “Creación de tiendas con elementos” en la página 74.

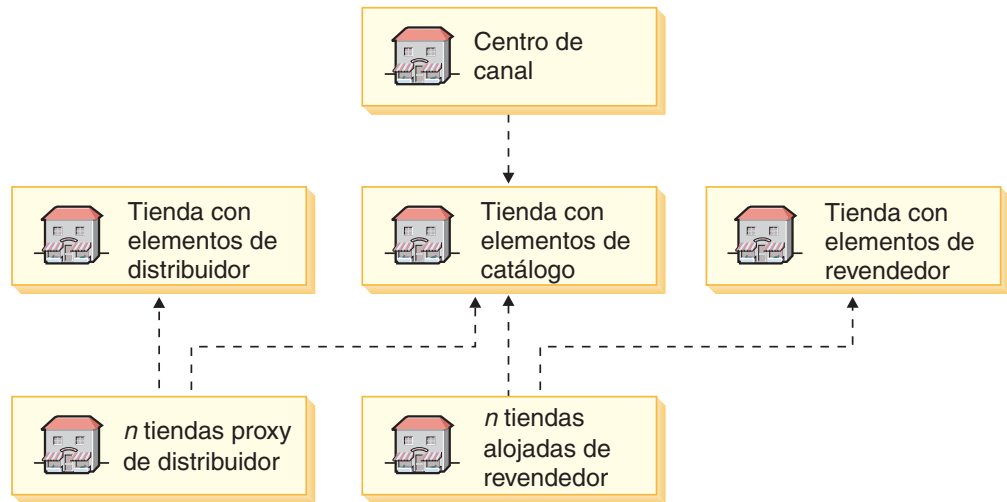
Tiendas proxy

WebSphere Commerce también da soporte a entidades conocidas como tiendas proxy. Una tienda proxy es una tienda que representa los elementos operativos de un asociado, proporciona la lógica de negocio que permite al sitio de WebSphere Commerce interactuar con un asociado externo. Por ejemplo, una tienda proxy puede capturar los pedidos transferidos a un sistema de captura de pedidos remoto, así como capturar la información de inventario de los proveedores o la información enviada a los centros de despacho de pedidos de un proveedor. A diferencia de una tienda cara al público, una tienda proxy no incluye ningún escaparate y los usuarios no pueden acceder a ella.

Creación de tiendas proxy

Crear una tienda proxy es muy similar a crear tiendas alojadas, ya que la mayor parte de los elementos de tienda del proxy se proporcionan a partir de tiendas existentes (incluidas tiendas con elementos). Tal y como se implementa en los ejemplos proporcionado con WebSphere Commerce, la tienda proxy no incluye un escaparate. Como resultado, solamente se comparten los elementos del catálogo de otra tienda. El diagrama siguiente ilustra las tiendas proxy de distribuidor utilizando los elementos de la tienda con elementos de distribuidor y de la tienda

con elementos de catálogo.



En lugar de proporcionar una interfaz de usuario para crear una tienda proxy, WebSphere Commerce implementa tiendas proxy mediante acuerdos de servicio que, a continuación, se importan a WebSphere Commerce, creando la tienda proxy. El acuerdo de servicio se rige mediante una plantilla, que determina la información que tiene que crear. La plantilla para crear tiendas proxy (TemplateReferralContract.xml) está disponible en el directorio siguiente:

dir_instalación_WC/xml/trading/xml

Para crear la tienda proxy, cree un acuerdo de servicio nuevo siguiendo la plantilla y luego impórtelo a WebSphere Commerce. Para obtener más información, consulte el Ayuda en línea a la producción de WebSphere Commerce.

Tiendas con elementos

Para facilitar la creación de tiendas cara al público y tiendas proxy, WebSphere Commerce implementa las tiendas con elementos. Las tiendas con elementos son grupos de recursos que se pueden compartir (elementos de negocio, procesos de negocio y elementos de escaparate) de los que pueden beneficiarse otras tiendas. Por ejemplo, en lugar de crear un catálogo como parte de una tienda de centro, ésta puede beneficiarse de una tienda con elementos de catálogo que otros canales del centro o asociados pueden compartir también. Una tienda con elementos suele constar de los elementos que se pueden utilizar en varias tiendas. Para obtener más información, consulte el apartado “Relaciones entre tiendas” en la página 70.

WebSphere Commerce proporciona tiendas con elementos de catálogo y tiendas con elementos de escaparate de ejemplo.

Creación de tiendas con elementos

Las tiendas con elementos son tiendas que proporcionan elementos a otra tienda. Tal y como se implementan para los ejemplos proporcionados con WebSphere Commerce, las tiendas con elementos constan de un grupo de elementos pero no son tiendas totalmente operativas. Para crear una tienda con elementos, debe seguir los mismos métodos que para crear los elementos en una tienda de Centro o de venta directa. Esto es, si desea que la tienda con elementos contenga elementos de catálogo, cree los datos de catálogo siguiendo las instrucciones de la Parte 6, “Desarrollo de los datos de la tienda”, en la página 117. Si la tienda con elementos contendrá elementos de escaparate, consulte la Parte 4, “Desarrollo del escaparate”, en la página 79

en la página 79. Si la tienda con elementos contendrá lógica de negocio, consulte la publicación *WebSphere Commerce, Guías de programación y aprendizaje*.

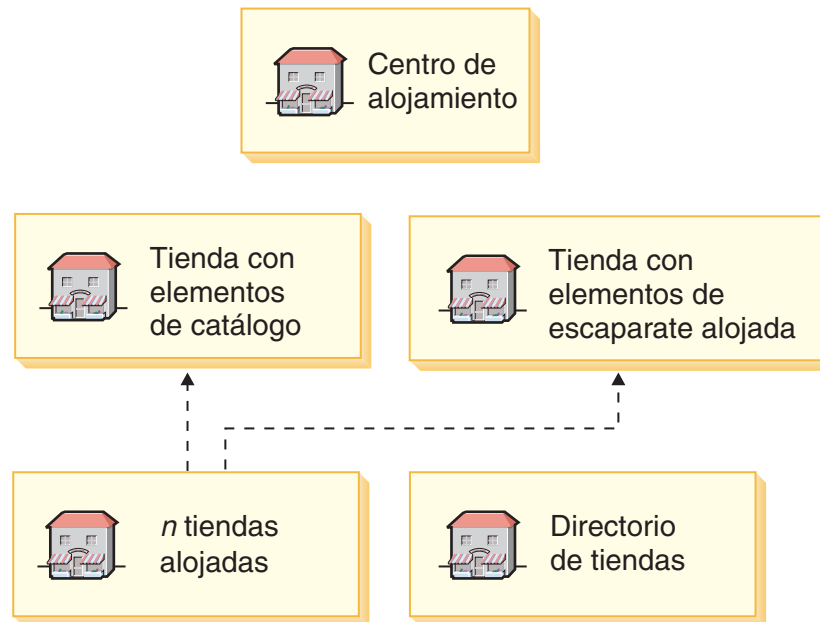
Tiendas de los modelos de negocio soportados

Los apartados siguientes ilustran cómo se implementan las tiendas en los negocios de ejemplo.

Nota: Dado que los ejemplos Directo al consumidor y Directo a B2B contienen cada uno una tienda de venta directa, no se describen aquí.

Alojamiento

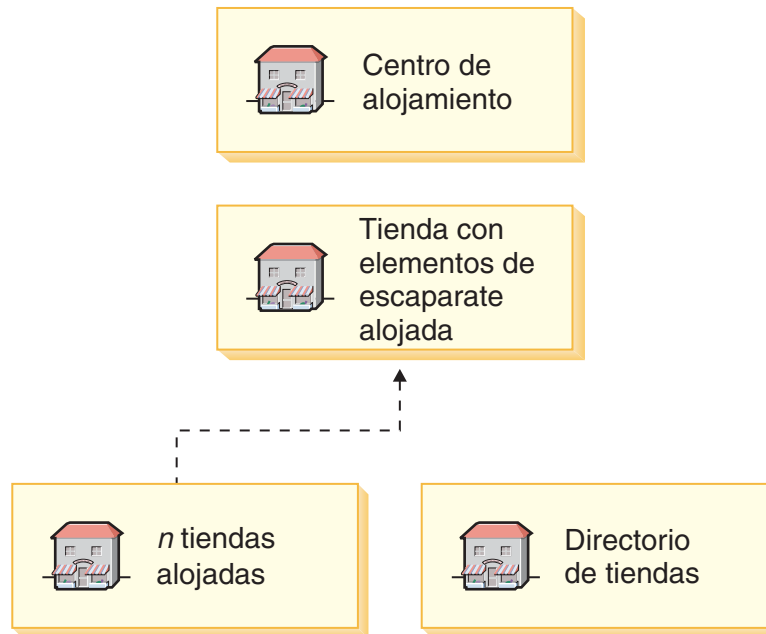
El diagrama siguiente ilustra los tipos de tiendas que componen el ejemplo de alojamiento.



El sitio de alojamiento de ejemplo contiene una tienda de centro (centro de alojamiento), dos tiendas con elementos (la tienda con elementos de catálogo y la tienda con elementos de escaparate alojada) y también el directorio de tiendas. El directorio de tiendas es un listado de todas las tiendas alojadas en el sitio y actúa como una puerta de acceso a las mismas. Las tiendas de alojamiento se crean utilizando los elementos de las dos tiendas con elementos.

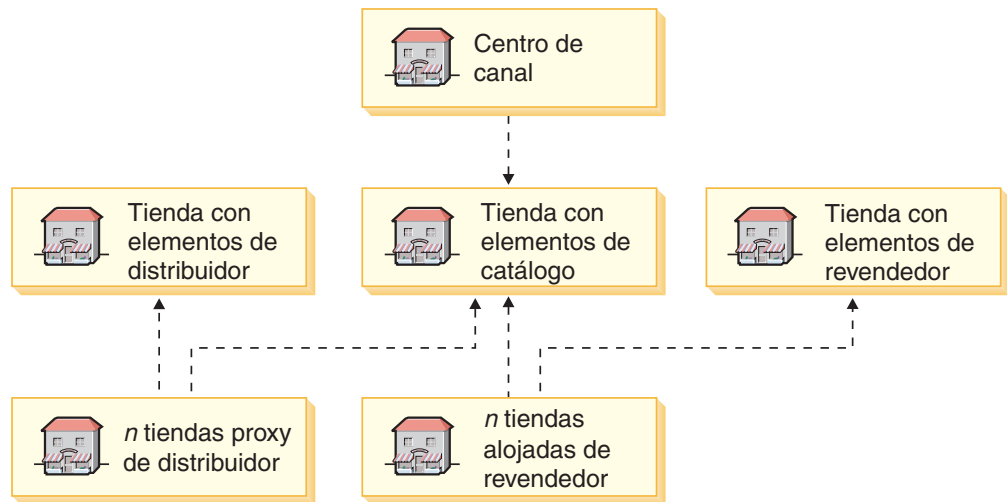
Tenga en cuenta que los clientes pueden optar por crear sus propios datos de catálogo, en lugar de utilizar el catálogo definido en una tienda con elementos de catálogo. Esta variación crea una segunda implementación del sitio de alojamiento,

como se ilustra en el diagrama siguiente:



Cadena de demanda

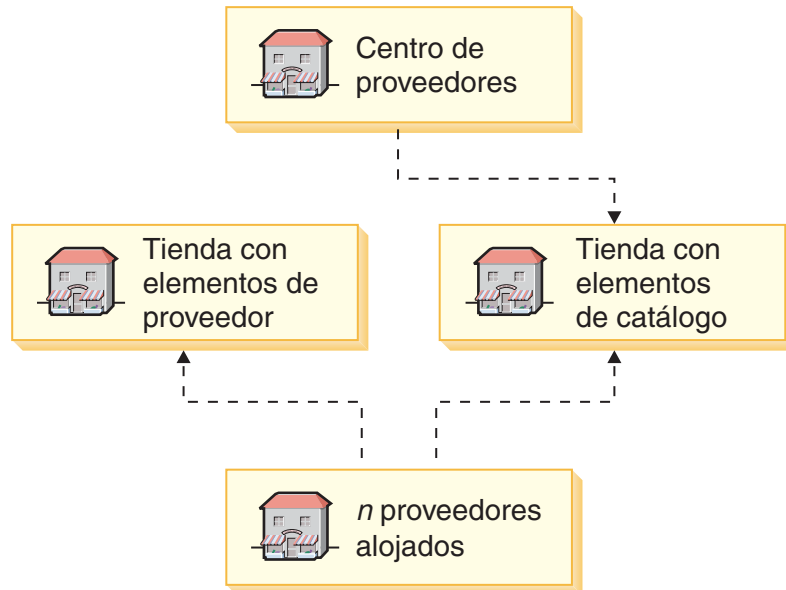
El diagrama siguiente ilustra los tipos de tiendas que componen el ejemplo de cadena de demanda.



El sitio de ejemplo de cadena de demanda contiene una tienda de centro (centro de canal) y tres tiendas con elementos (la tienda con elementos de distribuidor, la tienda con elementos de catálogo y la tienda con elementos de escaparate de revendedor). Tenga en cuenta que el centro de canal utiliza los elementos de catálogo definidos en la tienda con elementos de catálogo. Las tiendas proxy de distribuidor se crean utilizando los elementos de la tienda con elementos de distribuidor, mientras que las tiendas alojadas de revendedor se crean utilizando los elementos de la tienda con elementos de catálogo y de la tienda con elementos de escaparate de revendedor.

Cadena de oferta

El diagrama siguiente ilustra los tipos de tiendas que componen el ejemplo de cadena de oferta.



El sitio de ejemplo de cadena de oferta contiene una tienda de centro (centro de proveedores) y dos tiendas con elementos (la tienda con elementos de catálogo y la tienda con elementos de proveedor). Tenga en cuenta que el centro de proveedores utiliza los elementos definidos en la tienda con elementos de catálogo. Los proveedores alojados se crean utilizando los elementos de la tienda con elementos de catálogo y de la tienda con elementos de proveedor.

Nota: El propietario del centro de proveedores define la taxonomía del catálogo (por ejemplo, la estructura de categorías y los productos y artículos que se pueden compartir) que los proveedores alojados utilizarán en la tienda con elementos de catálogo.

Parte 4. Desarrollo del escaparate

Capítulo 8. Desarrollo del escaparate

Este capítulo proporciona una visión general de la arquitectura del escaparate de WebSphere Commerce, incluido el modo en que se muestra a los clientes la parte externa de la tienda, los elementos Web como, por ejemplo, las páginas HTML, los archivos JSP, las hojas de estilo, las imágenes, los gráficos y otros tipos de archivos multimedia.

Arquitectura del escaparate

WebSphere Commerce utiliza un sistema de *mandatos* y *vistas* para mostrar a los clientes los elementos Web de un escaparate.

- Los *mandatos* efectúan un proceso de negocio específico como, por ejemplo, añadir un producto al carro de la compra, procesar un pedido, actualizar el listín de un cliente o mostrar una página de producto específica. Cuando la acción finaliza, el mandato devuelve una vista.
- Las *vistas* muestran el resultado de los mandatos y las acciones de usuario, es decir, presentan las páginas de la tienda (archivos JSP) a los clientes. Para que la vista invoque un archivo JSP, el nombre del archivo JSP debe estar registrado con la vista en la tabla de registro de vistas (VIEWREG). El archivo JSP correspondiente se almacena utilizando el nombre de archivo JSP del subdirectorio (*dir_tienda*) de la tienda bajo la aplicación Web de tiendas de WebSphere Commerce.

Tanto a los mandatos como a las vistas se les invoca utilizando los URL. Por ejemplo, cuando un cliente pulsa **Carro de la compra** en la tienda de ejemplo, el cliente invoca el URL `https://nombresistpral/vía/OrderItemDisplay?`, que se pasa a WebSphere Commerce Server. Éste llama al mandato `OrderItemDisplay` y la página de carro de la compra se muestra al cliente.

Cuando un cliente pulsa **Ayuda** en la tienda de ejemplo, el cliente invoca el URL `https://nombresistpral/vía/HelpView?`, que se pasa a WebSphere Commerce Server. Éste llama a `HelpView`, que devuelve la página de ayuda.

WebSphere Commerce Server también puede correlacionar varios mandatos con un URL, lo cual permite que cada tienda tenga, opcionalmente, su propia implementación de ese mandato.

De forma parecida, WebSphere Commerce Server también le permite correlacionar varios archivos JSP con una sola vista, en la que cada tienda puede, opcionalmente, registrar distintos nombres de archivo JSP para distintos tipos de dispositivo.

Mandatos y vistas por omisión

WebSphere Commerce proporciona mandatos y vistas por omisión que puede utilizar en la tienda. Encontrará una lista de estos mandatos y vistas en el archivo `wcs.bootstrap.xml`. Los archivos de rutina de carga se encuentran en el directorio siguiente:

- `dir_instalación_WC/schema/xml`

Si necesita un mandato o una vista que no se proporciona, puede crearlo. Para obtener información acerca de cómo crear mandatos y vistas, consulte la publicación *WebSphere Commerce, Guías de programación y aprendizaje*.

Creación de las páginas de tienda

La tarea más extensa cuando se crea el escaparate es crear las páginas de la tienda. Antes de empezar el trabajo de desarrollo en las páginas de la tienda, debe completar las actividades planificadas siguientes:

- Desarrollar una lista de las páginas de tienda que se necesitan
- Desarrollar una lista de los URL de mandatos y vistas
- Asociar nombres de archivo JSP a las vistas

Nota: Mientras planifica las páginas de tienda, también debe crear una estrategia de almacenamiento en antememoria. Para obtener más información sobre el almacenamiento en antememoria, consulte el Capítulo 9, “Almacenamiento en antememoria de las páginas de tienda”, en la página 89.

Desarrollar una lista de las páginas de la tienda

Para desarrollar una lista de las páginas necesarias para crear la tienda, debe conocer los requisitos funcionales y de negocio de la tienda, así como los procesos de negocio que se hayan definido.

Trabajar con ejemplos de uso

Muchas personas recopilan los requisitos en forma de ejemplos de uso. Los ejemplos de uso definen los procesos de negocio de la tienda, en forma de interacciones entre el cliente y el sistema propuesto. En el caso de una tienda en línea, los ejemplos de uso pueden definir cómo se registra un cliente en la tienda, cómo examina el catálogo o cómo hace el pedido de un artículo.

En la ayuda en línea se proporciona un conjunto de ejemplos de uso, que detallan los procesos de negocio para las tiendas de ejemplo. Estos ejemplos de uso pueden ayudarle a comprender con más exactitud el flujo de las tiendas de ejemplo y se pueden utilizar como guía si desea crear ejemplos de uso para su propia tienda.

A continuación se muestra un ejemplo de uso de Registro:

Ejemplo de uso de Registro: El proceso de registro permite a los clientes entrar información personal en la base de datos.

Actor:

- Cliente

Flujo principal: El cliente selecciona **Regístrate** en el marco lateral. A continuación, el sistema muestra una página con los campos siguientes:

- Correo electrónico
- Contraseña
- Verificar contraseña
- Nombre
- Apellido
- Edad (opcional)
- Sexo (opcional)

El cliente entra la información adecuada en los campos anteriores y selecciona **Someter**. El sistema crea un nuevo cliente y guarda la información del cliente (E1, E2, E3). El sistema solicita al cliente que gestione su cuenta siguiendo el proceso que se describe en el Ejemplo de uso de Gestionar cuenta personal

Flujos alternativos: Ninguno.

Flujos de excepción: E1: La dirección de correo electrónico ya existe:

- Si la dirección de correo electrónico ya existe en el sistema, se muestra un mensaje de error solicitando al usuario que escriba otra dirección de correo electrónico. El ejemplo de uso se reanuda desde el principio.

E2: Faltan campos obligatorios:

- Si no se ha cumplimentado alguno de los campos siguientes (Dirección de correo electrónico, Contraseña, Verificar contraseña, Nombre, Apellido), el sistema emite un mensaje de error. El ejemplo de uso se reanuda desde el principio.

E3: Contraseña no válida:

- Si la contraseña no es válida o no coincide con la contraseña de verificación, el sistema emite un aviso.

Determinar el flujo de compra de la tienda: Tanto si desarrolla ejemplos de uso para ilustrar los procesos de negocio de la tienda como si usa otro método, cuando los procesos de negocio están disponibles se puede crear el flujo de compra de su tienda.

Nota: Puesto que los ejemplos de uso a menudo contienen información sobre el flujo, del tipo "Si el cliente selecciona **Someter**, se muestra la página Pedido", los ejemplos de uso pueden proporcionar información muy útil para la creación de diagramas de flujo de compra.

El flujo de compra refleja los requisitos y los procesos de negocio definidos para la tienda, que ilustran los movimientos de un cliente en la tienda. Por ejemplo, un cliente puede entrar en su sitio Web a través de la página de presentación y le puede solicitar que se registre antes de examinar el catálogo, o puede elegir que los clientes puedan ver el catálogo como invitados, sin tener que registrarse. Algunos flujos de compra permiten a los clientes pasar por una "caja rápida", mientras que otros flujos de compra requieren que el cliente realice todos los pasos de pasar por caja cada vez que efectúa una compra. O bien, su flujo de compra puede ofrecer a los clientes las dos opciones de pasar por caja.

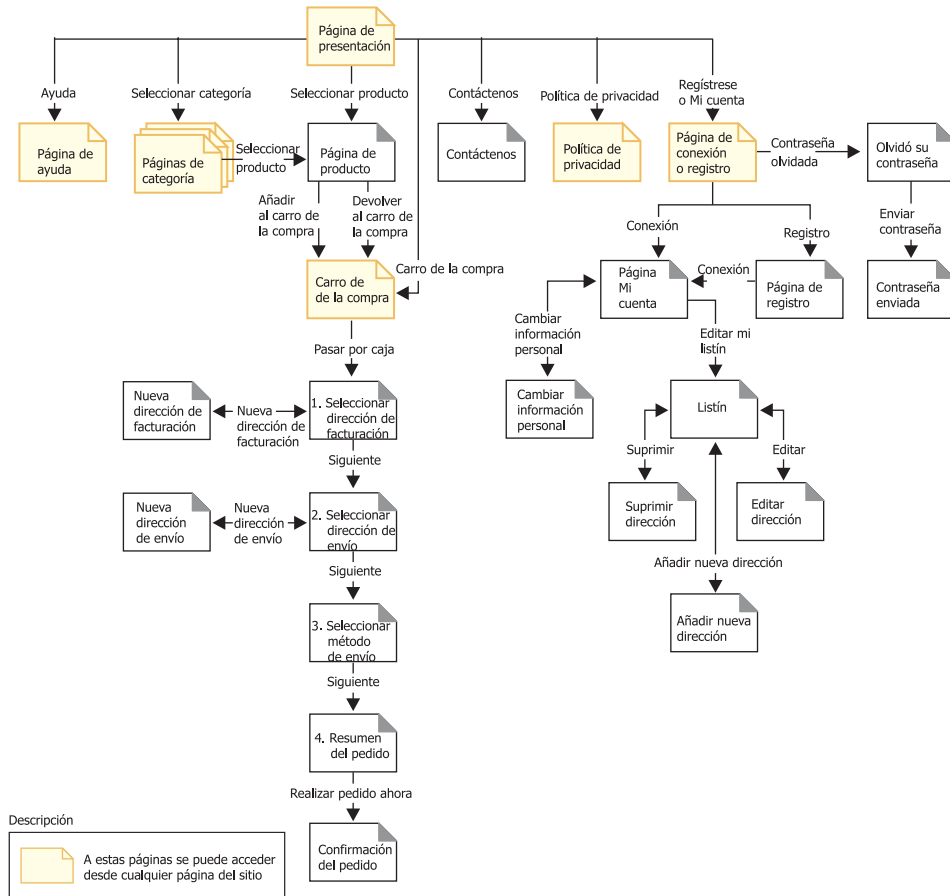


Para comprobar que el diagrama de flujo de la tienda está completo, asegúrese de que todos los pasos en los ejemplos de uso de su tienda están ilustrados en el diagrama de flujo de la tienda.

La correlación visual del flujo de compra, como se hace en el siguiente diagrama del flujo de compra de la tienda de ejemplo FashionFlow, permite ver cómo se moverán los clientes en la tienda.

Nota: Este diagrama sólo contiene una parte del flujo de la tienda FashionFlow. Para ver todo el flujo, consulte la publicación *WebSphere Commerce, Guía de*

tiendas de ejemplo.



El diagrama del flujo de compra de FashionFlow es muy sencillo. Aunque incluye el flujo principal del recorrido de un cliente por la tienda, no incluye las situaciones de error. Por ejemplo, lo que sucede en caso de que el cliente se conecte utilizando una contraseña incorrecta o escriba un número de tarjeta de crédito no válido. Sin embargo, incluso un diagrama sencillo como éste le permite desarrollar una lista de las páginas necesarias para la tienda. Para empezar, deberá crear una vista para cada página que se lista en el diagrama de flujo de compra.

Por ejemplo, si fuera a crear una tienda con el mismo flujo de compra que en el diagrama de FashionFlow, debería crear las páginas siguientes:

Nota: La tabla siguiente lista los nombres de las vistas utilizadas para la tienda FashionFlow

Páginas del diagrama de flujo de compra de la tienda FashionFlow (según las ve el cliente)	Vista correspondiente
Página de presentación	StoreCatalogDisplayView
Página de ayuda	HelpView
Contáctenos	ContactView
Política de privacidad	PrivacyView
Página de conexión o registro	LogonForm
Contraaseña olvidada	LogoffView

Páginas del diagrama de flujo de compra de la tienda FashionFlow (según las ve el cliente)	Vista correspondiente
Contraseña enviada	ResetPasswordForm
Página Mi cuenta	LogonForm
Modificar información personal	UserRegistrationForm
Listín	AddressBookForm
Añadir nueva dirección	AddressForm
Suprimir dirección	AddressBookForm
Editar dirección	AddressForm
Página de registro	UserRegistrationForm
Carro de la compra	OrderItemDisplayViewShiptoAssoc
Seleccionar dirección de facturación	BillingAddressView
Nueva dirección de facturación	AddressForm
Seleccionar dirección de envío	MultipleShippingAddressView
Nueva dirección de envío	AddressForm
Seleccionar método de envío	MultipleShippingMethodView
Resumen del pedido	AllocationCheck
Confirmación del pedido	OrderOKView

Nota: Muchas de las vistas utilizadas en FashionFlow se crearon específicamente para FashionFlow. Estas vistas se listan en el archivo `command.xml` en el archivador de tienda de FashionFlow. Para obtener más información, consulte “Cómo registrar mandatos, vistas y URL en WebSphere Commerce” en la página 151.

La tabla anterior sólo muestra el conjunto básico de páginas que tiene que crear. Para determinar las demás páginas que debe crear, puede observar detenidamente los ejemplos de uso u otros métodos utilizados para definir los procesos de negocio.

Páginas de error: Los flujos de excepción de los ejemplos de uso también pueden ayudarle a determinar las páginas de error que tiene que crear para la tienda. El ejemplo de uso del registro para FashionFlow especifica los siguientes flujos de excepción:

- La dirección de correo electrónico ya existe: Si la dirección de correo electrónico ya existe en el sistema, éste muestra un mensaje de error solicitándole al usuario que entre otra dirección de correo electrónico. El ejemplo de uso se reanuda desde el principio.
- Faltan campos obligatorios: Si no se ha cumplimentado alguno de los campos siguientes (Dirección de correo electrónico, Contraseña, Verificar contraseña, Nombre, Apellido), el sistema emite un mensaje de error. El ejemplo de uso se reanuda desde el principio.
- Contraseña no válida: Si la contraseña no coincide con la contraseña de verificación, el sistema emite un aviso.

Como consecuencia, deberá crear una página de error o un mensaje de error para cada flujo de excepción.

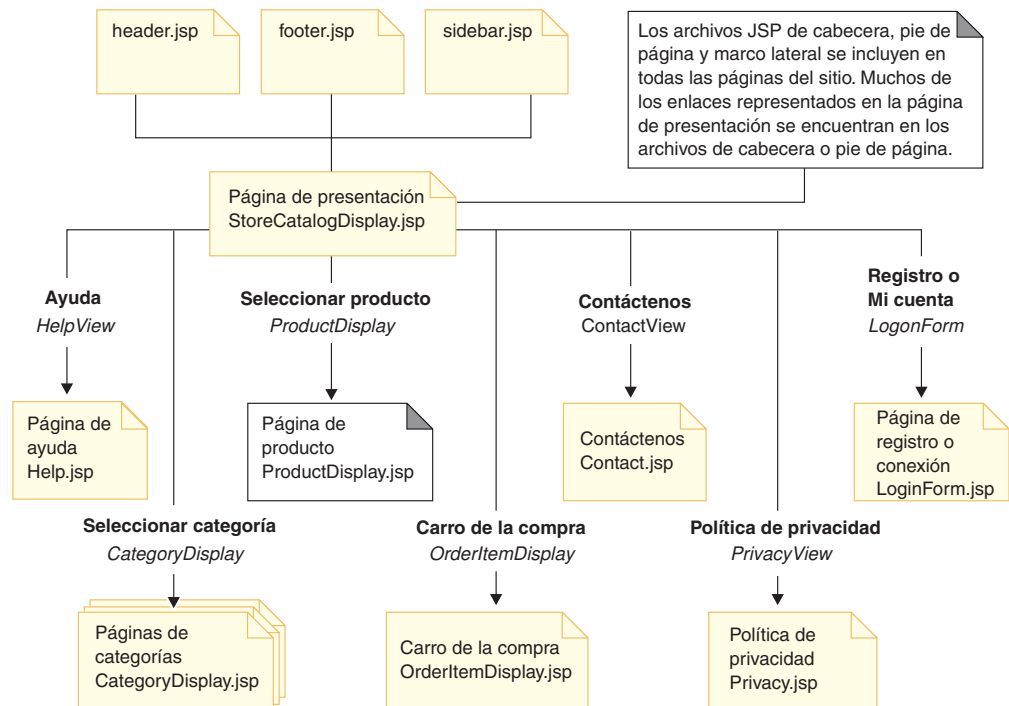
Desarrollar una lista de los URL de mandatos y vistas

Como se ha visto en el diagrama de flujo de compra de FashionFlow, los procesos de negocio como el registro o pasar por caja, pueden requerir varias páginas. Para combinar estas páginas en un flujo o proceso de negocio operativo, en lugar de que sean solamente un conjunto de páginas, debe incluir mandatos y vistas en sus páginas.

Desarrollar una lista de los URL necesarios

Así como ha desarrollado una lista de las páginas que se necesitan para crear la tienda, también debe desarrollar una lista de los URL de mandato y de vista necesarios para implementar los procesos de negocio para su tienda. Utilizando el diagrama de flujo de compra para la tienda y la lista de mandatos y vistas por omisión, identifique los URL necesarios para completar cada acción.

Conocer los URL de mandatos y de vistas que se utilizan en las tiendas de ejemplo puede ayudarle a determinar los URL que necesita en su tienda. La ilustración siguiente identifica los URL para algunas de las acciones que se muestran en el diagrama de flujo de compra de la tienda FashionFlow. Para obtener más detalles, consulte la información sobre tiendas de ejemplo en la publicación *WebSphere Commerce, Guía de tiendas de ejemplo*.



Leyenda

	Estas páginas pueden accederse desde cualquier página del sitio.
Negrita	Acción del cliente
<i>Cursiva</i>	URL

Asociar nombres de archivo JSP a las vistas

WebSphere Commerce Server utiliza mandatos de vista para componer una vista como respuesta a una petición de cliente. WebSphere Commerce Server proporciona los siguientes mandatos de vista:

- `HttpForwardViewCommandImpl`: Este mandato de vista reenvía la petición de vista a un archivo JSP.
- `HttpRedirectViewCommandImpl`: Este mandato de vista reenvía la petición de vista a otro URL.
- `HttpDirectViewCommandImpl`: Este tipo de mandato de vista envía la vista de respuesta directamente al cliente. No llama a un archivo JSP. Las vistas directas permiten que los mandatos de controlador generen respuesta de salida (en lugar del mandato de vista).

Utilice el mandato de vista `HttpForwardViewCommandImpl` para presentar directamente los archivos JSP. Por ejemplo, en el diagrama que ilustra los URL que se utilizan en la tienda FashionFlow, para visualizar la página de ayuda (`Help.jsp`), `HelpView` se registra en el registro de vistas y se asocia con `Help.jsp` y con el mandato `HttpForwardViewCommandImpl`. Esto se muestra en el ejemplo siguiente:

```
<viewreg
viewname="HelpView"
devicefmt_id="-1"
  storeent_id="@storeent_id_1"
interfacename="com.ibm.commerce.command.ForwardViewCommand"
classname="com.ibm.commerce.command.HttpForwardViewCommandImpl"
properties="docname=Help.jsp"
internal="0"
https="0"
/>
```

Tenga en cuenta que se utiliza el nombre de clase (`classname`) totalmente calificado para la interfaz y la clase de implementación.

Utilice el mandato de vista `HttpForwardViewCommandImpl` para presentar las vistas que devuelve un mandato de visualización. Un mandato de visualización lee los datos de la base de datos pero no los modifica. Por ejemplo, en el diagrama que ilustra los URL utilizados en FashionFlow, el mandato `OrderItemDisplay` devuelve la vista `OrderItemDisplayViewShiptoAssoc`. Durante el registro de esta vista en el registro de vistas, se asocian a la misma `OrderItemDisplay.jsp` y `HttpForwardViewCommandImpl`. Esto se muestra en el ejemplo siguiente:

```
<viewreg
viewname="OrderItemDisplayViewShiptoAssoc"
devicefmt_id="-1"
  storeent_id="@storeent_id_1"
interfacename="com.ibm.commerce.command.ForwardViewCommand"
classname="com.ibm.commerce.command.HttpForwardViewCommandImpl"
properties="docname=OrderItemDisplay.jsp"
internal="0"
https="0"
/>
```

Debe asociar un nombre de archivo JSP a cada vista asociada a todo mandato de visualización que va a utilizar (por ejemplo, `OrderItemDisplay`). Para obtener más información acerca de cómo asociar nombres de archivo JSP con vistas, consulte el apartado "Cómo registrar mandatos, vistas y URL en WebSphere Commerce" en la página 151.

Nota: Los mandatos para mostrar productos y categorías devuelven vistas, así como nombres de archivos JSP. Estos nombres de archivos JSP que muestran productos y categorías están almacenados en los datos de catálogo. Para obtener más información, consulte “Visualización de los elementos de catálogo de la tienda” en la página 177. Opcionalmente, puede asignar distintos nombres de archivo JSP para que muestren productos y categorías para cada grupo de miembros o idioma soportado por la tienda.

El mandato de vista `HttpRedirectViewCommandImpl` se utiliza para presentar la salida de un mandato no de visualización (un mandato que modifica la base de datos). Un mandato no de visualización debe estar asociado a un mandato de visualización para evitar que el mandato se vuelva a ejecutar accidentalmente si el cliente vuelve a cargar la página o pulsa el botón Atrás.

Para redirigir a un mandato de visualización, especifique el mandato de visualización utilizando el parámetro `&URL=` en el URL del mandato no de visualización. Por ejemplo, cuando añada información de dirección en el formulario de dirección de la tienda de ejemplo `FashionFlow` y pulsa **Someter**, se llama al mandato `AddressAdd`. El URL que se utiliza para invocar el mandato `AddressAdd` especifica `AddressBookForm` como el parámetro `&URL=`. El resultado es que se redirige al mandato de visualización `AddressBookForm`, que devuelve la vista `AddressBookForm`. Cuando la vista `AddressBookForm` se registró en el registro de vistas, se asoció a `AddressBookForm.jsp` y a `HttpForwardViewCommandImpl`.

Debe utilizar la técnica `URL=parámetro` para todos los mandatos no de visualización. Los mandatos que no son de visualización son mandatos que generan cambios en los datos de la base de datos.

Capítulo 9. Almacenamiento en antememoria de las páginas de tienda

Mientras desarrolla el escaparate también necesita determinar cómo almacenar en la antememoria las páginas de tienda. En este capítulo se describe cómo crear e implementar una estrategia de almacenamiento en antememoria para la tienda.

Planificación de la estrategia de almacenamiento en antememoria

Cuando determine una estrategia de almacenamiento en antememoria, en primer lugar, debe tener en cuenta los temas siguientes:

- Las páginas que deben almacenarse en la antememoria
- Si las páginas deben almacenarse en la antememoria como páginas completas o como fragmentos de páginas.

Páginas que deben almacenarse en la antememoria

Cuando cree su estrategia de almacenamiento en antememoria de alto nivel, en primer lugar debe determinar las páginas de la tienda que deben almacenarse en la antememoria. Las páginas que son buenas candidatas para el almacenamiento en antememoria son las páginas a las que se accede con frecuencia y que a su vez son estables durante un período de tiempo y su contenido pueden volverlo a utilizar diferentes usuarios. Por ejemplo, las páginas de visualización de catálogo son buenas candidatas para habilitar el almacenamiento en antememoria.

Las páginas se han de almacenar en la antememoria como páginas completas o como fragmentos de páginas

En la Versión 5.5, WebSphere Commerce utiliza el servicio de almacenamiento en antememoria dinámico de WebSphere Application Server que permite que WebSphere Commerce dé soporte al almacenamiento en antememoria de páginas Web completas o de fragmentos de páginas. Almacenar en la antememoria una página Web completa simplemente almacena en la antememoria la página entera como una entidad incluso si está compuesta de varios fragmentos más pequeños. Los fragmentos de página pueden incluir una cabecera separador, una barra lateral o un pie de página. Incluso el texto principal de la página puede estar dividido en varios fragmentos. Por ejemplo, un fragmento de la página de texto principal puede mostrar un producto mientras que un segundo fragmento puede mostrar el precio. Fragmentar las páginas le permite mostrar el contenido personalizado a los usuarios individuales. Las páginas de tienda de ejemplo que se proporcionan con WebSphere Commerce constan de varios fragmentos (cabecera, barra lateral, pie de página, contenido principal).

Si las páginas de tienda constan de fragmentos es posible que tenga la oportunidad de almacenarlas en la antememoria como fragmentos. Almacenar en la antememoria los fragmentos individuales le permite almacenar en la antememoria partes de la página que puede volver a utilizar un número mayor de audiencia. Cuando una página contiene información personalizada solamente para una pequeña porción de la audiencia, si se almacena esta página en la antememoria como una página completa no se podrá volver a utilizar con mucha frecuencia porque solamente dicha porción de la audiencia podría volver a utilizar algún día la página almacenada en la antememoria. Por ejemplo, si una página muestra un mensaje de bienvenida para cada cliente de la cabecera y se almacena en la

antememoria basándose en el ID de usuario, entonces, solamente dicho usuario en concreto puede volver a utilizar esta página almacenada en la antememoria. No obstante, si divide la página en fragmentos, puede almacenar en la antememoria los fragmentos que pueden volver a utilizar para la mayor parte del público. Por ejemplo, el pie de página, la barra lateral y los fragmentos de visualización de productos se pueden aplicar a todos los usuarios, mientras que los fragmentos de precio y cabecera se pueden personalizar.

Cuando se solicita la página, se vuelven a ensamblar los fragmentos individuales para generar la página.

Las páginas de tienda se pueden almacenar en la antememoria utilizando el almacenamiento en antememoria de toda la página o de fragmentos o se puede utilizar una combinación de ambos métodos.

Desarrollo de una estrategia de almacenamiento en antememoria más detallada

Después de haber determinado qué páginas y fragmentos de página se deben almacenar en la antememoria, es necesario que determine una estrategia de almacenamiento en antememoria más detallada. Para cada página o fragmento que piense almacenar en antememoria, es necesario determinar lo siguiente:

- Cómo se solicita la página o fragmento
- Si la página o fragmento se basa en una relación de tienda
- Cómo se invalidarán los datos almacenados en la antememoria

Cómo se solicita la página o fragmento

El modo en que se solicita el archivo JSP (tanto si es una sola página o un fragmento de página) determina el modo en el que WebSphere Application Server lo almacenará en la antememoria. Por ejemplo, WebSphere Application Server necesita saber si el archivo JSP se visualizará como respuesta a un servlet, objeto, EJB o un mandato. Como resultado, tiene que compilar una lista de cómo se solicitará cada página o que piensa almacenar en la antememoria.

La página o fragmento se basa en una relación de tienda

Como se describe en el Capítulo 14, “Relaciones entre tiendas”, en la página 143 y en el Capítulo 7, “Arquitectura de la tienda”, en la página 67, las tiendas pueden tener relaciones con otras tiendas que les permite utilizar los datos de otra tienda. Por ejemplo, la tienda A puede utilizar los datos del catálogo definidos en la tienda B. Las tiendas también pueden tener relaciones con varias tiendas, lo que les permite utilizar los datos de diferentes fuentes. Como parte del plan detallado de almacenamiento en antememoria, debe determinar si los datos visualizados en cada página o fragmento se basan en una relación con otra tienda. Si una página no muestra información de otra tienda, cada vez que los datos de otra tienda se actualizan, es necesario actualizar también las páginas almacenadas en la antememoria. Para obtener información sobre cómo almacenar en antememoria las relaciones entre tiendas, consulte el apartado “Implementación del almacenamiento en antememoria de las páginas de tienda que utilicen relaciones entre tiendas” en la página 97.

Cómo se invalidarán los datos almacenados en la antememoria

Para cada página o fragmento de página que piensa almacenar en antememoria, determine cuando la página o fragmento almacenado en antememoria ya no es válido y suprima en la antememoria las entradas de antememoria correspondientes. Este proceso se conoce como invalidación. Para poder determinar cuándo se ha modificado una página almacenada en antememoria y, por lo tanto, ha dejado de ser válida, debe determinar qué hace que caduque la página almacenada en antememoria. Por ejemplo, una página de carro de la compra almacenada en la antememoria no es válida cuando un cliente añade un elemento nuevo al carro. Las páginas almacenadas en la antememoria también se pueden invalidar cuando un administrador actualiza la tienda con WebSphere Commerce Accelerator o cuando se añade un catálogo nuevo con Loader package o con las herramientas de WebSphere Commerce Accelerator.

Después de haber compilado una lista de todos los modos posibles en que se puede invalidar el fragmento de página almacenado en antememoria, debe determinar los sucesos que se utilizan para provocar la invalidación. Los sucesos que provocan la invalidación pueden ser una petición de servlet, un mandato de controlador o un mandato de tarea, etc. Por ejemplo, si actualiza una descripción de producto utilizando las herramientas de gestión de productos en WebSphere Commerce Accelerator, WebSphere Commerce invoca internamente los mandatos, `AddCatalogEntryDescCmd` o `UpdateCatalogEntryDescCmd` para realizar los cambios. Si desea invalidar las páginas almacenadas en antememoria que modifican estos mandatos, debe añadir las políticas de invalidación al archivo `cachespec.xml` que interceptará la ejecución de los mandatos y activar la invalidación. Para obtener más información acerca de cómo implementar la invalidación, consulte lo siguiente:

- El capítulo "Almacenamiento en antememoria dinámico" de la publicación *WebSphere Commerce, Guía de administración*, para obtener instrucciones sobre cómo configurar políticas de invalidación nuevas y un ejemplo de la invalidación de la antememoria.
- El apartado "Invalidación de datos almacenados en antememoria en el archivo `cachespec.xml`" en la página 96, para obtener las instrucciones sobre cómo fusionar las políticas de invalidación de ejemplo que proporciona WebSphere Commerce con el archivo `cachespec.xml` de la tienda.

Implementación de la estrategia de almacenamiento

Después de haber recopilado toda la información detallada que necesita para su estrategia de almacenamiento en antememoria, puede implementarla creando un archivo de políticas de antememoria que defina la información que ha recopilado, incluido lo que se ha de almacenar en antememoria y cómo, y cómo se invalidarán las páginas almacenadas en la antememoria. El servicio de antememoria dinámica de WebSphere Application Server utiliza este archivo de políticas de antememoria conocido como `cachespec.xml` para implementar el almacenamiento en antememoria en la tienda.

Cada tienda de ejemplo proporcionada con WebSphere Commerce incluye un archivo `cachespec.xml` que define la estrategia de almacenamiento en antememoria de dicha tienda. Estos archivos se encuentran en el directorio siguiente:

`dir_instalación_WC/samples/dynacache/BusinessModel`

Puede cambiar estos archivos si la tienda está basada en un ejemplo o puede utilizar uno de estos archivos como base para el archivo `cachespec.xml` de la tienda.

Descripción del archivo `cachespec.xml`

Para poder almacenar en la antememoria las páginas de la tienda de WebSphere Commerce, debe definir objetos almacenables en antememoria en el archivo `cachespec.xml`. WebSphere Commerce solamente utiliza un subconjunto de los elementos definidos en el archivo `cachespec.xml`. En esta sección se describe este subconjunto de elementos. Para obtener información más detallada acerca del archivo `cachespec.xml`, consulte en el Centro de información de WebSphere Application Server (<http://www.ibm.com/software/webservers/appserv/infocenter.html>), el "Archivo `cachespec.xml`". Para obtener más información, consulte el capítulo "Almacenamiento en antememoria dinámico" de la publicación *WebSphere Commerce, Guía de administración*.

Descripción de los elementos utilizados por WebSphere Commerce

WebSphere Commerce utiliza los elementos siguientes en el archivo `cachespec.xml`:

- Class
- Name
- Property

En el ejemplo siguiente se describe el uso de estos cuatro elementos:

```
<cache-entry>
  <class>servlet</class>
    <name>/FashionFlow/ShoppingArea/CatalogSection/CategorySubsection
/StoreCatalogDisplay.jsp</name>
  <property name="save-attributes">false</property>
```

Class: El elemento `class` es un elemento necesario. Determina el modo en que WebSphere Application Server interpretará el resto de la definición de políticas de antememoria restante. WebSphere Commerce utiliza los valores `class` siguientes:

- `command`
- `servlet`

El valor `command` hace referencia a las clases que utilizan el modelo de programación de WebSphere Commerce.

El valor `servlet` hace referencia servlets o archivos JSP desplegados en el motor del servlet de WebSphere Application Server.

Nota: En WebSphere Commerce versión 5.5, sólo está soportada la invalidación de mandatos.

Name: `Name` es el nombre de clase totalmente calificado del servlet o mandato. `Name` es un elemento obligatorio.

Los valores de `name` para los mandatos deben incluir el nombre del paquete. Por ejemplo, el nombre de clase `com.ibm.commerce.dynacache.commands.MemberGroupsCacheCmdImpl`

Los valores de nombre para los archivos JSP y el servlet deben incluir el URI completo del archivo JSP o servlet que se ha de almacenar en la antememoria. Por ejemplo, `com.ibm.commerce.server.RequestServlet.class`

```
/ToolTech/ShoppingArea
/CatalogSection/CategorySubsection/StoreCatalogDisplay.jsp.
```

Property: El elemento `property` toma el formato siguiente: `<property name=clave>valor</property>`, donde *clave* es el nombre de la propiedad que se está definiendo y *valor* es el valor correspondiente. Puede establecer las propiedades opcionales en un objeto almacenable en la antememoria. Por ejemplo, `<property name="consume-subfragments">true</property>`

Cuando WebSphere Commerce almacena en antememoria páginas de tienda, se utilizan las propiedades siguientes:

Propiedad	Valor	Clases válidas	Descripción
EdgeCacheable	True o false. El valor por omisión es false.	Servlet	Si el valor es true, entonces el servlet o archivo JSP determinado se solicita externamente desde un Edge Server. Si el archivo JSP o servlet se puede almacenar en antememoria depende del resto de la especificación de antememoria.

consume-subfragments	True o false. El valor por omisión es false.	Servlet	<p>Cuando se almacena en antememoria un servlet solamente se almacena el contenido de dicho servlet. Los contenedores de cualquier otro fragmento para los que se crean inclusiones o envíos. Los CSF (consume-subfragments) indican a la antememoria que continúe guardando el contenido cuando encuentre un servlet hijo mediante include. La entrada padre (la que está marcada con CSF) incluirá todo el contenido de todos los fragmentos en la entrada de antememoria, lo que dará como resultado una entrada de antememoria de gran tamaño sin ninguna inclusión o envío, pero con el contenido de todo el árbol de entradas. Este método puede guardar una importante cantidad de proceso del servidor de aplicaciones, pero generalmente sólo es útil cuando la petición HTTP externa contiene toda la información necesaria para determinar todo el árbol de fragmentos incluidos.</p>
save-attributes	True o false. El valor por omisión es true.	Servlet	<p>Cuando se establece save-attributes en false, los atributos de la petición no se guardan con la entrada de antememoria.</p>

store-cookies	True o false. El valor por omisión es true.	Servlet	Cuando se establece store-cookies en false, los atributos de la petición no se guardan con la entrada de antememoria.
---------------	---	---------	---

Por omisión, DynaCache almacena en antememoria los cookies (cuando el almacenamiento en antememoria se realiza mediante la clase servlet) junto con las entradas de antememoria. No obstante, los cookies de WebSphere Commerce y los atributos de la petición contienen información específica del usuario que no se debe almacenar en la antememoria. Como resultado, son obligatorios los nombres y valores de propiedades siguientes cuando se almacenan en la antememoria páginas completas.

```
<property name="save-attributes">false</property>
<property name="store-cookies">false</property>
```

El nombre y valor de la propiedad siguientes son obligatorios para todas las entradas de antememoria definidas para los archivos JSP:

```
<property name="save-attributes">false</property>
```

Descripción de las normas de cache-ID

Un cache-ID identifica de forma exclusiva una entrada de antememoria. Para que WebSphere Application Server pueda almacenar en antememoria un objeto, debe saber cómo generar un ID exclusivo para las diferentes invocaciones de dicho objeto. Estos ID se crean a partir del código Java personalizado escrito por el usuario o a partir de las normas de una política de antememoria de la entrada de antememoria.

En el archivo cachespec.xml, el elemento cache-id define las normas para generar los ID. Cada entrada de antememoria puede tener varias normas de tipo cache-ID que se ejecutarán en el orden definido hasta que una norma devuelva un ID de antememoria que no esté vacío o no queden más normas que ejecutar. Si ninguna de las normas de generación de cache-ID genera un ID de antememoria válido, entonces el objeto no se almacena en la antememoria.

Estos ID se desarrollan de uno de los modos siguientes:

- Utilizando los elementos de los componentes definidos en la política de antememoria de una entrada de antememoria.
- Escribiendo código Java personalizado para crear el ID a partir de las variables de entrada y del estado del sistema.

Descripción de las normas de dependency-ID

Los elementos de ID de dependencia especifican identificadores de grupos de antememoria adicionales que asocian varias entradas de antememoria con el mismo identificador de grupo. El ID de dependencia se genera concatenando la serie básica de ID de dependencia con los valores que devuelven los elementos de componentes. Si un componente obligatorio devuelve un valor nulo, entonces ni se genera el ID de dependencia ni se utiliza.

Puede validar los ID de dependencia explícitamente mediante la API de almacenamiento en antememoria dinámico de WebSphere o mediante otro elemento de invalidación de cache-entry. Pueden haber varias normas de ID de dependencia por entrada de antememoria. Todas las normas de ID de dependencia

se ejecutan por separado. Para obtener más información sobre cómo se definen las normas de ID de dependencia, consulte el capítulo "Almacenamiento en antememoria dinámica" de la publicación *WebSphere Commerce, Guía de administración*.

Descripción de las normas de invalidación

Las normas de invalidación se pueden definir exactamente del mismo modo que los ID de dependencia. No obstante, los ID que generan las normas de invalidación se utilizan para invalidar las entradas de antememoria que tienen los mismos ID de dependencia. El ID de invalidación se genera concatenando la serie básica de ID de invalidación con los valores que devuelve su elemento de componente. Si un componente obligatorio devuelve un valor nulo, entonces ni se genera el ID de invalidación ni se utiliza. Pueden haber varias normas de invalidación por entrada de antememoria. Todas las normas de invalidación se ejecutan por separado. Para obtener más información sobre cómo se definen las normas de invalidación, consulte la publicación *WebSphere Commerce, Guía de administración*.

Invalidación de datos almacenados en antememoria en el archivo cachespec.xml

Por omisión, los archivos `cachespec.xml` que se envían con los archivadores de tienda de ejemplo no incluyen las políticas de invalidación. Si desea automatizar la invalidación de antememoria automática utilizando DynaCache en una tienda de ejemplo, o una tienda basada en un ejemplo, debe añadir políticas de invalidación en el archivo `cachespec.xml` de la tienda. Las políticas de invalidación de ejemplo se proporcionan en varios archivos `cachespec.xml` del directorio siguiente:

dir_instalación_WC/samples/dynacache/invalidation

Este directorio contiene varios archivos `cachespec.xml` para las áreas de función, incluido el catálogo, el carro de la compra, la tienda, etc. Cada archivo contiene políticas de invalidación para dicha área específica.

Si piensa almacenar en antememoria las páginas de catálogo de la tienda, debe añadir las políticas a la tienda desde los archivos siguientes:

- *dir_instalación_WC/samples/dynacache/invalidation/catalog/cachespec.xml*
- *dir_instalación_WC/samples/dynacache/invalidation/membergroup/cachespec.xml*

Nota: Para estas normas de invalidación de grupos de miembros necesita añadir ID de dependencia adicionales a las entradas de antememoria. Para obtener más información, consulte el contenido de este archivo `cachespec.xml`.

- *dir_instalación_WC/samples/dynacache/invalidation/store/cachespec.xml*

Adición de políticas de invalidación de ejemplo al archivo cachespec.xml de la tienda

Para poder añadir a la tienda las políticas de invalidación que se proporcionan en los archivos de invalidación de ejemplo, haga lo siguiente:

1. Abra el archivo `cachespec.xml` de la tienda.
 - Directorio
dir_instalación_WAS/installedApps/nombre_célula/WC_instanceName/Stores.war/WEB-INF.

Si la tienda no tiene definidas políticas de antememoria y está basada en un ejemplo proporcionado con WebSphere Commerce, puede utilizar un archivo `cachespec.xml` de ejemplo del directorio siguiente:

- *dir_instalación_WC/samples/dynacache/BusinessModel*
2. Abra el archivo `cachespec.xml` de invalidación de ejemplo. Los archivos `cachespec.xml` de invalidación de ejemplo se encuentran en el directorio siguiente:
 - *dir_instalación_WC/samples/dynacache/invalidation*
 3. Copie las políticas de invalidación del archivo de invalidación de ejemplo en el archivo `cachespec.xml` de la tienda. Puede colocar las políticas de invalidación al final del archivo `cachespec.xml` de la tienda después del último elemento.
 4. Asegúrese de que los ID de invalidación coincidan con los ID de dependencia correspondientes de las políticas de almacenamiento en antememoria. Si ningún ID de dependencia coincide, entonces no se ejecutarán las políticas de invalidación y debe cambiar el ID de invalidación de la norma de invalidación o el ID de la norma `dependency-id`, para que coincidan.
- Nota:** Es posible que la tienda tenga requisitos de negocio adicionales o diferentes que requieran que añada políticas de invalidación e ID de dependencia adicionales.
5. Si es necesario, cambie el nombre y el directorio de los archivos JSP en las secciones que se han copiado de los archivos de invalidación de ejemplo para que coincidan con la información del resto del archivo `cachespec.xml` de la tienda.
 6. Guarde el archivo.

Implementación del almacenamiento en antememoria de las páginas de tienda que utilicen relaciones entre tiendas

Si la tienda está utilizando datos definidos en otra tienda mediante una relación entre tiendas, debe utilizar los atributos especificados mediante el filtro de antememoria para definir las relaciones. El filtro de antememoria es un filtro de servlet que define los atributos de petición de la sesión y la información de relaciones entre tiendas que puede utilizar DynaCache de WebSphere Application Server. A continuación, DynaCache utiliza esta información para crear los ID de antememoria y los ID de dependencia que se van a utilizar para la invalidación de antememoria. Para obtener una lista de la configuración de atributos de petición para la información de sesión, consulte el capítulo "Almacenamiento en antememoria dinámico" de la publicación *WebSphere Commerce, Guía de administración*.

El filtro de antememoria crea la información sobre relaciones de tienda llamando a los métodos `getStorePath()` y `getStoresForRelatedStore()` desde `StoreAccessBean`. La información correspondiente se lista en la tabla siguiente:

Tabla 3.

Tipo de relación entre tiendas	Identificador de relación entre tiendas	Nombre de los atributos de petición para <code>getStorePath()</code>	Nombre de los atributos de petición para <code>getStoresForRelatedStore()</code>
Política de negocio de IBM Commerce	-1	DC_busN	DC_bus_RS_N
Campañas de negocio de IBM Commerce	-3	DC_campN	DC_camp_RS_N
Catálogo de negocio de IBM Commerce	-4	DC_catN	DC_cat_RS_N

Tabla 3. (continuación)

Mandato de negocio de IBM Commerce	-5	DC_cmdN	DC_cmd_RS_N
Tienda alojada de IBM Commerce	-6	DC_hostN	DC_host_RS_N
Precio de IBM Commerce	-7	DC_prcN	DC_prc_RS_N
Consulta de IBM Commerce	-8	DC_refN	DC_ref_RS_N
Segmentación de IBM Commerce	-9	DC_segN	DC_seg_RS_N
URL de IBM Commerce	-10	DC_urlN	DC_url_RS_N
Vista de IBM Commerce	-11	DC_viewN	DC_view_RS_N
Inventario de IBM Commerce	-13	DC_invN	DC_inv_RS_N
Artículo básico de IBM Commerce	-14	DC_baseItemN	DC_baseItem_RS_N
Tienda de canal de IBM Commerce	-15	DC_chsN	DC_chs_RS_N
Conversión de moneda de IBM Commerce	-17	DC_currConvN	DC_currConv_RS_N
Formato de moneda de IBM Commerce	-18	DC_currFmtN	DC_currFmt_RS_N
Moneda soportada de IBM Commerce	-19	DC_supCurrN	DC_supCurr_RS_N
Moneda de contravalor de IBM Commerce	-20	DC_cterCurrN	DC_cterCurr_RS_N
Formato de medición de IBM Commerce	-21	DC_meaFmtN	DC_meaFmt_RS_X

Nota: El filtro de antememoria establece varios atributos de petición cuando varios ID de tienda devueltos como DynaCache no soportan una matriz de atributos de petición. Por ejemplo, si `getStorePath()` devuelve una matriz [10051, 10002] para el recurso id -4 (catálogo de negocio de IBM commerce), entonces los atributos de petición se configurarán como

- DC_cat0 is 10051
- DC_cat1 is 10002

Ejemplo de almacenamiento en antememoria de relaciones entre tiendas

Para comprender cómo funcionan las páginas de antememoria que utilizan relaciones entre tiendas, tenga en cuenta el ejemplo siguiente.

Cuando se publica el archivador de tienda compuesto de ejemplo DemandChain.sar y luego se crea una tienda alojada (por ejemplo, ResellerOne) en el sitio, se crearán las tiendas siguientes:

Tabla 4.

ID de tienda	Directorio	Tipo de tienda
10001	CommercePlaza	Centro de canal
10002	CommercePlazaCatalog	tienda con elementos de catálogo
10003	CommercePlaza	proxy de distribuidor
10004	ConsumerDirectResellerProfile	tienda con elementos de escaparate alojada
10051	ResellerOne	tienda alojada de revendedor

ResellerOne (10051), la tienda alojada de revendedor, utiliza los elementos definidos en la tienda con elementos de escaparate (10004) y la tienda con elementos de catálogo (1002).

Para poder configurar la relación de almacenamiento en antememoria, el filtro de antememoria obtiene la información siguiente:

Tabla 5.

ID de tienda	Tipo de relación	getStorePath()	getStoresFor RelatedStore()
10001	-1 (política de negocio) -4 (catálogo) -7 (precio) -17 (formato de moneda) -19 (moneda soportada)	10002	No es aplicable
10001	-6 (tienda alojada)	10051	No es aplicable
10051	-1 (política de negocio) -14 (artículo base)	10051, 10002, 10004	10051
10051	-3 (campañas) -5 (mandato) -10 (URL) -11 (vista)	10051, 10004	10051
10051	-4 (catálogo) -7 (precio) -17 (conversión de moneda) -18 (formato de moneda) -19 (moneda soportada) -20 (moneda de contravalor) -21 (formato de medida)	10051, 10002	10051

A continuación, el filtro de antememoria establece los atributos de petición siguientes:

Tabla 6.

Relación entre tiendas	ID de tienda 10051	ID de tienda 10051	ID de tienda 10001
-1 (política de negocio)	DC_bus0=10051 DC_bus1=10002 DC_bus2=10004	DC_bus_RS_0=10051	DC_bus0=10002
-2 (impuesto)	DC_tax0=10051 DC_tax1=10004	DC_tax_RS_0=10051	
-4 (catálogo)	DC_cat0=10051 DC_cat1=10002	DC_cat_RS_0=10051	DC_cat0=10002
-6 (tienda alojada)	DC_host0=10051	DC_host_RS_0=10001	DC_host0=10051

Cuando el catálogo de la tienda con elementos de catálogo (10002) se modifica, las páginas de catálogo de la tienda ResellerOne (10051) también deben invalidarse antes de poder utilizar la información de la tienda con elementos de catálogo (10002). Para poder invalidar las páginas en 10051, deben establecerse varios ID de dependencia para esta relación entre tiendas. En el ejemplo siguiente, se muestra la configuración de ID de dependencias de StoreCatalogDisplay:

```
<!-- Inicio de los ID de dependencia de relación entre tiendas -->
<!-- DC_cat1 es el ID de tienda con perfil de catálogo -->
<dependency-id>storeId
<component id="DC_cat1" type="attribute">
<required>true</required>
</component>
</dependency-id>

<dependency-id>storeId:catalogId
<component id="DC_cat1" type="attribute">
<required>true</required>
</component>
<component id="catalogId" type="attribute">
<required>true</required>
</component>
</dependency-id>

<dependency-id>StoreCatalogDisplay:storeId
<component id="DC_cat1" type="attribute">
<required>true</required>
</component>
</dependency-id>
<!-- Fin de los ID de dependencia de relaciones entre tiendas-->
```

Los ID de dependencia adicionales se crean del modo siguiente:

- storeId:10002
- storeId:catalogId:10002:10051
- StoreCatalogDisplay:storeId:10002

Una vez definidos los ID de dependencia adicionales, siempre que se produzcan cambios en la tienda con elementos de catálogo 10002 que hagan que se invaliden las páginas de tienda con elemento de catálogo, se invalidarán también las páginas de la tienda alojada (10051).

Sustitución de las funciones de CacheCommand por el almacenamiento en antememoria dinámico

En las versiones anteriores de WebSphere Commerce se utilizaba CacheCommand (com.ibm.commerce.cache.commands.CacheCommandImpl) para implementar configuraciones de antememoria avanzada, por ejemplo, el almacenamiento en antememoria de páginas según el estado y tipo del usuario que determina un perfil de cliente.

En la Versión 5.5, utilizando el almacenamiento en antememoria dinámico se puede almacenar en la antememoria el resultado del servlet o archivo JSP como lo haría con el mandato CacheCommand, añadiendo la lógica del mandato a un archivo JSP.

Observe el ejemplo siguiente:

El mandato StoreCatalogDisplay puede mostrar varias cabeceras diferentes basándose en los atributos de estado y tipo del usuario. Para almacenar en antememoria el archivo JSP de cabecera, cree un archivo JSP nuevo, CacheParametersSetup.jsp que incluya los atributos de estado y tipo del usuario. Por ejemplo:

```
<%@ page import="com.ibm.commerce.command.CommandContext" %>
<%
    String userState = null;
    String userType = null;

    CommandContext cmdcontext = (CommandContext) request.getAttribute
(ECCConstants.EC_COMMANDCONTEXT);
    if (cmdContext != null) {
        userState = cmdcontext.getUser().getState();
        userType = cmdcontext.getUser().getRegisterType();
    }
%>
```

A continuación, StoreCatalogDisplay.jsp incluye de forma estática CacheParametersSetup.jsp y de forma dinámica incluye CachedHeaderDisplay.jsp utilizando userState y userType como parámetros de entrada:

```
<%@ include file="CacheParametersSetup.jsp"%>
<jsp:include page="CachedHeaderDisplay.jsp" flush="true">
  <jsp:param name="storeId" value="<%= storeId %>" />
  <jsp:param name="catalogId" value="<%= catalogId %>" />
  <jsp:param name="langId" value="<%= languageId %>" />
  <jsp:param name="userState" value="<%= userState %>" />
  <jsp:param name="userType" value="<%= userType %>" />
</jsp:include>
```

El archivo CachedHeaderDisplay.jsp contiene la lógica para mostrar información diferente en función de los parámetros de entrada.

```
<%
    if (userType.equals("G")) {
%>
        <table cellpadding="0" cellspacing="0" border="0" width="100%" height="28">
            .
            .
            .
        </table>
<%
    }
    else {
%>
        <table cellpadding="0" cellspacing="0" border="0" width="100%" height="28">
            .
            .
            .
```

```

        </table>
<%
    }
%>

```

Para poder completar el almacenamiento en antememoria, deben identificarse los parámetros de entrada mediante una norma de ID de antememoria.

```

<cache-entry>
  <class>servlet</class>
  <name>../CachedHeaderDisplay.jsp</name>
  <property name="save-attributes">>false</property>

  <cache-id>
    <component      id="storeId" type="parameter">
      <required>>true</required>
    </component>
    <component      id="catalogId" type="parameter">
      <required>>true</required>
    </component>
    <component      id="userState" type="parameter">
      <required>>true</required>
    </component>
    <component      id="userType" type="parameter">
      <required>>true</required>
    </component>
  < /cache-id>
  . . .
</cache-entry>

```

Parte 5. Visión general de los datos de la tienda

Capítulo 10. Datos de la tienda

Este capítulo proporciona una visión general de la arquitectura de datos de la tienda de WebSphere Commerce Server y de los elementos de datos que crean una tienda. El modelo de información de WebSphere Commerce Server también se trata en este capítulo.

¿Qué son los datos de la tienda?

Los datos de la tienda es la información que se carga en la base de datos de WebSphere Commerce Server y que permite el funcionamiento de la tienda. Para que una tienda funcione correctamente, debe tener disponibles los datos para dar soporte a todas las actividades de los clientes. Por ejemplo, para que un cliente efectúe una compra, la tienda debe contener un catálogo de los productos en venta (datos de catálogo), los datos asociados al proceso de pedidos (datos sobre impuestos y envío) y el inventario para realizar la petición (datos de inventario y de despacho de pedidos).

Modelo de información de los datos de la tienda

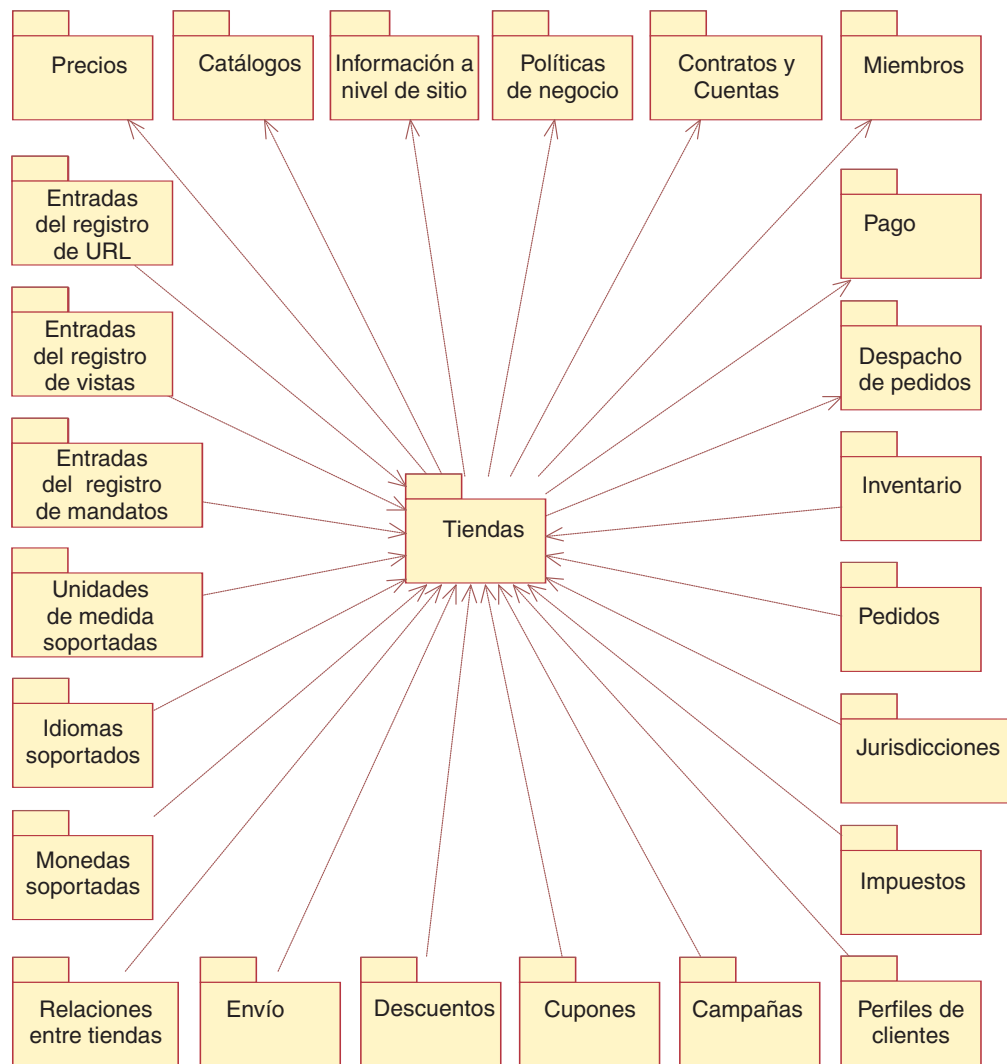
Esta guía utiliza un modelo de información para ilustrar cómo se estructuran los datos de la tienda en WebSphere Commerce Server. El modelo de información de WebSphere Commerce Server es un resumen de alto nivel de la información que contienen los modelos de datos de WebSphere Commerce Server. El modelo de información destaca las características más importantes de los modelos de datos, pero no incluye los detalles específicos de las implementaciones del objeto y del esquema.

Por ejemplo, ciertas tablas y objetos en los modelos de datos que contienen datos de relaciones entre entidades (como, por ejemplo, las parejas de claves externas) no se muestran como entidades en el modelo de información. En su lugar, estas relaciones de entidades están implementadas por las líneas de relaciones entre entidades en los modelos de información. El modelo de información también difiere del modelo de datos en que en este último cada entidad representa a una tabla mientras que en el modelo de información cualquiera de los objetos representados puede correlacionarse con la misma tabla de base de datos o un objeto individual puede correlacionarse con varias tablas de base de datos. El modelo de información tampoco ilustra las *extensiones detalladas* (que son los atributos de datos adicionales de una entidad que se almacenan en una tabla diferente como consecuencia de los diferentes aspectos de su implementación; por ejemplo, la descripción del producto es una extensión de la entidad de producto que se almacena en un lugar diferente). Por último, a diferencia del modelo de datos, el modelo de información puede ilustrar también conceptos de herencia. Para obtener más información sobre los datos de relaciones entre entidades y sobre las extensiones detalladas, consulte el modelo de datos en la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.



Para obtener más información sobre los modelos de objetos y de datos de WebSphere Commerce, consulte la ayuda en línea de WebSphere Commerce.

El diagrama siguiente ilustra los elementos de datos de una tienda de WebSphere Commerce.



Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce. Para obtener más información sobre el modelo de información, consulte "Modelo de información de los datos de la tienda" en la página 105. Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, "Descripción de UML", en la página 463.

Cada uno de los elementos de datos ilustrados en los diagramas anteriores se trata en profundidad en los capítulos de la Parte 6, "Desarrollo de los datos de la tienda", en la página 117.

Nota:

En la anotación UML, una línea discontinua con una flecha que apunta de un objeto a otro objeto indica que el primer objeto tiene una dependencia sobre el segundo objeto. En este diagrama, se hace referencia a los objetos mostrados como paquetes. Observe que los datos de algunos paquetes

como, por ejemplo, las listas de monedas soportadas, son específicas de una tienda determinada y, de este modo, dicho paquete aparece como dependiente del paquete de la tienda. Otros paquetes como, por ejemplo, los catálogos, no utilizan ninguna tienda determinada sino que cada tienda puede utilizar los catálogos y, de este modo, el objeto Tienda aparece como dependiente en el paquete Catálogos. Como resultado, las listas de Monedas soportadas forman *parte de* una tienda, mientras que una Tienda *utiliza* Catálogos.

Una *parte de* una Tienda que tiene un interés determinado son sus Relaciones entre tiendas con otras Tiendas. Cada relación de tienda indica que una Tienda depende de otra Tienda para proporcionar algún servicio o información. Se pueden definir relaciones para permitir que los datos de una Tienda, por ejemplo, la lista de Monedas soportadas, puedan ser utilizados por otra Tienda. En este escenario, la primera Tienda actúa como proveedor, o contenedor, de los datos que utiliza la segunda Tienda, el cliente. A medida que se crean más Tiendas de cliente, también pueden definir relaciones que indican que se obtienen determinados datos de determinadas Tiendas. De este modo, las relaciones entre Tiendas permiten compartir datos, ya que los datos los crea y mantiene una vez la Tienda proveedora y los utilizan varias Tiendas de clientes. Para obtener más información sobre las relaciones entre tiendas, consulte el Capítulo 14, “Relaciones entre tiendas”, en la página 143.

Los datos del modelo de información se pueden clasificar de los modos siguientes:

- por subsistema
- por tipo de datos

Modelo de información de datos de la tienda visto por subsistema

Cada uno de los elementos de datos del modelo de información de datos de la tienda se puede agrupar en las áreas de funciones siguientes:

Tabla 7.

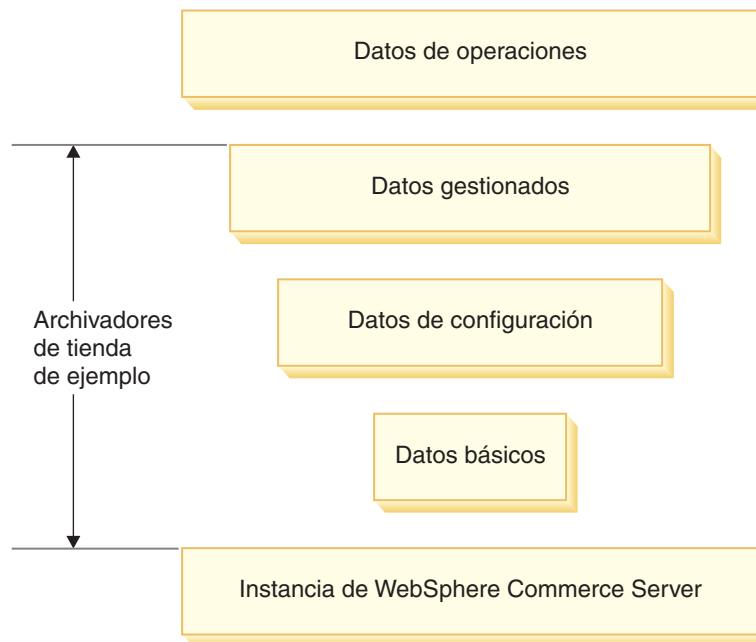
Comercia- lización	Marketing	Intercam- bio	Gestión de pedidos	Catálogo	Miembro	Tiempo de ejecución
Descuentos	Campañas	Contratos	Envío	Catálogos	Organiza- ciones	Organiza- ciones
Proveedores	Perfiles de clientes	Cuentas	Impuestos	Precios	Grupos	URL, mandatos y registro de vistas
Subastas	Actividad de correo electrónico	RFQ	Jurisdic- ciones		Usuarios	Idiomas soportados
	Cupones		Pedidos			Unidades de medida soportadas
			Inventario			Monedas soportadas
			Despacho de pedidos			Sitio
			Pago			Tienda

Tabla 7. (continuación)

						Relaciones entre tiendas
						Políticas de negocio

Modelo de información de datos de la tienda visto por tipo de datos

Los datos de las tiendas de WebSphere Commerce se ajustan a los tipos descritos en el siguiente diagrama. Cada uno de los elementos de datos de la tienda ilustrados en el diagrama del apartado “Modelo de información de datos de la tienda visto por subsistema” en la página 107, puede clasificarse como perteneciente a uno o más de los tipos de datos de tienda ilustrados a continuación.



Instancia de WebSphere Commerce Server

El nivel más básico de datos se encuentra en la instancia de WebSphere Commerce Server. Cuando se crea una instancia, los archivos de rutina de carga que tienen formato XML, llenan la base de datos con información. Los archivos de rutina de carga crean los siguientes tipos de datos:

- los tipos de usos de cálculo, tipos de dispositivo, (navegadores, correo electrónico, I-Mode, etc.), tipos de mensajes, roles y direcciones
- el ID de administración por omisión, WCSADMIN
- los mandatos, URL y vistas por omisión
- las políticas de negocio por omisión
- los grupos de acceso y las políticas de control de acceso por omisión
- los idiomas y las monedas soportadas por la instancia
- las unidades de cantidad y las conversiones de la unidad de cantidad por omisión

- los trabajos planificados y los códigos de estado por omisión
- los términos y condiciones por omisión
- la organización por omisión, que se puede utilizar como el propietario de la tienda
- la organización por omisión del sitio
- el grupo de tiendas por omisión
- la información por omisión para la transición

Esta información está disponible para todas las tiendas que hay en esa instancia, y se identifica como Información a nivel de sitio en el diagrama que se muestra en “Modelo de información de datos de la tienda visto por subsistema” en la página 107.

Para obtener información más detallada sobre los archivos de rutina de carga y las tablas de base de datos que rellenan, consulte la ayuda en línea de WebSphere Commerce.

Datos básicos

El siguiente nivel de datos de tienda son los datos básicos. Los datos básicos se dividen en dos niveles:

- Organización
- Tienda

Los datos básicos de la organización crean los datos mínimos para un entorno específico de modelo de negocio, incluido:

- La estructura organizativa.
- Roles de usuario predefinidos.
- Políticas de control de acceso necesarias.

Tanto los archivadores de tienda compuestos de ejemplo como los archivadores de tienda de los componentes de estructura de organización de ejemplo disponen de datos básicos de organización.

Los datos básicos crean los datos imprescindibles para una tienda dentro de dicho entorno, incluidos:

- El identificador de tienda en la tabla STOREENT. Esto crea una tienda en la base de datos.
- El contrato por omisión.
- El identificador de tienda en las tablas de base de datos de contratos.
- El identificador de miembro de la organización propietaria de la tienda según las tablas de base de datos de contrato
- El directorio de tienda en la tabla STORE. El directorio de la tienda es el directorio en el que se encuentran los elementos web de la tienda.
- El apodo o identificador de la dirección de la tienda en la tabla STADDRESS. El apodo es exclusivo para cada tienda.

Tanto los archivadores de tienda compuestos de ejemplo como los archivadores de tienda de los componentes de ejemplo disponen de datos básicos de tienda.

Esta información se habrá creado automáticamente, si ha publicado cualquiera de los archivadores de tienda de ejemplo indicados anteriormente utilizando el programa de utilidad de publicación en la Consola de administración. El programa de utilidad de publicación permite seleccionar la organización por omisión que

puede actuar como propietario de la tienda, o puede crear otra organización que actúe como propietario utilizando la Consola de administración de organizaciones. Si no ha publicado un archivador de tienda compuesto de ejemplo y lo utiliza como base para la tienda, tendrá que cargar esta información en la base de datos utilizando Loader package o editar directamente la base de datos. Para obtener más información sobre cómo utilizar Loader package, consulte el Capítulo 37, “Visión general de la carga de datos de tienda”, en la página 361.

Los datos de la tienda del diagrama que se muestra en “Modelo de información de datos de la tienda visto por subsistema” en la página 107 son datos básicos.

Datos de configuración

Los datos de configuración controlan la ejecución del servidor de comercio. La ejecución de servidor de comercio proporciona un entorno en el que se despliegan y se ejecutan las aplicaciones de comercio. La infraestructura está formada por la ejecución de mandatos, el manejo de excepciones, el control de datos, el acceso a datos y la persistencia. La ejecución del servidor de comercio se beneficia de los servicios de tiempo de ejecución que proporciona WebSphere Application Server para dar soporte a las aplicaciones de WebSphere Commerce Server. Los datos de configuración determinan los mandatos, vistas y archivos JSP que utilizará su tienda para mostrar las páginas de la tienda

Los siguientes elementos de datos identificados en el diagrama que se muestra en “Modelo de información de datos de la tienda visto por subsistema” en la página 107 están clasificados como Datos de configuración:

- Entradas del registro de mandatos
- Entradas del registro de vistas
- Entradas del registro de URL

Datos gestionados

Los datos gestionados son los datos que crea el vendedor y que los clientes sólo pueden leer en el sitio del vendedor. Puesto que el vendedor tiene el control absoluto sobre el estado de estos datos, los datos gestionados pueden administrarse mediante un sistema de gestión de contenido.

Los siguientes elementos de datos identificados en el diagrama que se muestra en “Modelo de información de datos de la tienda visto por subsistema” en la página 107 están clasificados como Datos gestionados:

- Políticas de negocio
- Campañas
- Catálogos
- Contratos
- Cupones
- Monedas
- Perfiles de clientes
- Descuentos
- Actividad de correo electrónico
- Centros de despacho de pedidos
- Inventario (información de configuración para elementos de catálogo)
- Jurisdicciones
- Idiomas
- Miembros

- Pago
- Precios
- Vendedores
- Envío
- Impuestos
- Unidades de medida
- Proveedores

Datos de operaciones

Los datos de operaciones son los datos que los clientes del sitio crean o modifican (directa o indirectamente) como resultado de sus interacciones con el sitio. Por ejemplo, los pedidos de los clientes se consideran datos de operaciones, al igual que los niveles de inventario, que suben y bajan según las operaciones de la tienda. Los clientes también se consideran datos de operaciones. Asimismo, los datos creados por el vendedor pueden ser datos de operaciones.

Puesto que los cambios de los datos de operaciones no están bajo el control absoluto del vendedor, no tendría sentido administrar estos datos mediante un sistema de gestión de contenido.

Los siguientes elementos de datos identificados en el diagrama que se muestra en “Modelo de información de datos de la tienda visto por subsistema” en la página 107 están clasificados como Datos de operaciones:

- Subastas
- Contratos
- Clientes
- Actividad de correo electrónico
- Despacho de pedidos
- Inventario (recepciones, recepciones esperadas, asignación de inventario)
- Pedidos
- Solicitud de presupuesto (RFQ)

Nota: En algunas instancias, la línea que separa los datos de operaciones de los datos gestionados puede ser difícil de determinar. Por ejemplo, en una tienda, los datos de clientes y de contratos pueden considerarse datos gestionados, mientras que en otra tienda, el mismo tipo de datos puede ser considerado de operaciones. La primera tienda puede gestionar los datos de los clientes y sus contratos relacionados porque tienen un conjunto específico de clientes (es decir, los clientes no se pueden registrar en línea). Sin embargo, la segunda tienda permite a los clientes registrarse en línea y crear información de los contratos en línea.

Un segundo ejemplo trata de los datos de catálogo. En un sitio de un solo vendedor, los datos del catálogo se consideran datos gestionados. En un sitio de cadena de valores, los datos del catálogo se pueden considerar datos de operaciones.

En algunos sitios, ciertos registros del mismo tipo de datos pueden considerarse gestionados, mientras que otros registros se consideran de operaciones. Por ejemplo, el contrato por omisión puede considerarse como datos gestionados, pero los contratos específicos negociados en línea pueden formar parte de los datos de operaciones. Otro ejemplo es la actividad de correo electrónico. La información sobre las actividades de correo electrónico

y las plantillas se consideran datos gestionados, pero las actividades de correo electrónico reales que se generan a partir de las plantillas y se envían a los clientes se consideran datos de operaciones, al igual que cualquiera de los sucesos resultantes de la actividad de correo electrónico, por ejemplo, cuando un cliente abre el correo electrónico o pulsa cualquier elemento del correo electrónico que se pueda seleccionar.

Tipos de datos de la tienda y negocios de ejemplo

Los negocios de ejemplo que se proporcionan con WebSphere Commerce incluyen la mayoría de los tipos de datos de tienda de la arquitectura de datos de tienda. Por ejemplo, para poder crear una tienda utilizando una tienda de ejemplo o para poder publicar una tienda de ejemplo, debe haber una instancia de WebSphere Commerce Server. A continuación, cuando cree una tienda basada en una tienda de ejemplo utilizando las herramientas del programa de utilidad de publicación de la Consola de administración, se crearán los datos básicos. Las tiendas de ejemplo incluyen todos los datos de configuración necesarios y la mayor parte de los datos gestionados requeridos para una tienda en funcionamiento. Al crear tiendas basadas en determinadas tiendas de ejemplo, es posible que se le indique que realice algunos pasos de configuración de datos, utilizando las herramientas de WebSphere Commerce Accelerator.

Herramientas para la creación de datos

WebSphere Commerce proporciona varias herramientas para crear y manipular los datos de su tienda. Estas herramientas se listan a continuación:

Loader package de WebSphere Commerce

Loader package consta principalmente de utilidades para preparar y cargar datos en una base de datos de WebSphere Commerce. Para obtener más información, consulte la Parte 10, "Publicación de la tienda", en la página 345.

Consola de administración

La Consola de administración le permite controlar el sitio o la tienda realizando operaciones administrativas y tareas de configuración. También puede utilizar la Consola de administración para crear nuevas organizaciones y usuarios, así como para asignar usuarios a roles. La Consola de administración también le permite identificar los tipos de mensajes y notificaciones que estarán disponibles en la tienda. La Consola de administración contiene el programa de utilidad de publicación que le permite publicar negocios y tiendas de ejemplo.

WebSphere Commerce Accelerator

WebSphere Commerce Accelerator consta de diversas herramientas en línea que le permiten crear y efectuar el mantenimiento de varios elementos de la tienda. Se puede crear y gestionar una gran parte de los datos de la tienda utilizando las herramientas de WebSphere Commerce Accelerator. Para obtener más información, consulte el "Diagrama de herramientas y datos de la tienda" en la página 113.

Consola de administración de organizaciones

La Consola de administración de organizaciones le permite crear y gestionar las organizaciones que acceden al sitio o la tienda. La Consola de administración de organizaciones también permite al administrador de compradores gestionar los compradores de la organización.

Diagrama de herramientas y datos de la tienda

El diagrama siguiente lista las herramientas que puede utilizar para crear cada tipo de datos.

Herramientas para crear datos	Datos básicos	Datos de configuración	Datos gestionados	Datos de operaciones
Loader package de WebSphere Commerce	Utilice Loader package para cargar datos básicos en forma de un archivo XML. Para obtener más información, consulte "Creación de elementos de datos de tienda en un archivo XML" en la página 136.	Utilice Loader package para cargar datos de configuración en forma de un archivo XML. Para obtener más información, consulte "Creación de un archivo XML para registrar mandatos, vistas y URL" en la página 151.	Utilice Loader package para cargar datos gestionados en forma de un archivo XML. Para obtener más información, consulte los capítulos correspondientes a los elementos de datos gestionados.	En general, los datos operativos no se pueden cargar con Loader package. No obstante, los datos del cliente seleccionado se pueden cargar utilizando Loader package.
Consola de administración	Cuando publica un archivador de tienda utilizando la Consola de administración, se crean los datos básicos automáticamente. Para obtener más información sobre cómo utilizar el programa de utilidad de publicación, consulte la Ayuda en línea a la producción de WebSphere Commerce.	No aplicable.	No aplicable.	No aplicable.

Herramientas para crear datos	Datos básicos	Datos de configuración	Datos gestionados	Datos de operaciones
WebSphere Commerce Accelerator	No aplicable.	No aplicable.	<p>Utilice WebSphere Commerce Accelerator para crear o editar los siguientes datos:</p> <ul style="list-style-type: none"> • Campañas • Contratos (debe haber un contrato por omisión en la base de datos para poder utilizar las herramientas de Gestión de relaciones comerciales de WebSphere Commerce Accelerator para crear contratos adicionales o cambiar los ya existentes. Utilice Loader package, el Asistente de creación de tiendas o publique un archivador de tienda para crear un contrato por omisión en la base de datos. • Jurisdicciones • Impuestos • Envío • Moneda • Idiomas 	<p>Los clientes crean datos de operaciones cuando se registran en la tienda o efectúan compras. Sin embargo, en algunos casos, puede utilizar WebSphere Commerce Accelerator para realizar pedidos para un cliente o para crear una devolución.</p> <p>WebSphere Commerce Accelerator también le permite gestionar el inventario (recepciones y recepciones esperadas).</p>

Herramientas para crear datos	Datos básicos	Datos de configuración	Datos gestionados	Datos de operaciones
WebSphere Commerce Accelerator (continuación)	No aplicable.	No aplicable.	<ul style="list-style-type: none"> • Despacho de pedidos • Descuentos • Catálogos (debe haber un catálogo maestro en la base de datos para poder utilizar las herramientas de Gestión de productos de WebSphere Commerce Accelerator para añadir o cambiar la información de los productos. Para obtener una vista diferente de la mercancía y de los servicios, cree un catálogo de ventas cambiando el fuente XML. Utilice Loader package o publique un archivador de tienda para crear un catálogo maestro en la base de datos.) • Precios 	No aplicable.

Herramientas para crear datos	Datos básicos	Datos de configuración	Datos gestionados	Datos de operaciones
Consola de administración de organizaciones	Utilice la Consola de administración de organizaciones para crear y gestionar organizaciones.	No aplicable.	No aplicable.	Los clientes y compradores se crean cuando entran en la tienda. No obstante, utilizando la Consola de administración de organizaciones, también puede gestionar usuarios y aprobar compradores o crear nuevos.

Parte 6. Desarrollo de los datos de la tienda

Los capítulos de esta sección explican cada uno de los elementos de datos de la tienda con más detalle. Estos elementos de datos se han organizado según la estructura de la arquitectura de datos de la tienda de WebSphere Commerce:

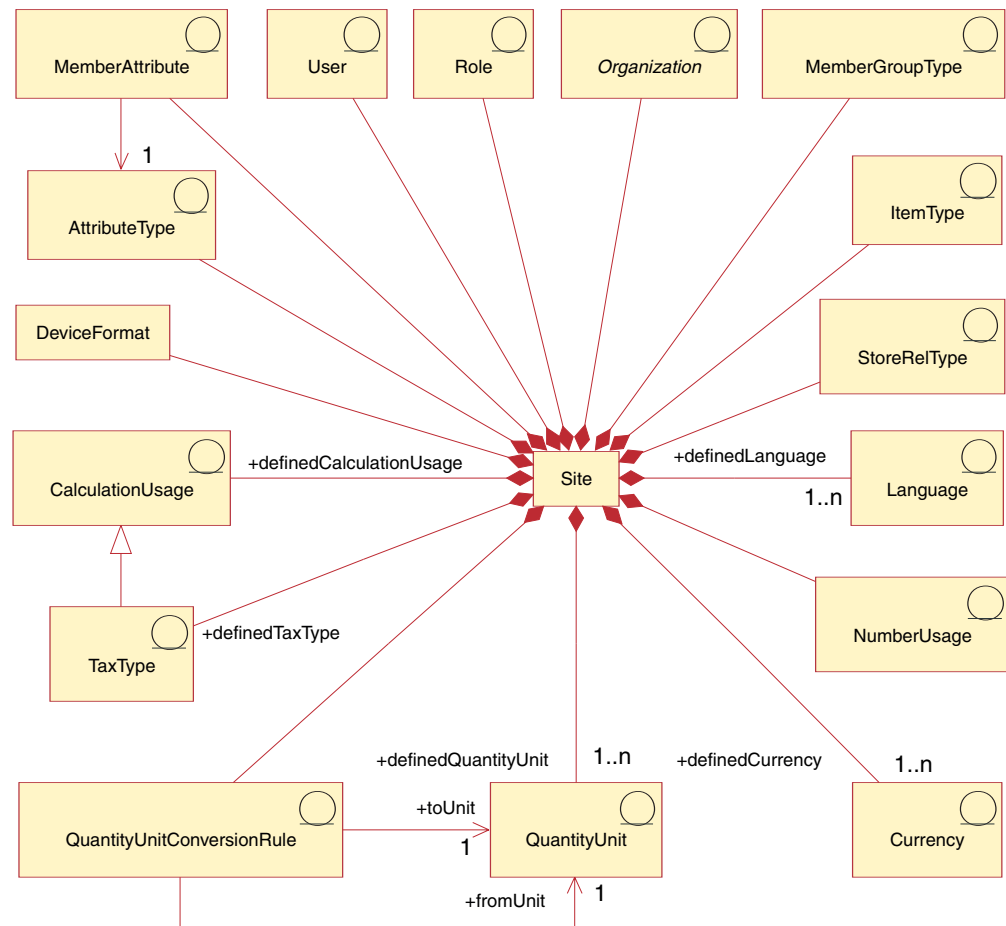
- Instancia del servidor WebSphere Commerce
 - Sitio
- Datos básicos
 - Organización
 - Tienda
 - Relaciones entre tiendas
- Datos de configuración
 - Registro de mandatos
 - Registro de vistas
 - Registro de URL
- Datos gestionados
 - Catálogo
 - Precios
 - Contratos (incluidas las políticas de negocio)
 - Despacho de pedidos
 - Campañas
 - Pago
 - Idiomas soportados
 - Monedas soportadas
 - Unidades de medida soportadas
 - Jurisdicciones
 - Envío
 - Impuestos
 - Descuentos
- Datos de operaciones
 - Inventario
 - Pedidos
 - Clientes
 - Subastas
 - RFQ

Capítulo 11. Elementos de sitio

Cada instancia de WebSphere Commerce Server tiene su propia base de datos de información relacional. Los archivos de rutina de carga crean una instancia e insertan información en las tablas de base de datos, después de que se cree el esquema. Una vez que se han cargado los datos, puede ver la información cargada previamente en la tabla de base de datos apropiada. Muchas de las tablas de base de datos contienen información, a nivel de tienda o de grupo de tiendas, específica de una tienda o grupo de tiendas. Algunas tablas contienen información que representa las funciones a nivel de sitio de WebSphere Commerce que todas las tiendas de la instancia pueden utilizar. Toda esta información la gestiona el administrador de sitio de WebSphere Commerce. En este capítulo se describen estas funciones. Para obtener más información acerca de los archivos de rutina de carga, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce. Para obtener información acerca de los elementos específicos de tienda, consulte el Capítulo 13, "Elementos de tienda", en la página 135.

Los elementos de sitio en WebSphere Commerce

El diagrama siguiente ilustra los tipos de datos que contiene el sitio y su relación con el sitio.





Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, “Descripción de UML”, en la página 463. Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce. Para obtener más información sobre el modelo de información, consulte “Modelo de información de los datos de la tienda” en la página 105.

Idioma

Un sitio puede tener muchos *idiomas* definidos en la tabla LANGUAGE y éstos pueden estar descritos en la tabla LANGUAGEDS. Normalmente, cada tienda da soporte a un subconjunto de estos idiomas añadiendo filas a la tabla STORELANG. Los diez idiomas definidos previamente son: alemán, chino tradicional y chino simplificado, japonés, coreano, italiano, francés, español, portugués de Brasil e inglés.

Atributos de miembros

Los *atributos de los miembros* se almacenan en la tabla MBRATTR y representan el conjunto de nombres de atributo definidos para los que se pueden almacenar valores para organizaciones o usuarios. Algunos ejemplos de nombres de atributo son JobFunction, ProcurementCard, SpendingLimit, ReferredBy y CountryOfOperation. Los valores de los atributos para organizaciones o usuarios específicos se almacenan en la tabla MBRATTRVAL, y estos valores pueden ser distintos para diferentes tiendas o grupos de tiendas.

Tipos de atributo

Los *tipos de atributo* se almacenan en la tabla ATTRTYPE y representan los tipos de datos definidos que se pueden utilizar para representar los valores de los atributos. Algunos ejemplos de tipos de datos son INTEGER, STRING y FLOAT.

Tipos de grupos de miembros

Los *tipos de grupos de miembros* se almacenan en la tabla MBRGRPTYPE y representan el conjunto de usos del grupo de miembros definido. Los usos se asignan a los grupos de miembros añadiendo filas a la tabla MBRGRPUSG. Algunos ejemplos de usos de grupo de miembros son AccessGroup (para usar con las políticas de control de acceso) y UserGroup (para propósitos generales como, por ejemplo, grupos de clientes).

Usuario

El *usuario* representa la identidad autenticada del usuario. Normalmente, los usuarios representan a clientes formalizando o aprobando pedidos en nombre de organizaciones compradoras, agentes de venta procesando los pedidos para las organizaciones vendedoras o efectuando el mantenimiento de elementos a nivel de tienda, o administradores de sitio efectuando el mantenimiento de la instancia de WebSphere Commerce Server. Cada usuario está asociado a un sitio y definido en la tabla USERS.

Organización

La *organización* representa a organizaciones y unidades de organización dentro de las organizaciones. Normalmente, las organizaciones representan a las entidades de negocio responsables de comprar y vender. Los pedidos que formalizan los clientes en una organización compradora Directo a B2B, se registran como si las

formalizara la organización compradora. Las tiendas, los catálogos y los centros de despacho de pedidos son propiedad de las organizaciones responsables de ciertos aspectos de venta. Las organizaciones están definidas en la tabla ORGENTITY.

Rol

El *rol* representa el conjunto de roles definidos que se pueden asignar a los usuarios dentro de las organizaciones. Por ejemplo, a un usuario se le puede asignar el rol de representante de atención al cliente dentro de una organización vendedora, o el rol de aprobador de compradores dentro de una organización compradora. Los nombres y las descripciones de los roles por omisión se insertan en la tabla ROLE. Para obtener más información sobre roles específicos, consulte la ayuda en línea de WebSphere Commerce.

Conversión de unidades de cantidad

Todos los sitios tienen *conversiones de cantidad*. Estas conversiones representan operaciones de multiplicación o división que se utilizan para la conversión entre distintas unidades de medida. Se almacenan en la tabla QTYCONVERT.

Unidades de cantidad

Las *unidades de cantidad* representan el conjunto de unidades de medida para el sitio. Están definidas en la tabla QTYUNIT y descritas en la tabla QTYUNITDSC. Cada tienda puede especificar cómo se redondea y formatea cada unidad de medida para su visualización, según el uso que se le quiera dar, añadiendo filas a la tabla QTYFORMAT.

Tipos de impuestos

Los *tipos de impuestos* representan usos de cálculo para calcular impuestos. Los impuestos sobre la venta y los impuestos sobre el envío son dos métodos de cálculo distintos que calculan impuestos. Los tipos de impuestos están definidos en la tabla TAXTYPE.

Uso de cálculo

Los *usos de cálculo* representan los diferentes tipos de cálculos que puede efectuar el mandato OrderPrepare. Los usos de cálculo se definen para los descuentos, costes de envío, impuestos sobre la venta, impuestos sobre el envío y cupones de venta electrónica. Los usos de cálculo están definidos en la tabla CALUSAGE.

Moneda

Todo sitio tiene un número de *monedas* definido en la tabla SETCURR y descrito en la tabla SETCURRDSC. Cada tienda da soporte a un subconjunto de estas monedas añadiendo filas a la tabla CURLIST, una fila para cada moneda soportada.

Nota: Para algunos de los elementos de sitio, como por ejemplo, el idioma, la moneda, la unidad de la cantidad y la norma de conversión de la unidad de la cantidad, el Administrador de sitio puede ampliar las posibilidades a nivel de sitio añadiendo filas a las tablas adecuadas. Para otros, pueden necesitarse personalizaciones relacionadas para ampliar las posibilidades a nivel de sitio que representan. Si un Administrador de sitio ha añadido una nueva utilización de números para mostrar subtotales con un símbolo de moneda personalizado, el programa que muestra los subtotales deberá personalizarse para especificar la nueva utilización de números para el valor de subtotal cuando se formateen los importes de subtotal para su visualización.

Utilización de números

La *utilización de números* representa cómo se van a utilizar los números. Las tiendas pueden especificar distintas normas de redondeo y formateo de los números que se visualizan, según la forma en que se utilicen. Por ejemplo, una tienda puede redondear los precios por unidad con cuatro decimales especificando la utilización de "precio unitario", y redondear otros importes en dos decimales especificando la utilización "por omisión". La utilización de números se define en la tabla NUMBRUSG y se describe en la tabla NUMBRUSGDS.

Tipos de artículo

Los *tipos de artículo* representan las diferentes clases de artículos. Los dos tipos de artículo base de WebSphere Commerce son kits dinámicos y artículos normales. Los tipos de artículos están definidos previamente en la tabla ITEMTYPE. Para obtener más información sobre los artículos base, consulte el Capítulo 29, "Elementos de inventario", en la página 289.

Formatos de dispositivo

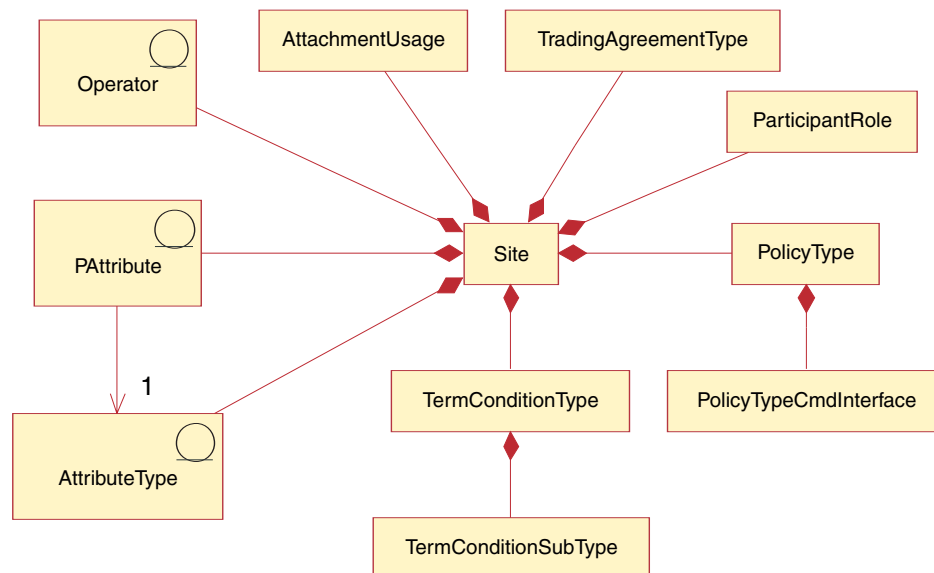
Los *formatos de dispositivo* se almacenan en la tabla DEVICEFMT y representan los diversos tipos de formato de dispositivo que utiliza un sitio como, por ejemplo, navegadores, I_MODE, correo electrónico, XMLMQ y XMLHTTP. Todos estos tipos de dispositivo permiten a los usuarios interactuar con el sitio a través de diferentes medios.

Tipos de relaciones entre tiendas

Un *tipo de relación entre tiendas* (StoreRelType) define el tipo de relación que existe entre dos tiendas. Cada tipo de relación entre tiendas define su propia relación, es decir, los roles que representará cada parte en la relación y la relación existente entre ambas. Un tipo de relación entre tiendas se define en la tabla STRELTYP y se describe en la tabla STRELTYPDS.

Datos de acuerdo comercial a nivel de sitio

El diagrama siguiente ilustra los tipos de datos de acuerdo comercial que contiene el sitio y su relación con el sitio.





Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, “Descripción de UML”, en la página 463. Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce. Para obtener más información sobre el modelo de información, consulte “Modelo de información de los datos de la tienda” en la página 105.

Tipo de acuerdo comercial

WebSphere Commerce proporciona varios mecanismos de comercio que rigen las interacciones entre compradores y vendedores. Un acuerdo comercial representa una instancia de un mecanismo de comercio y registra las propiedades de esa instancia de un mecanismo de comercio. Cada contrato, cuenta de negocio y RFQ en WebSphere Commerce está representado por un acuerdo comercial. Hay un acuerdo comercial individual que rige todas las subastas en WebSphere Commerce. WebSphere Commerce da soporte a varios *tipos de acuerdos comerciales*, incluido la cuenta, el contrato, las RFQ, el intercambio y las subastas. Los tipos de acuerdo comercial se definen en la tabla TRDTYPE. Para obtener más información sobre los acuerdos comerciales, consulte el Capítulo 18, “Elementos de contrato”, en la página 195.

Rol de participante

Los participantes de los acuerdos comerciales adoptan roles específicos en cada acuerdo comercial. WebSphere Commerce da soporte a varios *roles de participante*, incluido el creador, vendedor, comprador, proveedor, aprobador, administrador, distribuidor, proveedor de servicio, revendedor, proveedor de alojamiento y destinatario. Los roles de participantes se definen en la tabla PARTROLE.

Tipo de política

WebSphere Commerce da soporte a varios tipos de políticas de negocio, incluido de precio, conjunto de productos, modalidad de envío, costes de envío, pagos y otras. Los tipos de políticas se definen en la tabla POLICYTYPE. Para obtener más información sobre las políticas de negocio, consulte el Capítulo 18, “Elementos de contrato”, en la página 195.

Interfaz del mandato de tipo de política

La *interfaz del mandato de tipo de política* es la interfaz del mandato Java para el objeto de política de negocio. El mandato para cada instancia de política debe implementar esta interfaz. Puede haber cero o más mandatos para cada objeto de política de negocio.

Tipo de términos y condiciones

Los términos y condiciones definen el comportamiento y las propiedades de un acuerdo comercial. WebSphere Commerce da soporte a varios *tipos de términos y condiciones*, incluyendo de fijación de precios, de pago y de envío. Los tipos de términos y condiciones se definen en la tabla TCTYPE. Para obtener más información sobre términos y condiciones, consulte el Capítulo 18, “Elementos de contrato”, en la página 195.

Subtipo de términos y condiciones

Cada tipo de términos y condiciones puede contener varios *subtipos de términos y condiciones*. Los subtipos de términos y condiciones se definen en TCSUBTYPE.

Atributo de personalización

El *atributo de personalización* le permite crear atributos para productos. El atributo de personalización se define en la tabla PATTRIBUTE. Cada atributo de personalización tiene solamente un tipo de atributo.

Tipo de atributo

El *tipo de atributo* define el tipo del atributo. Los tipos de atributos se definen en la tabla ATTRTYPE.

Operador

Los *operadores* utilizados en el sitio son el operador sencillo (permite un sólo valor), el operador compuesto (rango, continuo) y el operador compuesto (conjunto). Los operadores se definen en la tabla OPERATOR.

Uso de adjuntos

Un adjunto es un documento de soporte para un documento de intercambio. Por ejemplo, puede ser una especificación de un producto o una hoja de cálculo con una lista de precios. *Uso de adjuntos* describe cómo y dónde se utilizarán los adjuntos. El uso de adjuntos se define en la tabla ATTACHUSG.

Creación de elementos de sitio en WebSphere Commerce

Al crear una instancia en WebSphere Commerce Server se crean los elementos de sitio. Para obtener más información acerca de cómo crear una instancia en WebSphere Commerce Server, consulte en la publicación *WebSphere Commerce, Guía de instalación*, el apartado "Creación de una instancia de WebSphere Commerce".

Capítulo 12. Elementos de miembro

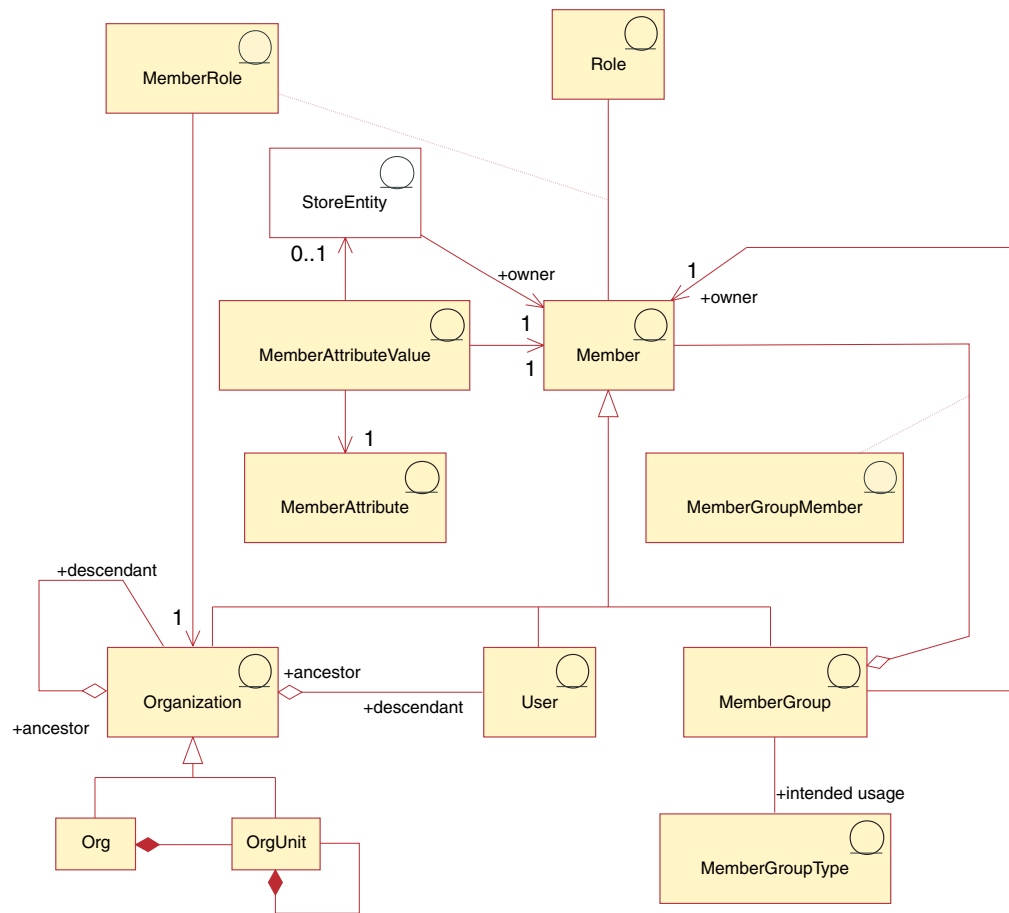
En este capítulo se describe en primer lugar el Subsistema de miembros de WebSphere Commerce, a continuación, se describen los tres tipos de miembros relacionados con los desarrolladores de tiendas: clientes, vendedores y administradores. Tenga en cuenta que WebSphere Commerce proporciona un Subsistema de miembros, que incluye miembros o usuarios y organizaciones.

Los elementos de miembro en WebSphere Commerce

Los elementos de miembro de WebSphere Commerce incluyen datos para los participantes del sistema WebSphere Commerce. Un miembro puede ser un usuario, un grupo de usuarios o una entidad de organización. Un administrador, por ejemplo un Administrador de sitio, asigna roles a los usuarios y a los miembros de la entidad de organización. Una vez que se ha asignado un rol a un miembro, el componente de control de acceso autoriza al miembro a participar en las actividades asociadas al rol. Por ejemplo, una organización puede ser un comprador, un vendedor o ambos. A un usuario también se le pueden asignar varios roles. Un administrador puede crear grupos de miembros, que son grupos de usuarios clasificados en categorías por diversas razones del negocio. Utilice la Consola de administración de WebSphere Commerce para crear organizaciones, usuarios, roles y grupos de miembros, y trabajar con ellos.

La lógica de negocio para los elementos de miembro proporciona servicios de gestión de perfiles y de registro de miembros. Otros servicios que están estrechamente relacionados con los elementos de miembro incluyen el control de acceso, la autenticación y la gestión de sesiones. Para obtener información detallada acerca de estos temas, consulte la ayuda en línea para desarrollo de WebSphere Commerce.

El diagrama siguiente ilustra los elementos de miembro de WebSphere Commerce. A continuación del diagrama, se proporcionan descripciones de cada elemento.



Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce Server. Para obtener más información sobre el modelo de información, consulte "Modelo de información de los datos de la tienda" en la página 105. Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, "Descripción de UML", en la página 463.

Miembros

Un *miembro* en WebSphere Commerce puede ser cualquiera de los siguientes:

- Una *entidad de organización*. Puede ser una organización, por ejemplo "IBM", o una unidad de organización dentro de una gran organización, por ejemplo la "División de Comercio electrónico" dentro de IBM.
- Un *usuario* (registrado o no registrado). Un usuario registrado tiene un identificador exclusivo y una contraseña que son necesarios para proporcionar los datos de perfil para el registro. Los usuarios registrados pueden clasificarse de acuerdo con el tipo de perfil: el tipo 'B' indica un usuario de empresa (o un cliente de Directo a B2B) y el tipo 'C' indica un usuario minorista (o un cliente de Directo al consumidor). Para obtener más información sobre usuarios registrados y no registrados, consulte "Miembros" en la ayuda en línea para desarrollo de WebSphere Commerce.

- Un *grupo de miembros*. Se trata de un grupo de usuarios clasificados por categorías por diversas razones del negocio. Las agrupaciones pueden utilizarse para el control de acceso, para las aprobaciones así como para el marketing (por ejemplo calcular descuentos y precios, y visualizar productos).

Cada *entidad de tienda* (es decir, una tienda o un grupo de tiendas) es propiedad de un miembro.

Atributos de miembro

Un miembro de WebSphere Commerce tiene un conjunto de *atributos* y cada atributo tiene un *valor* asociado a él. Un perfil de usuario básico para un miembro incorpora información de registro, datos estadísticos, información de dirección, historial de compras y otros atributos diversos.

Un perfil de usuario de empresa contiene la misma información que un perfil de usuario básico, así como información de empleo, por ejemplo un número de empleado, un puesto de trabajo o una descripción de trabajo. Durante el registro, los usuarios de empresa deben identificar la organización de negocio a la que pertenecen. Los perfiles para las entidades de organización incluyen esta información adicional, por ejemplo el nombre y categoría de negocio de la organización.

Las normas de control de acceso imponen la autorización de usuario para la gestión de perfiles. Los perfiles de miembros pueden contener diversos atributos relacionados con el negocio y personales (por ejemplo, roles, información de pago, direcciones, monedas e idiomas preferidos y dispositivos de tipo "pervasive computing"). Los atributos pueden ser sensibles a la tienda. Se da soporte a estos atributos para usuarios y entidades de organización pero no así para grupos de miembros.

Roles

Cada usuario puede desempeñar uno o más roles para una organización. Un Administrador de sitio asigna un rol o roles a cada miembro. Por ejemplo, como miembro de la organización IBM, el rol de Juan Pérez como Representante de servicio al cliente significa que Juan realiza tareas para los clientes de IBM y les ayuda con las consultas y los asuntos relacionados con la información de registro, los pedidos o las devoluciones. Puede que Juan también tenga el rol de Supervisor de servicio al cliente, que tiene todas las responsabilidades de las tareas descritas anteriormente, así como la autorización de aprobaciones y de supervisión de otros Representantes de servicio al cliente.

El sistema WebSphere Commerce proporciona el siguiente conjunto de tipos de roles por omisión:

- Roles de relaciones comerciales
- Roles de servicio al cliente
- Roles de marketing
- Roles operativos
- Roles de gestión de organización
- Roles de gestión de productos y comercialización
- Roles de operaciones técnicas

Para obtener detalles sobre cada uno de estos roles, consulte en la ayuda en línea de WebSphere Commerce el tema "Roles". Un Administrador de sitio puede asignar estos roles, así como cualquier rol nuevo creado por el Administrador de

sitio, por entidad de organización; es decir, los usuarios que pertenecen a una entidad de organización pueden asumir los roles asignados a dicha entidad de organización.

Cuando se asigna un rol a un usuario, el rol se establece en el ámbito de la entidad de organización. Ésta puede ser una entidad de organización y no necesariamente tiene que ser uno de los antecesores del usuario. No obstante, dado que los roles se heredan, el usuario desempeñará el rol asignado a cualquier descendiente de la organización para la que se asigna el rol. Por ejemplo, si se asigna un rol a un usuario en la Organización raíz, entonces el usuario desempeñará dicho rol para todas las entidades de organización.

Los roles de WebSphere Commerce se pueden asignar manualmente mediante la Consola de administración de organizaciones, y de forma automática mediante los mandatos de registro y gestión de sesiones. Esta asignación de roles automática esta basada en los valores de configuración especificados en el archivo MemberRegistrationAttributes.xml. WebSphere Commerce 5.5 proporciona el archivo MemberRegistrationAttributes.xml, que se puede modificar para adaptarlo a requisitos de registro concretos. Para obtener más información sobre la asignación de roles automática y el archivo MemberRegistrationAttributes.xml, consulte el tema de la ayuda en línea para desarrollo de WebSphere Commerce "Archivos XML y DTD de MemberRegistrationAttributes".



Para obtener información más detallada sobre la estructura de los elementos de miembro en WebSphere Commerce, consulte los modelos de datos y de objeto de miembro en la ayuda en línea de WebSphere Commerce.

Los elementos de cliente en WebSphere Commerce

En WebSphere Commerce, un *cliente* es un usuario. Un cliente puede navegar por el catálogo en línea de la tienda, formalizar un pedido, crear una lista de intereses, configurar direcciones (por ejemplo, para contactos generales, facturación, envío) y comprar en la tienda o al vendedor. Un cliente también es un usuario. El diagrama siguiente ilustra los elementos que un cliente necesita para formalizar un pedido en una tienda.

Como se muestra en el diagrama anterior, el sistema WebSphere Commerce contiene miembros. A cada usuario o entidad de organización se le puede asignar un rol.

Nota: En WebSphere Commerce, un *miembro* puede ser una entidad de organización, un usuario o un grupo de miembros. Consulte el apartado "Miembros" en la página 126 para obtener más información.

En este caso, el usuario es un cliente. Un cliente debe proporcionar información de dirección y puede tener una lista de artículos de interés. El diagrama ilustra la relación recíproca entre un miembro (cliente) y los elementos de cliente asociados a él: un cliente debe proporcionar una dirección de la cual debe ser propietario y puede tener una lista de intereses para comprar en una tienda; la dirección y la lista de intereses dependen de la existencia de un cliente.

Información de dirección

Un cliente debe proporcionar tres tipos de información de dirección cuando realice compras en una tienda: la dirección de la persona de contacto, la dirección de

facturación y la dirección de envío. A continuación, se describen estos tipos de direcciones; cada dirección puede ser exclusiva o puede ser la misma:

- La dirección de la persona de contacto se utiliza para fines diversos; por ejemplo, en relación al estado o a los cambios de un pedido y para avisar sobre los próximos eventos de la tienda (promociones o mantenimiento de la tienda). La dirección de la persona de contacto del cliente incluye el nombre y el número de la calle, la ciudad, el estado o la provincia, el código postal, el país o la región, la dirección de correo electrónico, el número de teléfono y el número de fax. Normalmente, la dirección de la persona de contacto es el lugar donde se puede localizar al cliente más fácilmente, por ejemplo la dirección del trabajo.
- La dirección de facturación se utiliza para enviar un recibo o factura de las compras realizadas. Una dirección de facturación incluye el nombre y el número de la calle, la ciudad, el estado o la provincia, el código postal, el país o la región, el número de teléfono y la dirección de correo electrónico. La dirección de facturación puede ser la misma que la dirección de la persona de contacto o la dirección de envío, o puede ser una dirección distinta.
- La dirección de envío se utiliza para la entrega de los artículos adquiridos. Una dirección de envío incluye el nombre y el número de la calle, la ciudad, el estado o la provincia, el código postal, el país o la región, el número de teléfono y la dirección de correo electrónico. La dirección de envío puede ser la misma que la dirección de la persona de contacto o la dirección de facturación, o puede ser una dirección distinta.

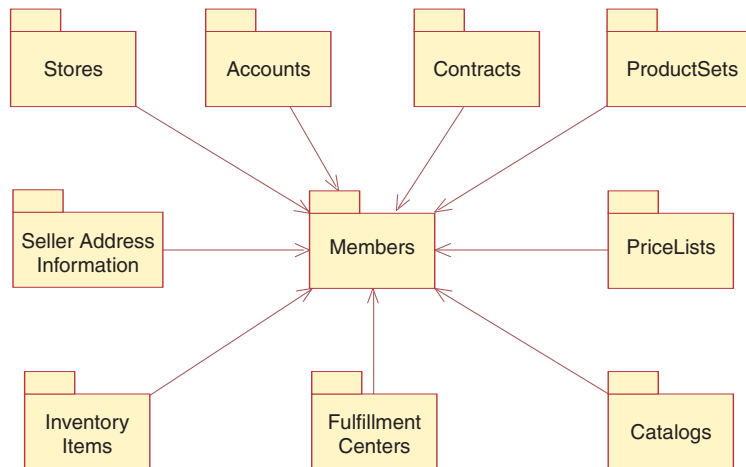
Listas de intereses

Las tiendas pueden soportar *listas de intereses*. Es decir, los clientes añaden a sus listas de intereses productos que desean solicitar en el futuro. Una lista de intereses no es un carro de la compra; una lista de intereses puede contener artículos de varias tiendas y no contiene los precios, las direcciones de envío, las modalidades de envío, la disponibilidad de inventario ni los importes calculados como, por ejemplo, los descuentos, los costes de envío y los impuestos.

Descripción de los elementos de vendedor en WebSphere Commerce

Un *vendedor* es un usuario en WebSphere Commerce. El vendedor supervisa la gestión y los objetivos generales de la tienda, además de hacer el seguimiento de las ventas de la tienda. El vendedor vende los artículos y los servicios al cliente. El rol de vendedor es equivalente al de comerciante y tiene acceso a todas las funciones de WebSphere Commerce Accelerator. El diagrama siguiente ilustra los elementos que un vendedor necesita para mantener una tienda y vender a los

clientes.



Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce Server. Para obtener más información sobre el modelo de información, consulte “Modelo de información de los datos de la tienda” en la página 105. Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, “Descripción de UML”, en la página 463.

Como se muestra en el diagrama anterior, el sistema WebSphere Commerce contiene miembros. Cada miembro tiene asignado un rol, por ejemplo un Representante de servicio al cliente para la tienda o un receptor de un almacén. El rol de vendedor puede mantener los elementos siguientes para poder vender a los clientes:

- Tiendas
- Cuentas (opcional)
- Contratos (o al menos el contrato por omisión de WebSphere Commerce)
- Conjuntos de productos
- Listas de precios
- Catálogos
- Centros de despacho de pedidos
- Artículos de inventario

El diagrama anterior ilustra la relación entre un miembro (vendedor) y los elementos de vendedor asociados, es decir, un vendedor puede tener los elementos listados anteriormente para mantener una tienda y los elementos necesitan tener un vendedor que los despliegue.

Tiendas

Una *tienda* en línea de WebSphere Commerce está compuesta por un conjunto de archivos HTML y JavaServer Pages, así como de elementos de impuestos, envío, pago, catálogo y otros elementos de base de datos, que están contenidos en un archivador de tienda. Una tienda también contiene datos de tienda, que es la información con la que se llena la base de datos de WebSphere Commerce para permitir que una tienda funcione.

Para obtener más información sobre las tiendas de WebSphere Commerce, consulte el Capítulo 13, “Elementos de tienda”, en la página 135 y la Parte 6, “Desarrollo de los datos de la tienda”, en la página 117.

Cuentas

Una tienda puede configurar *cuentas* de negocio para los clientes a fin de permitirles comprar en la tienda. Una cuenta contiene la información siguiente:

- El nombre de la cuenta, que es normalmente el nombre de la organización a la que está asociado el cliente. Esta organización tiene contratos definidos con la tienda, que estipulan los términos para que el cliente compre en la tienda. Por ejemplo, la organización IBM puede tener contratos con la Compañía de suministros de oficina ABC.
- El nombre del representante, que es el nombre de la organización representante dentro de la organización del vendedor que es responsable de la cuenta.
- El número de contratos que pertenecen a la cuenta.

Para obtener más información sobre las cuentas de WebSphere Commerce, consulte el apartado “Cuentas (cuentas de negocio)” en la página 196 y la ayuda en línea de WebSphere Commerce.

Contratos

Normalmente, en WebSphere Commerce todos los clientes deben comprar bajo un *contrato*. Cada cuenta entre el cliente y el vendedor debe estar asociada a uno o más contratos (o al menos un *contrato por omisión* para que los clientes, registrados o no, compren en la tienda o si se desea que los clientes puedan comprar productos no cubiertos por otros contratos). Un contrato permite al cliente comprar productos en una tienda a un precio especificado durante un periodo de tiempo especificado, bajo los términos y condiciones y las políticas de negocio que se estipulan en el contrato. El vendedor despliega el contrato para que los clientes puedan comprar en la tienda.

El comprador de un contrato puede ser un usuario, una organización o un grupo de miembros. En el caso del usuario, el comprador se considera el cliente. En el caso de una organización, que se define como un comprador en un contrato, cualquier hijo de esta organización puede actuar como comprador para el contrato. En el caso de un grupo de miembros, cualquier usuario del grupo de miembros puede actuar como comprador para el contrato.

Para obtener más información sobre los contratos de WebSphere Commerce y el contrato por omisión que un vendedor puede utilizar, consulte el apartado “Contratos” en la página 197.

Conjuntos de productos

Los *conjuntos de productos* proporcionan un mecanismo para que un vendedor clasifique los catálogos en línea en categorías de subconjuntos lógicos a fin de que un vendedor pueda permitir que diversos clientes se beneficien de diferentes vistas de catálogo. Además, un vendedor puede crear un contrato para un cliente y estipular que el cliente sólo pueda comprar productos bajo un conjunto de productos predefinido.

Para obtener más información sobre los conjuntos de productos de WebSphere Commerce, consulte “Conjuntos de productos” en la página 158.

Listas de precios

Una *lista de precios* está asociada al precio que un vendedor ofrece o presenta a un cliente. Un vendedor puede listar precios diferentes para el mismo producto para clientes diferentes. En WebSphere Commerce, una oferta de precio se conoce también como *propuesta de comercio* y representa el precio de una entrada de catálogo y los criterios que el cliente debe satisfacer para poder pagar dicho precio.

En WebSphere Commerce, un objeto Oferta forma parte de un contenedor de propuestas de comercio, que es propiedad de un miembro. Un contenedor de propuestas de comercio contiene propuestas de comercio y puede dejarse disponible para todos los clientes o sólo para clientes de determinados grupos mediante los acuerdos comerciales o los contratos. A veces un contenedor de propuestas de comercio (TradingPositionContainer) se denomina lista de precios. Existen dos clases de listas de precios: una lista de precios estándar que contiene los precios base para los productos del catálogo de tienda o una lista de precios personalizada que especifica la lista de productos y sus precios personalizados.

Para obtener más información sobre las listas de precios de WebSphere Commerce, consulte el Capítulo 17, "Elementos de fijación de precios", en la página 187.

Catálogos

Una tienda de WebSphere Commerce utiliza como mínimo un *catálogo* en línea para exponer las mercancías y los servicios que el vendedor tiene en venta. Normalmente un catálogo en línea contiene precios, imágenes y descripciones de los artículos en venta. Un catálogo en línea también puede presentar mercancías en categorías diferentes que faciliten la navegación.

Cada tienda del sistema WebSphere Commerce debe tener un *catálogo maestro*, que se utiliza para la gestión de catálogos. El catálogo maestro es la ubicación central para gestionar la mercancía de un vendedor; es el catálogo individual que contiene todos los productos, artículos, relaciones y precios estándar para todo lo que está en venta en la tienda. Si un vendedor tiene más de una tienda, el catálogo maestro puede compartirse entre estas tiendas.

Para obtener más información sobre los conjuntos de productos de WebSphere Commerce, consulte el Capítulo 16, "Elementos de catálogo", en la página 155.

Centros de despacho de pedidos

Los *centros de despacho de pedidos* los utilizan las tiendas como almacenes de inventario y como centros de envío y recepción. Un vendedor puede tener uno o muchos centros de despacho de pedidos asociados.

Desde la perspectiva de un servidor WebSphere Commerce, un objeto Centro de despacho de pedidos es independiente del objeto Tienda. Gestiona el inventario y el envío de los productos. Para enviar un pedido, el centro de despacho de pedidos se basa en el objeto Modalidad de envío especificado por el cliente. El objeto Modalidad de envío indica la empresa de transportes y el método de envío para despachar los pedidos. En un centro de despacho de pedidos, el objeto Acuerdo de envío indica que un objeto Tienda ha dispuesto con un objeto Centro de despacho de pedidos enviar los productos utilizando una determinada modalidad de envío.

Para obtener más información sobre los centros de despacho de pedidos de WebSphere Commerce, consulte el Capítulo 19, "Elementos de despacho de pedidos", en la página 215 y el Capítulo 26, "Elementos de envío", en la página 249.

Artículos de inventario

Los *artículos de inventario* incluyen todo aquello que se puede contabilizar físicamente en el centro de despacho de pedidos de un vendedor. El sistema WebSphere Commerce define tipos específicos de inventario que se pueden despachar, por ejemplo artículos, productos, códigos de artículo, paquetes compuestos y paquetes; pero todos ellos se consideran inventario. Los productos se configuran para el despacho en las herramientas de Gestión de productos en WebSphere Commerce Accelerator.

Para obtener más información acerca de los elementos de inventario de WebSphere Commerce, consulte la ayuda en línea para desarrollo de WebSphere Commerce y el Capítulo 29, "Elementos de inventario", en la página 289.

Descripción de los elementos de administrador en WebSphere Commerce

Los administradores simplemente son usuarios o miembros con roles asignados que les permiten realizar determinadas actividades administrativas. Consulte el apartado "Los elementos de miembro en WebSphere Commerce" en la página 125 para obtener información más detallada sobre los elementos que se pueden asociar con un administrador.

Creación de elementos de miembro en WebSphere Commerce

Para crear un vendedor (una organización que actúa como propietaria de la tienda) y mantener información acerca del vendedor, utilice la Consola de administración de WebSphere Commerce. Para obtener más información, consulte el tema "Crear una organización" de la ayuda en línea para desarrollo de WebSphere Commerce.

Para crear un administrador, utilice la Consola de administración de WebSphere Commerce para crear el usuario, luego asigne el rol deseado a este usuario. Para obtener más información, consulte los temas "Crear un usuario" y "Asignación de roles por nombre distinguido de usuario" de la ayuda en línea para desarrollo de WebSphere Commerce.

El Desarrollador de tiendas no crea un cliente: cuando un cliente se registra en una tienda, el sistema WebSphere Commerce recopila y mantiene la información de registro.

Las tiendas de ejemplo que se proporcionan con WebSphere Commerce contiene cada una sus propias versiones del archivo MemberRegistrationAttributes.xml, que se utiliza para configurar la asignación de roles automatizada para los mandatos de registro y gestión de sesiones. Si opta por modificar la estructura de la organización o si tiene requisitos determinados para asignar roles, tendrá que modificar este archivo. Consulte el tema de la ayuda en línea para desarrollo de WebSphere Commerce "Archivos XML y DTD MemberRegistrationAttributes" para obtener más información sobre este archivo y cómo configurarlo para que se adapte a sus necesidades.

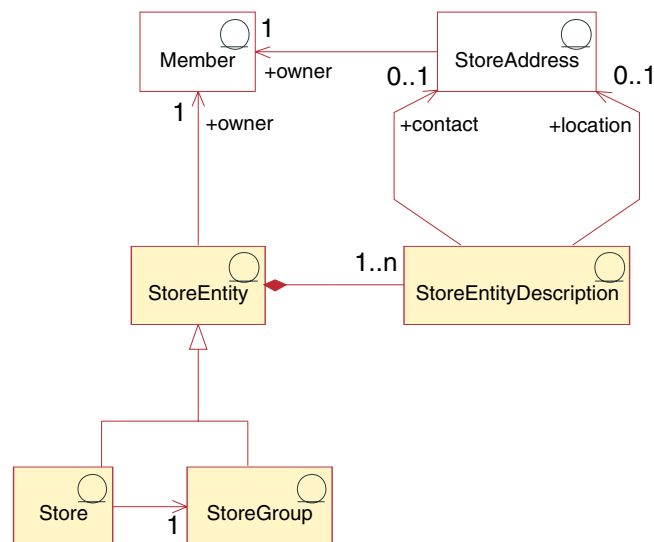
Capítulo 13. Elementos de tienda

Para crear una tienda en WebSphere Commerce, primero debe crear los siguientes elementos en la base de datos:

- la tienda
- el grupo al que pertenece
- el objeto abstracto de entidad de tienda que representa, de forma dual, una tienda o grupo de tiendas.

Los elementos de tienda en WebSphere Commerce

El diagrama siguiente ilustra los elementos de tienda en WebSphere Commerce Server.



Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce Server. Para obtener más información sobre el modelo de información, consulte “Modelo de información de los datos de la tienda” en la página 105. Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, “Descripción de UML”, en la página 463.

Entidad de tienda

Una *entidad de tienda* es una superclase abstracta que puede representar una tienda o un grupo de tiendas.

Una entidad de tienda tiene un propietario (un miembro). Para obtener más información sobre los miembros, consulte “Los elementos de miembro en WebSphere Commerce” en la página 125.

Descripción de la entidad de tienda

La *descripción de la entidad de tienda* describe la entidad de tienda. Una entidad de tienda puede tener una descripción. Si la tienda da soporte a varios idiomas, la

descripción de la entidad de tienda puede hacerse en varios idiomas. La descripción puede incluir una dirección de contacto para la entidad de tienda y también una dirección de ubicación para la entidad de tienda.

Tienda

Una *tienda* es una entidad de tienda. Toda tienda debe pertenecer a un grupo de tiendas.

Grupo de tiendas

Un *grupo de tiendas* es un conjunto de tiendas. Un grupo de tiendas es una entidad de tienda. El grupo de tiendas actúa como contenedor de la información, que se puede almacenar a nivel de grupos de tienda y que pueden compartir todas las tiendas del grupo de tiendas. Por ejemplo, las tiendas del mismo grupo de tiendas pueden compartir información como categorías de impuestos, idiomas soportados, monedas soportadas, códigos de cálculo y jurisdicciones de envío.

Actualmente, sólo puede haber y mantenerse un grupo de tiendas a nivel de administración de sitio dentro de WebSphere Commerce Server.



Para obtener información más detallada sobre la estructura de los elementos de tienda en WebSphere Commerce Server, consulte los modelos de datos y de objeto de tienda en la ayuda en línea de WebSphere Commerce.

Creación de elementos de tienda en WebSphere Commerce

Las herramientas de tienda de WebSphere Commerce Accelerator le permiten crear o editar los siguientes elementos de tienda:

- el identificador de tienda y el identificador de miembro en los elementos de contacto
- el identificador de tienda en la tabla STOREENT
- el directorio de tienda en la tabla STORE
- el apodo de la dirección en la tabla STADDRESS
- la descripción de tienda
- la dirección de la tienda

Por esto, tiene dos opciones para crear elementos de tienda:

- Editar los elementos de tienda ya existentes de una de las tiendas de ejemplo que se proporcionan con WebSphere Commerce.
- Crear elementos de tienda con el formato de un archivo XML que se puede publicar como parte de un archivador de tienda o que se puede cargar utilizando Loader package.

Para obtener información sobre la creación de elementos de tienda en forma de archivo XML, consulte el apartado “Creación de elementos de datos de tienda en un archivo XML”. Para obtener más información sobre cómo crear descuentos utilizando WebSphere Commerce Accelerator, consulte la Ayuda en línea a la producción de WebSphere Commerce.

Creación de elementos de datos de tienda en un archivo XML

Cree los elementos de tienda en formato de archivos XML que se puedan cargar en la base de datos utilizando Loader package. Si crea una tienda globalizada, deberá crear archivos XML para cada entorno nacional al que la tienda dé soporte. El archivo específico de entorno nacional debe especificar toda la información de las

descripciones, para que se pueda traducir fácilmente. Para obtener más información sobre el paquete Loader, consulte la Parte 10, "Publicación de la tienda", en la página 345.

Las tiendas de ejemplo, de las cuales se toman muchos de los ejemplos de estas tareas, utilizan un archivo `store.xml` para toda la información que no es necesario traducir y otro archivo `store.xml` para cada entorno nacional que la tienda soporta, con la información que debe traducirse. Los archivos específicos del entorno nacional contienen toda la información de las descripciones.

Para crear elementos de tienda, haga lo siguiente:

1. Revise la información del Capítulo 38, "Carga de grupos de elementos de base de datos de WebSphere Commerce", en la página 405.
2. Revise los archivos XML que se utilizan para crear elementos de tienda para las tiendas de ejemplo. Todos los archivos para las tiendas de ejemplo se encuentran en el archivo archivador de tienda correspondiente.

Los archivos de archivador de tienda se encuentran en el directorio siguiente:

- `dir_instalación_WC/samplestores`

Nota: La publicación *WebSphere Commerce, Guía de tiendas de ejemplo* contiene información sobre cada uno de los elementos de datos contenidos en las tiendas de ejemplo.

Cada tienda de ejemplo incluye varios archivos `store.xml`, que incluyen la información de la tienda por idioma. Dado que las tiendas de ejemplo se traducen a varios idiomas, habrá varios archivos `store.xml` en cada tienda. Para ver los archivos `store.xml` en el archivador de tienda, descomprímalo utilizando un programa ZIP. Los archivos `store.xml` se encuentran en el directorio de datos. El archivo `store.xml` específico de un idioma está situado en el subdirectorio específico del entorno nacional del directorio de datos.

3. Revise la información del Apéndice B, "Creación de los datos", en la página 465.
4. Cree un archivo `store.xml`, copiando uno de los archivos `store.xml` de los archivadores de tienda de ejemplo o creando uno nuevo. Para obtener más información, consulte el archivo `wcs.dtd`. El archivo DTD se encuentra en el siguiente directorio:
 - `dir_instalación_WC/schema/xml`
5. Cree una entidad de tienda.

- a. Utilizando el ejemplo siguiente como muestra, defina una entidad de tienda en su archivo XML para la tabla STOREENT.

```
<storeent
  storeent_id="@storeent_id_1"
  member_id="%MEMBER_ID"
  type="S"
  identifier="ToolTech"
  setccurr="EUR"
/>
```

donde

- `storeent_id` es una clave exclusiva generada.
- `member_id` es el propietario de la entidad de tienda.
- `type` es el tipo de entidad de tienda: G = GrupoTiendas, S = Tienda.
- `identifier` es una serie que, junto con el propietario, identifica de forma exclusiva la entidad de tienda.

- setcurr es la moneda por omisión de una entidad de tienda; es decir, la moneda que utilizará un cliente que todavía no tiene una moneda preferida. Si es NULL para una tienda, la moneda por omisión se obtiene de su grupo de tiendas.
6. Cree una dirección de tienda.
 - a. Utilizando el ejemplo siguiente como muestra, cree la dirección o direcciones de tienda en el archivo XML para la tabla STADDRESS. Si está creando una tienda globalizada, deberá incluir esta información en un archivo XML específico del entorno nacional.

```
<staddress
  staddress_id="@staddress_id_es_ES_1"
  member_id="%MEMBER_ID"
  nickname="storeaddress_Spanish"
  address1="Avenida Meridiana 12xx"
  address2="Suite 9xx"
  businesstitle="ToolTech"
  city="Hospitalet"
  state="Barcelona"
  zipcode="0xx1x"
  country="España"
  phone1="34-900-555-1234"
  fax1="34-900-555-4321"
  email1="info@tooltech.xxx"
/>
```

donde

- staddress_id es una clave exclusiva generada.
 - member_id es el propietario de la entidad de tienda.
7. Cree una descripción de la entidad de tienda.
 - a. Utilizando el ejemplo siguiente como muestra, cree la descripción de la entidad de tienda en el archivo XML para la tabla STOREENTDS. Si está creando una tienda globalizada, deberá incluir esta información en un archivo XML específico del entorno nacional.

```
<storeentds
  description="Entidad de tienda de modelos de comercio"
  language_id="%es_ES"
  displayname="ToolTech"
  storeent_id="@storeent_id_1"
  staddress_id_cont="@staddress_id_es_ES_1"
  staddress_id_loc="@staddress_id_es_ES_1"
/>
```

donde

- description es una descripción más larga de la entidad de tienda, adecuada para mostrarla a los clientes.
 - language_id es el idioma por omisión para la información que se muestra a los clientes que compran en la tienda.
 - displayname es una descripción breve de la entidad de tienda, adecuada para mostrarla a los clientes.
 - storeent_id es la entidad de tienda.
 - staddress_id_cont es la dirección de contacto de la entidad de tienda.
 - staddress_id_loc es la ubicación física de la entidad de tienda.
8. Cree una tienda en la base de datos.
 - a. Utilizando el ejemplo siguiente como muestra, defina una tienda en su archivo XML para la tabla STORE.

```
<store
  store_id="@storeent_id_1"
  directory="ToolTech"
  ffmcenter_id="@ffmcenter_id_1"
/>
```



```

language_id="&es_ES"
storegrp_id="-1"
allocationgoodfor="43200"
bopmpadfactor="0"
defaultboffset="2592000"
ffmselectionflags="0"
maxboffset="7776000"
rejectedordexpiry="259200"
rtnffmctr_id="@ffmcenter_id_1"
pricerefflags="0"
storetype="B2B"
/>

```

donde

- store_id es una clave exclusiva generada.
- directory es el directorio en el que se encuentran los elementos Web específicos de la tienda. El directorio se encuentra bajo el directorio raíz de documentos del módulo Web Store.war.
- ffmcenter_id es el centro de despacho de pedidos por omisión para la tienda.
- language_id es el idioma por omisión para la información que se muestra a los clientes que compran en la tienda.
- storegrp_id es el grupo de tiendas al que está asociada la tienda. Este número se genera en la tabla STOREGRP.
- allocationgoodfor significa que el planificador ReleaseExpiredAllocations puede utilizarse para anular asignaciones de inventario ATP cuando haya transcurrido el número de segundos especificado desde que se efectuaron las asignaciones.
- bopmpadfactor indica si esta tienda calcula los importes de pedido (como impuestos o costes de envío) de forma diferente para los distintos centros de despacho de pedidos; el importe de un pedido sometido previamente puede cambiar cuando se asignen finalmente centros de despacho a los artículos de pedido en espera. Este factor de relleno representa el porcentaje que puede aumentar el importe de pedido presentado a Payment Manager, si fuera necesario. Por ejemplo, especifique 5 para permitir un incremento de hasta el 5 por ciento.
- defaultboffset se utiliza cuando no se puede determinar una fecha estimada de disponibilidad para un artículo de pedido en espera; en ese caso, en el futuro se establecerá en este número de segundos.
- maxboffset significa que, si el tiempo de disponibilidad estimado para un artículo de pedido en espera va a exceder normalmente este número de segundos en el futuro, se establecerá en este número de segundos en el futuro.
- rejectedordexpiry son los pedidos con el pago en estado Denegado por un tiempo superior a este número de segundos y son candidatos para ser cancelados.
- rtnffmctr_id es el centro de despacho de pedidos por omisión para devolver mercancía a la tienda.
- pricerefflags contiene distintivos de bit que controlan en qué Acuerdos comerciales (TradingAgreements) y Ofertas (Offers) se buscará cuando la implementación por omisión del mandato de tarea GetContractUnitPrices renueve los precios:
 - 1 = usePreviousOnly - Utilizar aquellos a los que hacen referencia los artículos de pedido. Da error si ya no se pueden utilizar.

- 2 = usePreviousOrSearchAgain - Igual que usePreviousOnly, pero en lugar de dar error cuando ya no se pueden utilizar, buscar en los que están guardados en las tablas ORDIOFFER y ORDITRD
- 4 = alwaysSearchAgain - Buscar siempre en los que están guardados en las tablas ORDIOFFER y ORDITRD.
- storetype indica uno de los siguientes tipos de tienda para una interfaz de usuario que proporciona las funciones adecuadas según el valor de StoreType:
 - B2B = Directo a B2B
 - B2C = Empresa a consumidor (directo al consumidor)
 - CHS = Centro de revendedores (Commerce Plaza)
 - CPS = Tienda con perfil de catálogo maestro (tienda con elementos de catálogo)
 - RHS = Tienda alojada de revendedor
 - RPS = Tienda con perfil de revendedor (tienda con elementos de escaparate de revendedor)
 - DPS = Tienda con perfil de distribuidor (tienda con elementos de distribuidor)
 - DPX = Tienda proxy de distribuidor
 - HCP = Centro de alojamiento de Commerce (Centro de alojamiento)
 - PBS = Directorio de tiendas
 - MPS = Tienda con perfil de revendedor (tienda con elementos de escaparate de alojamiento)
 - MHS = Tienda alojada de comerciante
 - SCP = Centro de proveedores
 - SPS = Tienda con perfil de proveedor (tienda con elementos de proveedor)
 - SHS = Tienda alojada de proveedor

Nota: Los nombres que aparecen entre corchetes son los nombres de los ejemplos correspondientes que se proporcionan con WebSphere Commerce.

9. Defina un idioma soportado para la tienda.
 - a. Utilizando el ejemplo siguiente como guía, defina un idioma soportado para la tienda en su archivo XML para añadir información a la tabla STORELANG. Si la tienda da soporte a varios idiomas, debe incluir esta información en un archivo XML específico del entorno nacional (uno para cada idioma soportado por la tienda).


```
<storelang
  language_id="&es_ES"
  storeent_id="@storeent_id_1"
/>
```

 donde
 - language_id es el idioma soportado por la entidad de tienda.
 - storeent_id es la entidad de tienda.
 - b. Utilizando el ejemplo siguiente como muestra, añada información sobre el idioma a la tabla STORELANGDS. Si la tienda da soporte a varios idiomas, debe incluir esta información en un archivo XML específico del entorno nacional (uno para cada idioma soportado por la tienda).

```
<storlangds
  description="Español"
  language_id="&es_ES"
  storeent_id="@storeent_id_1"
  language_id_desc="& es_ES"
/>
```

donde

- description es una descripción breve del idioma, adecuada para mostrarla a los clientes en una lista de selección.
- language_id es el idioma de la descripción.
- storeent_id es la entidad de tienda que da soporte al idioma.
- language_id_desc es el idioma que se describe.



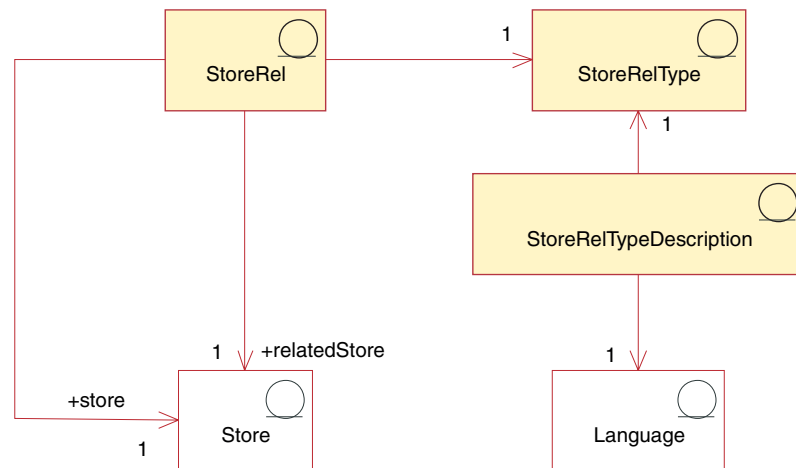
Para obtener más información sobre cómo utilizar @ y & consulte el Apéndice B, "Creación de los datos", en la página 465.

Capítulo 14. Relaciones entre tiendas

WebSphere Commerce da soporte a varios tipos de relaciones entre las tiendas de un sitio. Por ejemplo, una tienda puede proporcionar servicios de alojamiento a otra tienda o una tienda puede utilizar los elementos de catálogo o moneda que proporciona otra tienda.

Descripción de las relaciones entre tiendas en WebSphere Commerce

El diagrama siguiente ilustra las relaciones entre tiendas en WebSphere Commerce Server.



Relaciones entre tiendas

Una *relación entre tiendas* (capturada en la tabla StoreRel) es la relación entre dos tiendas. Todas las relaciones entre tiendas son direccionales, es decir, en cada relación entre tiendas una tienda proporciona los servicios y la segunda tienda de la relación utiliza estos servicios. Por ejemplo, la tienda A utiliza los catálogos que proporciona la tienda B.

Cada relación entre tiendas tiene un tipo de relación entre tiendas (StoreRelType).

Tipos de relaciones entre tiendas

Un *tipo de relación entre tiendas* (StoreRelType) define el tipo de relación que existe entre dos tiendas. Cada tipo de relación entre tiendas define su propia relación, es decir, los roles que representará cada parte en la relación y la relación existente entre ambas.

Tipos de relaciones entre tiendas a los que da soporte WebSphere Commerce

WebSphere Commerce da soporte a varias relaciones entre tiendas. Los tipos de relaciones por omisión que proporciona WebSphere Commerce se pueden agrupar aproximadamente en dos categorías:

- Relaciones en las que una tienda proporciona elementos de datos a otra tienda. Por ejemplo, la tienda A proporciona datos de catálogo que se utilizan en la tienda B.

- Relaciones en las que una tienda tiene una "relación de negocio" con otra tienda, es decir, una tienda puede alojar otra tienda o una tienda puede transferir un carro de la compra a otra tienda.

Relaciones en las que una tienda proporciona los elementos de datos a otra tienda:

Tabla 8.

Tipo de relación	Descripción	Para obtener más información, consulte
com.ibm.commerce.businessPolicy	Una tienda utiliza las políticas de negocio definidas en otra tienda.	Capítulo 18, "Elementos de contrato", en la página 195
com.ibm.commerce.campaigns	Una tienda utiliza campañas definidas en otra tienda.	Capítulo 20, "Elementos de campaña", en la página 223
com.ibm.commerce.catalog	Una tienda utiliza los datos de catálogo definidos en otra tienda.	Capítulo 16, "Elementos de catálogo", en la página 155
com.ibm.commerce.command	Una tienda utiliza los mandatos definidos en otra tienda.	Capítulo 15, "Datos de los registros de URL, de mandatos y de vistas", en la página 149
com.ibm.commerce.price	Una tienda utiliza los datos de precio definidos en otra tienda.	Capítulo 17, "Elementos de fijación de precios", en la página 187
com.ibm.commerce.segmentation	Una tienda utiliza los datos de perfil de cliente definidos en otra tienda.	Capítulo 32, "Perfiles de cliente", en la página 307
com.ibm.commerce.URL	Una tienda utiliza los URL definidos en otra tienda.	Capítulo 15, "Datos de los registros de URL, de mandatos y de vistas", en la página 149
com.ibm.commerce.view	Una tienda utiliza las vistas definidas en otra tienda.	Capítulo 15, "Datos de los registros de URL, de mandatos y de vistas", en la página 149
com.ibm.commerce.storeitem	Una tienda utiliza los elementos definidos en otra tienda.	Capítulo 29, "Elementos de inventario", en la página 289
com.ibm.commerce.propertyFiles	Una tienda utiliza los archivos de propiedades definidos en otra tienda.	
com.ibm.commerce.currency.conversion	Una tienda utiliza los índices de conversión de moneda definidos en otra tienda.	Capítulo 23, "Elementos de moneda", en la página 235
com.ibm.commerce.currency.supported	Una tienda utiliza las monedas soportadas en otra tienda.	Capítulo 23, "Elementos de moneda", en la página 235
com.ibm.commerce.currency.format	Una tienda utiliza los formatos de moneda definidos en otra tienda.	Capítulo 23, "Elementos de moneda", en la página 235

Tabla 8. (continuación)

com.ibm.commerce.currency.countervalue	Una tienda utiliza los contravalores de moneda definidos en otra tienda.	Capítulo 23, "Elementos de moneda", en la página 235
com.ibm.commerce.measurement.format	Una tienda utiliza las unidades de medida definidas en otra tienda.	Capítulo 24, "Elementos de unidad de medida", en la página 243

Una tienda puede tener relaciones con varias tiendas. Es decir, es posible que la tienda A desee utilizar los recursos de catálogo de las tiendas B, C y D. Para facilitar este tipo de relaciones entre varias tiendas, debe proporcionar un orden de secuencia para las tiendas desde las que una tienda va a utilizar los elementos. El orden de secuencia de las relaciones entre tiendas funciona de los siguientes modos:

- Alteración temporal: Si la relación entre tiendas sigue el método de alteración temporal del orden de secuencia, se utilizará la relación entre tiendas que tenga el número de secuencia más bajo. Las siguientes relaciones entre tiendas utilizan el método de alteración temporal:
 - mandato
 - moneda
 - medida
 - precio
 - archivos de propiedades
 - elementos de tienda
 - URL
 - vistas
- Fusión: Si la relación entre tiendas sigue el orden de secuencia del método de fusión, WebSphere Commerce busca todas las relaciones entre tiendas asociadas a dicha tienda y fusiona los datos de todas las tiendas asociadas. Las siguientes relaciones entre tiendas utilizan el método de fusión:
 - políticas de negocio
 - campañas
 - catálogo
 - segmentación

Todos los tipos de relaciones entre tiendas se designan utilizando el método de orden de secuencia de alteración temporal o fusión.

Nota: Aunque no existe un tipo de relación entre tiendas para contratos, se puede desplegar un sólo contrato para varias tiendas. Para obtener más información, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

Relaciones en las que una tienda tiene una "relación de negocio" con otra tienda:

Tabla 9.

Tipo de relación	Descripción
com.ibm.commerce.hostedStore	La tienda de Centro aloja las tiendas de revendedor, de proveedor o alojadas.

Tabla 9. (continuación)

com.ibm.commerce.referral	La tienda de Centro tiene una relación de referencia con los distribuidores. La tienda de Centro puede transferir un carro de la compra a una tienda de distribuidor. Generalmente, la tienda que recibe el carro de la compra es una tienda proxy para un sistema externo.
com.ibm.commerce.channelStore	Una tienda actúa como tienda de Centro para otra tienda. Esta relación define la relación entre el directorio de tiendas y el Centro de alojamiento.

Descripción del tipo de relación entre tiendas

Una *descripción del tipo de relación entre tiendas* describe el tipo de relación. Cada tipo de relación entre tiendas describe solamente un tipo de relación. La descripción del tipo de relación entre tiendas puede estar disponible en más de un idioma.

Creación de relaciones entre tiendas en WebSphere Commerce

Cree las relaciones de tienda en formato de archivos XML que se puedan cargar en la base de datos utilizando Loader package. Para obtener más información sobre Loader package, consulte la Parte 10, "Publicación de la tienda", en la página 345.

Nota: Si utiliza el Asistente de creación de tiendas para crear tiendas alojas (para obtener más información, consulte el apartado "Asistente de creación de tiendas" en la página 72) o acuerdos de servicio para crear tiendas de proxy de distribuidores (para obtener más información, consulte el apartado "Creación de tiendas proxy" en la página 73) muchas de estas relaciones entre tiendas se crean automáticamente.

Para crear elementos de relaciones entre tiendas, haga lo siguiente:

1. Revise la información del Capítulo 38, "Carga de grupos de elementos de base de datos de WebSphere Commerce", en la página 405.
2. Revise los archivos XML que se utilizan para crear elementos de tienda para las tiendas de ejemplo. Todos los archivos para las tiendas de ejemplo se encuentran en el archivo archivador de tienda correspondiente.

Los archivos de archivador de tienda se encuentran en el directorio siguiente:

- *dir_instalación_WC/samplestores*

Nota: La publicación *WebSphere Commerce, Guía de tiendas de ejemplo* contiene información sobre cada uno de los elementos de datos contenidos en las tiendas de ejemplo.

3. Revise la información del Apéndice B, "Creación de los datos", en la página 465.
4. Cree un archivo *storrelation.xml*, copiando uno de los archivos *storrelation.xml* de los archivadores de tienda de ejemplo o creando uno nuevo. Para obtener más información, consulte el archivo *wcs.dtd*. El archivo DTD se encuentra en el siguiente directorio:
 - *dir_instalación_WC/schema/xml*
5. Cree una relación entre tiendas.

- a. Utilizando el ejemplo siguiente como muestra, defina una entidad de tienda en su archivo XML para la tabla STOREREL.

```
<storere1  
store_id="@storeent_id_1"  
  relatedstore_id="@storeent_id_2"  
  streltype="-4"  
sequence="0"  
state="1"  
>
```

donde

- store_id es la tienda principal que utiliza los servicios de la tienda relacionada. .
- relatedstore_id es la tienda que proporciona el servicio que utiliza la tienda principal.
- streltype es el tipo de relación. Los tipos de relaciones por omisión son los siguientes:
 - -1 com.ibm.commerce.businessPolicy
 - -3 com.ibm.commerce.campaigns
 - -4 com.ibm.commerce.catalog
 - -5 com.ibm.commerce.command
 - -6 com.ibm.commerce.hostedStore
 - -7 com.ibm.commerce.price
 - -8 com.ibm.commerce.referral
 - -9 com.ibm.commerce.segmentation
 - -10 com.ibm.commerce.URL
 - -11 com.ibm.commerce.view
 - -13 com.ibm.commerce.inventory
 - -14 com.ibm.commerce.storeitem
 - -15 com.ibm.commerce.channelStore
 - -16 com.ibm.commerce.propertyFiles
 - -17 com.ibm.commerce.currency.conversion
 - -18 com.ibm.commerce.currency.format
 - -19 com.ibm.commerce.currency.supported
 - -20 com.ibm.commerce.currency.countervalue
 - -21 com.ibm.commerce.measurement.format
- sequence define el orden de la secuencia de selección cuando se ha definido más de una tienda relacionada para el mismo tipo de relación. El valor por omisión es 0.
- state es el estado de la relación (0 = inactivo, 1 = activo). El valor por omisión es 1.



Para obtener más información sobre cómo utilizar @ y & consulte el Apéndice B, "Creación de los datos", en la página 465.

Capítulo 15. Datos de los registros de URL, de mandatos y de vistas

Los registros de URL, de mandatos y de vistas forman parte de la infraestructura de mandatos de WebSphere Commerce que se describe detalladamente en la publicación *WebSphere Commerce, Guías de programación y aprendizaje*, capítulo uno "Visión General", capítulo dos "Patrones de diseño" y capítulo seis "Implementación de mandatos". En este apartado se proporciona una breve visión general para poder comprender cómo los registros de URL, de mandatos y de vistas se ajustan al modelo de información.

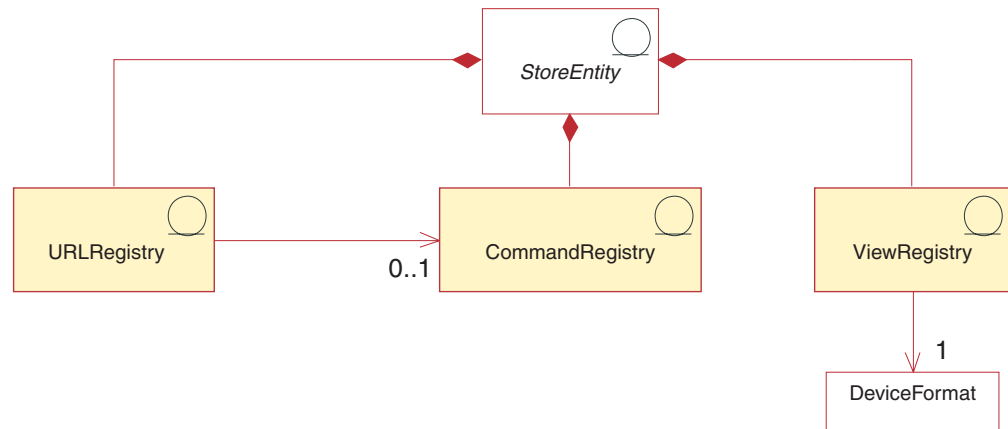


Para obtener información más detallada sobre la estructura de elementos de mandato y de vista en WebSphere Commerce Server, consulte los modelos de datos de mandatos y vistas en la ayuda en línea de WebSphere Commerce.

Descripción de los registros de URL, de mandatos y de vistas en WebSphere Commerce

La infraestructura de mandatos de WebSphere Commerce determina el modo en que se ejecuta un mandato y, a continuación, devuelve una respuesta según la vista que haya devuelto el mandato ejecutado. La ejecución del mandato y la respuesta depende de cada tienda, lo que significa que el mismo mandato se puede implementar de forma diferente para cada tienda y también puede devolver respuestas diferentes para cada tienda.

El diagrama siguiente ilustra la estructura de registros de URL, de mandatos y de vistas en WebSphere Commerce Server.



Registro de URL

El registro de URL correlaciona un nombre de mandato con la interfaz real del mandato que se va a ejecutar. Cada entrada del registro de URL es sensible a la tienda, esto es, cada tienda puede definir una interfaz diferente para el mismo valor de URL. Si no se puede encontrar la versión de la tienda del registro de URL, entonces se utilizará el registro de URL definido para el sitio (tienda 0). Por omisión, todos los registros de URL se definen para el sitio.

Otras tiendas pueden utilizar los URL definidos y registrados en una tienda. Para que una tienda pueda utilizar los URL definidos en otra tienda, debe crearse una relación entre tiendas de tipo `com.ibm.commerce.URL`. Para obtener más información, consulte el Capítulo 14, "Relaciones entre tiendas", en la página 143.

Registro de mandatos

Todos los mandatos, tanto si son mandatos de controlador como mandatos de tarea, se pueden definir en el registro de mandatos. Si se define un mandato en el registro de mandatos, dicha definición se utilizará como la implementación del mandato cuando se ejecute el mandato. Si no se define un mandato en el registro de mandatos, dicha definición se utilizará una implementación por omisión. Se asigna una implementación por omisión a cada interfaz de mandatos, de modo que si el mandato no está definido en el registro de mandatos se utiliza la implementación por omisión.

Si se define un mandato en el registro de mandatos como un mandato de nivel de sitio (tienda 0), se utiliza la implementación de nivel de sitio, excepto cuando se ejecuta el mandato para una tienda que tiene definida una implementación diferente del mandato.

El registro de mandatos permite que tiendas diferentes utilicen los mismos mandatos para ampliar una parte o todas las implementaciones sin tener que cambiar el flujo original del mandato.

Otras tiendas pueden utilizar los mandatos definidos y registrados en una tienda. Para que una tienda pueda utilizar los mandatos definidos en otra tienda debe crearse una relación entre tiendas de tipo `com.ibm.commerce.command`. Para obtener más información, consulte el Capítulo 14, "Relaciones entre tiendas", en la página 143.

Registro de vistas

Una vez ejecutado un mandato, en la mayor parte de los casos, el solicitante del mandato requiere que se le devuelva una respuesta. Cuando se determina la respuesta, la infraestructura de mandatos tiene en cuenta los factores siguientes:

- La vista encontrada en las propiedades de la respuesta una vez ejecutado el mandato.
- La tienda en cuyo nombre se ha ejecutado el mandato.
- El formato de dispositivo de la petición cuando se ésta se ha realizado.

Cada vista que devuelva una respuesta se debe definir en el registro de vistas, ya sea por tienda o, por omisión, por sitio. Normalmente, toda tienda definirá la vista para cada formato de dispositivo posible de la petición de entrada. No obstante, si una tienda no ha definido una vista, se utilizará la vista por omisión del sitio. El adaptador que maneja la petición decidirá qué formato de dispositivo y qué formato de dispositivo por omisión se ha de utilizar cuando determine la vista a la que se ha de llamar. No hay un formato de dispositivo genérico, por lo tanto, dependiendo de los diferentes tipos de peticiones que WebSphere Commerce puede aceptar, puede haber una vista definida para cada formato de dispositivo.

Otras tiendas pueden utilizar las vistas definidas y registradas en una tienda. Para que una tienda pueda utilizar las vistas definidas en otra tienda debe crearse una relación entre tiendas de tipo `com.ibm.commerce.view`. Para obtener más información, consulte el Capítulo 14, "Relaciones entre tiendas", en la página 143.

Creación de nuevos mandatos, vistas y URL

Cuando crea una instancia de WebSphere Commerce Server, los mandatos, las vistas y los URL que se proporcionan con WebSphere Commerce se registran en la base de datos de WebSphere Commerce Server, en las tablas correspondientes: CMDREG, VIEWREG y URLREG. Estos mandatos, vistas y URL están disponibles para que los utilicen todas las tiendas que residen en la instancia.

WebSphere Commerce también proporciona archivos JSP por omisión para mostrar las vistas por omisión. Estos archivos JSP están asociados a las vistas en la tabla VIEWREG.

Si crea mandatos, vistas o URL nuevos, o personaliza alguno ya existente, debe registrarlos en las tablas de base de datos correspondientes (CMDREG, VIEWREG y URLREG) para que puedan utilizarse en la tienda. Si crea nuevos archivos JSP para utilizarlos en la tienda, debe asociarlos a la vista correspondiente en la tabla VIEWREG.

Nota: Si crea un nuevo archivo JSP, pero le da el mismo nombre que el archivo JSP por omisión asociado a la vista, no es necesario registrar el nuevo archivo JSP en la tabla VIEWREG.

Nota: Cuando cree vistas nuevas, asegúrese de que asocia las políticas de control de acceso a cada vista nueva. Para obtener más información, consulte “Adición de control de acceso a la tienda” en la página 315.

Para obtener más información sobre cómo crear o personalizar mandatos, vistas o URL, consulte la publicación *WebSphere Commerce, Guías de programación y aprendizaje*. La publicación *WebSphere Commerce, Guías de programación y aprendizaje* contiene también información sobre cómo y cuándo se han de registrar mandatos, vistas, URL y archivos JSP.

Cómo registrar mandatos, vistas y URL en WebSphere Commerce

Si crea o personaliza varios mandatos, vistas, URL o archivos JSP nuevos para la tienda, quizá desee registrarlos utilizando un archivo XML, que puede cargar posteriormente en la base de datos utilizando Loader package, o como parte de un archivador de tienda que se puede publicar utilizando el programa de utilidad de publicación en la Consola de administración. Para obtener más información sobre Loader package, consulte la Parte 10, “Publicación de la tienda”, en la página 345.

Nota: Antes de crear un archivo XML para cargar mandatos nuevos o personalizados, consulte la publicación *WebSphere Commerce, Guías de programación y aprendizaje* para obtener más detalles sobre cómo funcionan los mandatos.

Creación de un archivo XML para registrar mandatos, vistas y URL

Para crear un archivo XML para registrar los mandatos, vistas y archivos JSP nuevos para la tienda, haga lo siguiente:

1. Revise la información del Capítulo 38, “Carga de grupos de elementos de base de datos de WebSphere Commerce”, en la página 405.
2. Revise los archivos XML que se utilizan para registrar mandatos, vistas y archivos JSP para las tiendas de ejemplo. Cada tienda de ejemplo incluye un

archivo `command.xml`, que contiene la información de registro. Los archivos de archivador de tienda se encuentran en el directorio siguiente:

- `dir_instalación_WC/samplestores`

Nota: La publicación *WebSphere Commerce, Guía de tiendas de ejemplo* contiene información sobre cada uno de los elementos de datos contenidos en las tiendas de ejemplo.

Para ver el contenido del archivador de tienda, utilice un programa de descompresión. El archivo `command.xml` está ubicado en el directorio `data`.

3. Revise la información del Apéndice B, "Creación de los datos", en la página 465.
4. Cree un archivo `command.xml`, copiando uno de los archivos `command.xml` de los archivadores de tienda de ejemplo o creando uno nuevo. Para obtener más información, consulte el archivo `wcs.dtd`. Los archivos DTD se encuentran en el directorio siguiente:

- `dir_instalación_WC/schema/xml`

5. Los mandatos de controlador deben registrarse en las tablas `URLREG` y `CMDREG`. Para registrar un mandato de controlador nuevo o personalizado en la tabla `URLREG`, cree una entrada en el archivo XML para cada mandato de controlador nuevo o personalizado, utilizando el ejemplo siguiente como muestra:

```
<urlreg
url="MiVisualizacióndeproductos"
storeent_id="@storeent_id_1"
interfacename="com.mitienda.commerce.catalog.commands.ProductDisplayCmd"
https="0"
description="Mandato de visualización de productos para mi tienda"
authenticated="0"
internal="0" />
```

donde

- `urlreg` es el nombre de la tabla de base de datos (`URLREG`) en la que se va a insertar esta información.
- `url` es el nombre de URI
- `storeent_id` es el identificador de entidad de tienda; el uso del símbolo `@` se denomina resolución de alias interno. Al utilizar la resolución de alias interno, la clave primaria (identificador) se sustituye por un alias en el documento XML. Este alias se utiliza luego en cualquier otra parte del archivo XML para hacer referencia a ese elemento. De este modo, no es necesario saber los índices exclusivos necesarios para poder crear el archivo XML. Durante la publicación, ID Resolver sustituye el símbolo `@` por un valor exclusivo. Para obtener más información, consulte el Apéndice B, "Creación de los datos", en la página 465.
- `interfacename` es el nombre de la interfaz de mandatos de controlador
- `https` es el HTTP seguro necesario para esta petición de URL. Utilice 1 cuando sea necesario HTTP seguro y 0 cuando no lo sea.
- `authenticated` indica si es necesario que el usuario esté conectado para esta petición de URL. Utilice 1 cuando sea necesaria la autenticación y 0 cuando no lo sea.

- `internal` indica si el mandato es interno de WebSphere Commerce. Los URL internos los utilizan las herramientas de WebSphere Commerce. Utilice 1 cuando sea interno y 0 cuando sea externo. Los URL que cree deberán ser externos.
6. Para registrar un nuevo mandato de controlador o de tarea en la tabla `CMDREG`, cree una entrada en el archivo XML para cada mandato de controlador o de tarea nuevo o personalizado, utilizando el ejemplo siguiente de un mandato de tarea como muestra (del archivo `command.xml` de la tienda de ejemplo `ToolTech`):

```
< cmdreg
storeent_id="@storeent_id_1"
interfacename="com.ibm.commerce.payment.commands.DoPaymentCmd"
classname="com.ibm.commerce.payment.commands.DoPaymentMPFCmdImpl"/>
donde
```

- `cmdreg` es el nombre de la tabla de base de datos (`CMDREG`) en la que se va a insertar esta información.
 - `storeent_id` es el identificador de entidad de tienda y el uso del símbolo `@` se denomina resolución de alias interno. Al utilizar la resolución de alias interno, la clave primaria (identificador) se sustituye por un alias en el documento XML. Este alias se utiliza luego en cualquier otra parte del archivo XML para hacer referencia a ese elemento. De este modo, no es necesario saber los índices exclusivos necesarios para poder crear el archivo XML. Durante la publicación, ID Resolver sustituye el símbolo `@` por un valor exclusivo. Para obtener más información, consulte el Apéndice B, "Creación de los datos", en la página 465.
 - `interfacename` es el nombre de la interfaz de mandatos
 - `classname` es el nombre de la clase de implementación del mandato. Normalmente, es el nombre de la interfaz añadiendo `Impl` al final.
7. Para registrar nuevas vistas, o para asociar nuevos archivos JSP a una vista, cree una entrada en la tabla `VIEWREG`, utilizando el ejemplo siguiente como muestra (del archivo `command.xml` de la tienda de ejemplo `ToolTech`):

```
<viewreg
viewname="OrderOptionsView"
devicefmt_id="-1"
storeent_id="@storeent_id_1"
interfacename="com.ibm.commerce.command.ForwardViewCommand"
classname="com.ibm.commerce.command.HttpForwardViewCommandImpl"
properties="docname=Shipping.jsp"
internal="0"
https="0"/>
donde
```

- `viewreg` es el nombre de la tabla de base de datos (`VIEWREG`) en la que se va a insertar esta información.
- `viewname` es el nombre de la vista.
- `devicefmt_id` es el tipo de dispositivo en el que se utilizará esta vista, por ejemplo, un navegador.
- `storeent_id` es el identificador de entidad de tienda; el uso del símbolo `@` se denomina resolución de alias interno. Al utilizar la resolución de alias interno, la clave primaria (identificador) se sustituye por un alias en el documento XML. Este alias se utiliza luego en cualquier otra parte del

archivo XML para hacer referencia a ese elemento. De este modo, no es necesario saber los índices exclusivos necesarios para poder crear el archivo XML. Durante la publicación, ID Resolver sustituye el símbolo @ por un valor exclusivo. Para obtener más información, consulte el Apéndice B, "Creación de los datos", en la página 465.

- `interfacename` es el nombre de la interfaz de mandatos de vista. Las opciones por omisión son `ForwardView`, `DirectView` y `RedirectView`.
- `classname` es el nombre de la clase de implementación de la vista. Normalmente, es el nombre de la interfaz añadiendo `Impl` al final.
- `properties` son las parejas nombre-valor por omisión definidas como propiedades de entrada del mandato. Si siempre se visualiza la misma página, establezca el nombre de archivo JSP en esta propiedad; por ejemplo, `docname=Shipping.jsp`.
- `internal` indica si la vista es interna de WebSphere Commerce. Las vistas internas las utilizan las herramientas de WebSphere Commerce. Utilice 1 cuando sea interno y 0 cuando sea externo. Las vistas que cree deberán ser externas.
- `https` es el HTTP seguro necesario para esta petición de URL. Utilice 1 cuando sea necesario HTTP seguro y 0 cuando no lo sea.



Para obtener más información sobre cómo utilizar @ y & consulte el Apéndice B, "Creación de los datos", en la página 465.

Capítulo 16. Elementos de catálogo

Al igual que un catálogo tradicional, el catálogo en línea contiene los artículos y servicios que se ofrecen para la venta. Aunque el tamaño y la estructura de los catálogos en línea pueden variar mucho de una tienda a otra, dependiendo del tipo y la cantidad de mercancía disponible para la venta, los catálogos requieren lo siguiente:

- Los artículos y servicios que se ofrecen, incluyendo:
 - Precios, que normalmente se incluyen en el catálogo en línea
 - Datos de los productos, como por ejemplo, descripciones e imágenes de la mercancía.
 - Categorías, ya que la mayoría de catálogos, aunque no todos, dividen la mercancía en categorías para que a los clientes les resulte más fácil examinarlos.
- Un método de visualización para mostrar lo que se va a vender. Las páginas de visualización de catálogo describen esquemáticamente cómo se presenta una página a los clientes y proporcionan un aspecto visual uniforme en las diversas páginas del catálogo. La estructura del catálogo depende de la mercancía.

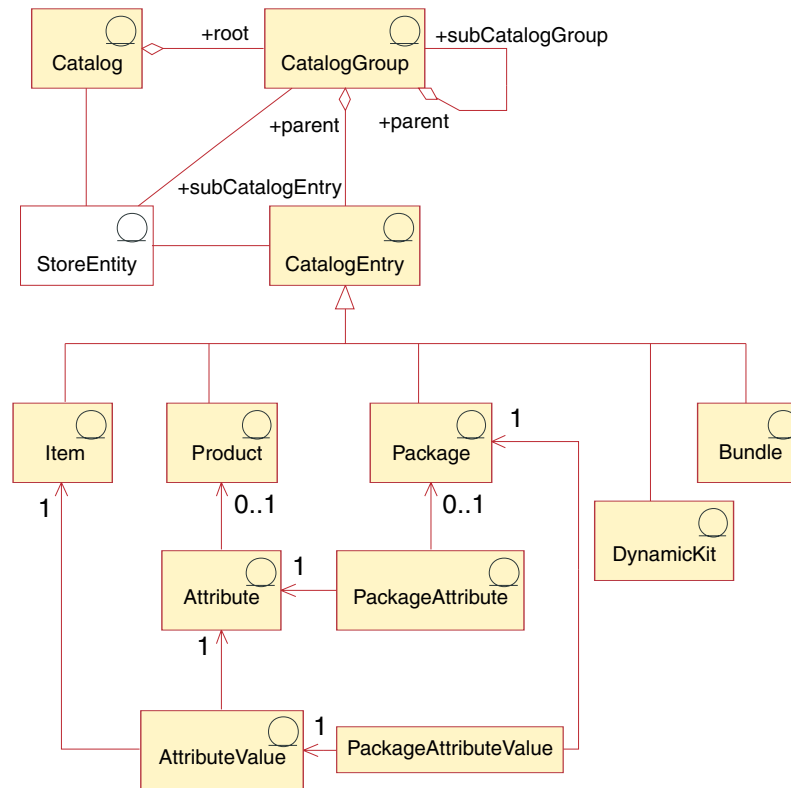
Los catálogos en WebSphere Commerce

WebSphere Commerce impone varios requisitos al catálogo en línea de la tienda. Cada tienda del sistema WebSphere Commerce debe tener un *catálogo maestro*, al que también se hace referencia simplemente como catálogo. El catálogo maestro es la ubicación central para gestionar la mercancía de la tienda. Es el catálogo individual que contiene todos los productos, artículos, relaciones y precios estándar para todo lo que está a la venta en la tienda.

Se puede compartir el catálogo maestro entre las tiendas y definir tantas tiendas como se necesiten. Además de crear un catálogo maestro para la gestión de catálogo, también puede elegir crear uno o más *catálogos de ventas* para visualizarlos. Un catálogo de ventas puede contener un subconjunto o las mismas entradas de catálogo que el catálogo maestro, pero tiene una estructura de categorías mucho más flexible para que lo visualicen los clientes. Aunque solamente hay un catálogo maestro, puede crear tantos catálogos de ventas como desee. Sin embargo, dado que el catálogo maestro es el que se utiliza para gestionar la mercancía en línea, recomendamos que también utilice el catálogo maestro como catálogo de ventas a fin de minimizar la actividad general de mantenimiento.

Si crea un nuevo catálogo maestro para una tienda de WebSphere Commerce o si está modificando un catálogo maestro ya existente de una tienda de ejemplo de WebSphere Commerce, como ToolTech, tendrá que asegurarse de que su catálogo cumple con estos requisitos. El diagrama siguiente muestra la estructura básica de

un catálogo maestro en WebSphere Commerce.



Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce. Para obtener más información sobre el modelo de información, consulte “Modelo de información de los datos de la tienda” en la página 105. Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, “Descripción de UML”, en la página 463.

Catálogo

El *catálogo* es el punto de partida del modelo de información. El catálogo contiene toda la información jerárquica y de navegación para el catálogo en línea y es un conjunto de grupos de catálogo y de entradas de catálogo que se pueden visualizar y adquirir en una tienda en línea.

En WebSphere Commerce, un catálogo se representa en la base de datos mediante una *entidad de catálogo*. Una entidad de catálogo consta de un ID de catálogo exclusivo y una descripción de catálogo, por ejemplo, el nombre de catálogo. Puesto que cada catálogo es una entidad aparte y exclusiva, puede asociarse fácilmente a una o más tiendas. Cada tienda del sistema WebSphere Commerce debe estar relacionada por lo menos con una entidad de catálogo.

Grupos de catálogo

Los *grupos de catálogo* son agrupaciones genéricas de las entradas de catálogo, que se crean por motivos de partición. Un grupo de catálogo pertenece a un catálogo y puede contener más de un grupo de catálogo o entradas de catálogo. Puede asociar grupos de catálogo con más de un catálogo. Los grupos de catálogo también se conocen como *categorías*.

Un catálogo plano es un catálogo que no agrupa sus productos en categorías sino que muestra una lista de productos. Aunque es posible crear un catálogo plano en WebSphere Commerce, es recomendable crear grupos en el catálogo por motivos de estructura y de navegación.

Para crear grupos de catálogo, en primer lugar debe ordenar el catálogo en una jerarquía o árbol invertido. El árbol empieza en los grupos de catálogo generales (denominadas categorías raíz o categorías superiores) y se ramifica en subcategorías cada vez más específicas hasta que no se puede dividir más. Los grupos de catálogo de nivel inferior, que sólo contienen productos, son categorías hoja (también denominadas final). Un grupo de catálogo es el padre de las categorías que están inmediatamente debajo de él, y un hijo del grupo de catálogo que está por encima. Por ejemplo, Moda para caballero es la agrupación de las categorías de ropa para hombre, mientras que los grupos de catálogo Pantalones y Camisas son agrupaciones de productos.

Entradas de catálogo

Cada grupo de catálogo contiene *entradas de catálogo*. Las entradas de catálogo representan mercancía que se puede solicitar en un catálogo en línea. Normalmente, las entradas tienen un nombre o número de pieza, una descripción, uno o más precios, imágenes y otros detalles. Una entrada de catálogo puede ser un producto, un artículo, un paquete, un paquete compuesto, un kit estático o un kit dinámico. Si es necesario, puede crear nuevos tipos de entrada de catálogo que no se ajusten a ninguno de los seis modelos existentes. A continuación encontrará más información sobre cada uno de estos tipos de entrada de catálogo.

Productos

Un *producto* es un tipo de entrada de catálogo. Un producto actúa como una plantilla para un grupo de artículos (o códigos de artículos) que muestran los mismos atributos. Por ejemplo, una camisa es un producto en el catálogo. Después de añadir atributos y valores de atributo a la camisa, cada variación se convierte en un artículo, como por ejemplo, una camisa negra de talla pequeña.

Artículos

Un *artículo* es una unidad tangible de mercancía que tiene un nombre, un número de pieza y un precio específicos. Por ejemplo, una camisa negra de talla pequeña es un artículo, mientras que una camisa es un producto. Todos los artículos que están relacionados con un producto determinado presentan el mismo conjunto de atributos y se distinguen por los valores de sus atributos.

Nota: Para los usuarios de WebSphere Commerce Accelerator, los términos artículos y *códigos de artículo* se consideran sinónimos. Cuando se utilizan las herramientas de Gestión de productos en WebSphere Commerce Accelerator, el artículo que puede pedirse recibe el nombre de código de artículo. En el esquema de base de datos de WebSphere Commerce, este tipo específico de entrada de catálogo se denomina artículo.

Paquetes compuestos

Un *paquete compuesto* es un conjunto de entradas de catálogo que permite a los clientes comprar varios artículos de una sola vez. Por ejemplo, un paquete compuesto para un sistema informático puede estar formado por una unidad central de proceso, un monitor, una unidad de disco duro y una unidad de CD-ROM. Un paquete compuesto es una agrupación de artículos o una combinación de productos, artículos y paquetes totalmente resueltos. Si selecciona un paquete compuesto que sólo contenga artículos, el paquete compuesto se descompone en códigos de artículo que se pueden pedir por separado y que se

añaden individualmente al carro de la compra. Sin embargo, si selecciona un paquete compuesto que contiene productos, será necesario convertir dichos productos en artículos mediante la resolución del código de artículo para poderlos añadir al carro de la compra. En cualquiera de los casos, una vez se ha descompuesto un paquete compuesto y los artículos que lo componían se han añadido al carro de la compra, los artículos pueden modificarse o eliminarse.

Paquetes

Un *paquete* es un conjunto indivisible de entradas de catálogo. Por ejemplo, un paquete de sistema informático puede contener una unidad central de proceso, un monitor y una unidad de disco duro específicos que no se pueden vender por separado. Del mismo modo que los productos, un paquete tiene atributos que lo definen y es un contenedor de paquetes totalmente resueltos. Un paquete totalmente resuelto es comparable a un código de artículo. Un paquete tiene su propio precio y es un código de artículo real que se puede pedir y que se puede añadir al carro de la compra. Un paquete no se puede dividir ni modificar durante la navegación o después de haberse colocado en el carro de la compra.

Nota: Para los usuarios de WebSphere Commerce Accelerator, paquetes y *kits premontados* se consideran sinónimos. Cuando se utilizan las herramientas de Gestión de productos en WebSphere Commerce Accelerator, un paquete recibe el nombre de kit premontado. En el esquema de base de datos de WebSphere Commerce, este tipo específico de entrada de catálogo se denomina un paquete.

Kits dinámicos

Un *kit dinámico* es un tipo de entrada de catálogo que el cliente puede configurar dinámicamente. Esta configuración (o agrupación) de productos se basa en los requisitos del cliente y se vende como una sola unidad. Los componentes de un kit dinámico los controla un configurador de productos externo mediante un conjunto de normas predefinidas y la interacción del usuario. La adición de un kit dinámico a un pedido es similar a la adición de un paquete. Al igual que sucede en un paquete, los componentes individuales de un kit dinámico no pueden modificarse y se debe llevar a cabo la configuración entera en su conjunto. Sin embargo, puede cambiar los componentes del kit dinámico volviendo a configurarlo mediante un configurador de productos externo.

Kits estáticos

Un *kit estático* es un grupo de productos que se solicitan como una unidad. La información sobre los productos contenidos en un kit estático se predefine y se controla en WebSphere Commerce. Los componentes individuales del pedido no se pueden modificar y se deben despachar juntos. Un kit estático quedará en espera si alguno de sus componentes no está disponible.

Un kit estático se crea en primer lugar como un paquete y luego el administrador de sitio lo configura.

Conjuntos de productos

Los *conjuntos de productos* están asociados a entradas de catálogo publicadas. Los conjuntos de productos proporcionan un mecanismo que permite dividir el catálogo en subconjuntos lógicos. Esta división le permite mostrar diferentes partes del catálogo a usuarios diferentes. Puede crear un contrato y especificar que los participantes del contrato sólo tienen autorización para comprar productos que estén incluidos en un conjunto de productos predefinido. WebSphere Commerce proporciona herramientas para crear y gestionar contratos y normas de filtros de autorización sobre el catálogo maestro.

Atributos

Los *atributos* son las propiedades de los productos de una tienda en línea. Existen dos tipos de atributos:

- Los atributos de definición son las propiedades como, por ejemplo el color o el tamaño. Los valores de atributo son la propiedad de un atributo, por ejemplo un color específico (azul o amarillo) o el tamaño (medio). Deberá predefinir los valores de atributo antes de asignarlos a los artículos. Los valores de atributo están implícitamente relacionados con sus atributos. Cada combinación de atributos y valores de atributos posibles es igual a un nuevo artículo. Después de crear atributos y sus valores, puede actualizar información como, por ejemplo, el nombre, la descripción y el tipo (texto, números enteros o números decimales). Para la resolución de códigos de artículos se utilizan los atributos de definición, en los que cada combinación posible de atributos y valores de atributo define un artículo.
- Por el contrario los atributos descriptivos simplemente proporcionan descripciones adicionales. Por ejemplo, algunas prendas de ropa sólo deben limpiarse en seco y nunca lavarse con agua, por lo que un atributo descriptivo puede especificar que sólo ha de lavarse en seco. Tenga en cuenta que los atributos descriptivos no se utilizan para la resolución de códigos de artículos y su función es mejorar las descripciones de los productos o permitir que la información específica de la empresa pueda personalizarse fácilmente.

Valores de atributo

Los *valores de atributo* son las propiedades de un atributo, como por ejemplo, un color específico (azul o amarillo) o el tamaño (pequeño, medio o grande). Deberá predefinir los valores de atributo antes de asignarlos a los artículos. Cada combinación posible de valores de atributos de definición define un artículo.

Atributos de paquete

Los *atributos de paquete* deben crearse a partir de los atributos de los productos que contienen los paquetes. Un paquete que sólo contiene artículos no tiene atributos de paquete.

Valores de atributo de paquete

Los *valores de atributo de paquete* son los valores asignados a los atributos de paquete. Los valores de los atributos de paquete se deben crear a partir de los valores de atributos de los productos contenidos en los paquetes.



Para obtener información más detallada sobre la estructura de los elementos de catálogo en WebSphere Commerce, consulte los modelos de datos de catálogo en la ayuda en línea de WebSphere Commerce.

Creación de elementos de catálogo en WebSphere Commerce

Para crear los elementos de catálogo para su tienda, debe crear un catálogo maestro añadiendo información a varias tablas de base de datos de WebSphere Commerce. Puede crear su catálogo utilizando archivos XML que se cargan en la base de datos utilizando Loader Package. Si crea un catálogo globalizado, necesitará archivos XML para cada entorno nacional al que la tienda dé soporte. Cada archivo XML específico de entorno nacional añade al catálogo, grupos de catálogo y entradas de catálogo, la información que se puede traducir, por ejemplo, las descripciones.

A continuación se ofrece una visión general del proceso de creación del catálogo:

1. En WebSphere Commerce, un catálogo se crea utilizando archivos XML. La creación de un catálogo empieza con una entidad de catálogo, que equivale a un catálogo de papel en la base de datos.
2. Cree la estructura y el flujo de navegación añadiendo grupos de catálogo que determinen las categorías y la distribución de la mercancía.
3. Cree información de inventario como base para las entradas de catálogo.
4. Añada la mercancía en forma de entradas de catálogo, que representan productos, códigos de artículos, paquetes, paquetes compuestos kits estáticos y kits dinámicos.
5. A los productos del catálogo se les añaden atributos y valores de atributo para diferenciar los distintos artículos.
6. Puede crear paquetes y paquetes compuestos para agrupar ciertas entradas de catálogo, con fines promocionales.
7. A continuación se crean las relaciones entre los grupos de catálogo y las entradas de catálogo. Estas relaciones determinan qué entradas pertenecen a un grupo de catálogo.
8. Puede crear asociaciones de comercialización para las entradas de catálogo, como estrategias de recomendación de productos.
9. Asocie el catálogo, los grupos de catálogo y las entradas de catálogo a la tienda de WebSphere Commerce.
10. En el paso final, debe crear:
 - a. Impuestos para su mercancía.
 - b. Métodos de envío.
 - c. Un centro de despacho de pedidos que actúe como almacén de inventario y un centro de recepción. Una tienda puede tener definidos más de un centro de despacho de pedidos.
 - d. Precio de la mercancía.

Creación de un catálogo maestro

Para crear un catálogo maestro que contenga varios niveles de categorías, realice las tareas siguientes:

Parte 1: Preparación para la creación del catálogo

1. Revise la información del catálogo y sus modelos de objetos y datos correspondientes incluidos en WebSphere Commerce. La información del catálogo es un componente de WebSphere Commerce Server que hace posible la navegación, la división, el establecimiento de categorías y las asociaciones de la mercancía que se puede adquirir.
2. Revise la información de WebSphere Commerce Loader package. Loader package consta principalmente de utilidades para preparar y cargar datos en una base de datos de WebSphere Commerce. Puede utilizar Loader package para cargar grandes cantidades de datos y actualizar los datos de su base de datos. Para obtener más información sobre Loader package, consulte la Parte 10, "Publicación de la tienda", en la página 345.
3. Revise la información del Apéndice B, "Creación de los datos", en la página 465.
4. Cree una organización mediante la Consola de administración, para que actúe como propietario del catálogo. Para obtener más información, consulte el tema "Crear una organización" de la ayuda en línea de WebSphere Commerce.

5. Cree un nuevo archivo XML para su catálogo maestro utilizando como guía las entradas XML existentes y los archivos `catalog.xml` de la tienda de ejemplo ToolTech. Si crea un catálogo globalizado, cree un archivo `catalog.xml` para cada entorno nacional al que la tienda dé soporte. El archivo específico de entorno nacional debe especificar toda la información de las descripciones, para que se pueda traducir fácilmente. En este ejemplo, se utilizará un archivo `catalog.xml` para toda la información que no es necesario traducir y otro archivo `catalog.xml` para cada entorno nacional soportado por la tienda, con la información que debe traducirse. O, si lo prefiere, puede utilizar el archivo XML existente de la tienda de ejemplo ToolTech y cambiar la información según sea necesario. Los archivos `catalog.xml` de la tienda de ejemplo ToolTech se encuentran en su archivo archivador de tienda. Para ver los archivos `catalog.xml`, descomprima el archivador de tienda utilizando un programa ZIP. Los archivos `catalog.xml` se encuentran en el directorio de datos siguientes:

- `dir_instalación_WC/samplestores`

Nota: La ayuda en línea de WebSphere Commerce contiene información sobre cada uno de los elementos de datos contenidos en las tiendas de ejemplo.

El archivo `catalog.dtd` está ubicado en el directorio siguiente:

- `dir_instalación_WC/xml/sar`

Parte 2: Creación de una entrada de catálogo

1. Utilizando el siguiente ejemplo de la tienda ToolTech como muestra, cree una entidad de catálogo añadiendo información en las tablas CATALOG y CATALOGDSC. Una entidad de catálogo representa un catálogo en la base de datos.

```
<catalog
catalog_id="@catalog_id_1"
member_id="@seller_b2b_mbr_id"
identifier="ToolTech"
description="Catálogo de ToolTech"
tpclevel="0"
/>
```

donde

- `catalog_id` es el número de referencia interno.
 - `member_id` es el número de referencia interno que identifica al propietario del catálogo.
 - `identifier` es un nombre externo para el catálogo.
 - `description` es una descripción del catálogo.
2. Utilice el siguiente ejemplo de la tienda ToolTech de ejemplo, y añada la descripción del catálogo en el archivo XML específico del entorno nacional para fines de traducción:

```
<catalogdsc
catalog_id="@catalog_id_1"
language_id="&es_ES;"
name="Catálogo maestro de la tienda"
/>
```

donde

- `catalog_id` es el número de referencia interno que relaciona esta información específica del idioma con un catálogo.
- `language_id` es el identificador del idioma.
- `name` es el nombre dependiente del idioma de la entrada de catálogo.

Parte 3: Creación de grupos de catálogo

1. Utilizando el siguiente ejemplo de la tienda ToolTech como muestra, cree grupos de catálogo añadiendo información a las tablas CATGROUP y CATGRPDESC. Los grupos de catálogo, conocidos también como categorías, son agrupaciones de otros grupos de catálogo o de productos. Complete esta tarea para cada grupo de catálogo en su catálogo:

```
<catgroup
catgroup_id="@catgroup_id_1"
member_id="@seller_b2b_mbr_id"
identifier="Herramientas para madera"
markfordelete="0"
/>
```

donde

- `catgroup_id` es el número de referencia interno del grupo de catálogo
- `member_id` es el número de referencia interno que identifica al propietario del catálogo.
- `identifier` es un nombre externo para el catálogo.
- `markfordelete` indica si el grupo de catálogo se ha marcado para que se suprima.
 - 0 = no.
 - 1 = sí.

2. Utilice el siguiente ejemplo de la tienda ToolTech de ejemplo como guía, y añada la descripción del grupo de catálogo en el archivo XML específico del entorno nacional para fines de traducción. Complete esta tarea para cada grupo de catálogo en su catálogo:

```
<catgrpdesc
language_id="&es_ES;"
catgroup_id="@catgroup_id_1"
name="Herramientas para madera"
shortdescription="Herramientas para madera"
longdescription="Herramientas para madera"
published="1"
/>
```

donde

- `language_id` es el identificador del idioma.
- `catgroup_id` es el número de referencia interno del grupo de catálogo.
- `name` es el nombre dependiente del idioma del catálogo.
- `shortdescription` es una descripción breve del grupo de catálogo.
- `longdescription` es una descripción detallada del grupo de catálogo.
- `published` indica si este grupo de catálogo debe visualizarse para el idioma indicado por `language_id`:
 - 0 = no.
 - 1 = sí.

Nota: Cada vez que crea un grupo de catálogo y su descripción, se cambia `catgroup_id` para representar un nuevo grupo de catálogo. Por ejemplo, `catgroup_id="@catgroup_id_2"`, `catgroup_id="@catgroup_id_3"`, `catgroup_id="@catgroup_id_4"`, etc.

3. Después de crear los grupos de catálogo, asigne al catálogo un grupo de catálogo de nivel superior añadiendo información a la tabla CATTOGRP. Este grupo de catálogo es el padre de los grupos de catálogo inmediatamente debajo de él. Complete esta tarea para cada grupo de catálogo de nivel superior en su catálogo. Utilice el siguiente ejemplo de la tienda ToolTech como muestra:


```
<cattogrp
catalog_id="@catalog_id_1"
catgroup_id="@catgroup_id_1"
/>
```

donde

- catalog_id es el número de referencia del catálogo.
- catgroup_id es el número de referencia del grupo de catálogo.

Nota: Cada vez que asigna al catálogo grupos de catálogo de nivel superior, se cambia catgroup_id para representar una nueva asociación de grupo de catálogo. Por ejemplo, catgroup_id="@catgroup_id_2", catgroup_id="@catgroup_id_3", catgroup_id="@catgroup_id_4", etc.

4. Una vez determinada la estructura jerárquica para los grupos de catálogo, cree relaciones entre los grupos de catálogo añadiendo información a la tabla CATGRPREL. Complete esta tarea para cada estructura de grupo de catálogo padre e hijo de su catálogo. Utilice el siguiente ejemplo de la tienda ToolTech como muestra:

```
<catgrprel
catgroup_id_parent="@catgroup_id_1"
catgroup_id_child="@catgroup_id_11"
catalog_id="@catalog_id_1"
sequence="0"
/>
```

donde

- catgroup_id_parent es el grupo de catálogo origen de esta relación.
- catgroup_id_child es el grupo de catálogo destino de esta relación.
- catalog_id es el número de referencia del catálogo.
- sequence es el número que determina el orden de visualización del contenido del grupo de catálogo.

Nota: Con cada relación de grupo de catálogo, se cambian catgroup_id_child y sequence para representar una nueva relación. Por ejemplo, las relaciones subsiguientes se visualizarían como catgroup_id_child="@catgroup_id_12" y sequence="1", catgroup_id_child="@catgroup_id_13" y sequence="2", etc. Si no está utilizando una estructura de navegación en su catálogo, puede suprimir la relación CATGRPREL.

Parte 4: Creación de información de inventario

1. Utilice el siguiente ejemplo de la tienda de ejemplo ToolTech como guía y cree información de inventario añadiendo información a las tablas BASEITEM, BASEITEMDSC, ITEMSPC, ITEMVERSN, VERSIONSPC, DISTARRANG y STOREITEM. Empiece creando los artículos base añadiendo información a la tabla BASEITEM. Los artículos base representan una familia general de productos con un nombre y descripción comunes. Complete esta tarea para cada grupo de artículos de inventario de su catálogo:

```
<baseitem
baseitem_id="@baseitem_id_102"
member_id="@seller_b2b_mbr_id"
markfordelete="0"
partnumber="toolttech_sku_102"
itemtype_id="ITEM"
quantitymeasure="C62"
quantitymultiple="1.0"
/>
```

donde

- `baseitem_id` es la clave exclusiva generada.
- `member_id` es el propietario del artículo base.
- `markfordelete` indica si el artículo base se ha marcado para que se suprima.
 - 0 = no.
 - 1 = sí.
- `partnumber` identifica de forma exclusiva el artículo base para el propietario.
- `itemtype` es el tipo de artículo base:
 - ITEM = artículos, paquetes o paquetes compuestos.
 - DNKT = kits dinámicos.
 - STKT = kits estáticos.
- `quantitymeasure` es la unidad de medida para `quantitymultiple`.
- `quantitymultiple` es la cantidad de artículo base que se mide en unidades integrales. Junto con `quantitymeasure`, indica lo que representa cada unidad integral.

Nota: Debe crear un artículo base para cada producto que cree en el catálogo. Cada vez que crea un artículo base, los números `baseitem_id` y `partnumber` cambian para crear un nuevo artículo base. Por ejemplo, un nuevo artículo base podría contener `baseitem_id="@baseitem_id_147"` y `partnumber="tooltech_sku_147"` como entradas, mientras que otro podría contener `baseitem_id="@baseitem_id_192"` y `partnumber="tooltech_sku_192"`, etc.

2. Utilizando el siguiente ejemplo de la tienda ToolTech como muestra, añada información sobre artículos especificados a la base de datos. Un artículo especificado es un artículo con valores para todos sus atributos y representa un artículo, paquete, paquete compuesto o kit dinámico del catálogo. Complete esta tarea para cada artículo especificado del catálogo:

```
<itemspc
itemspc_id="@itemspc_id_106"
baseitem_id="@baseitem_id_102"
markfordelete="0"
partnumber="T0000106"
member_id="@seller_b2b_mbr_id"
discontinued="N"
/>
```

donde

- `itemspc_id` es la clave exclusiva generada.
- `baseitem_id` es el artículo base del producto.
- `markfordelete` indica si el artículo especificado se ha marcado para que se suprima.
 - 0 = no.
 - 1 = sí.
- `partnumber` identifica de forma exclusiva el artículo especificado para el propietario.
- `member_id` es el propietario del artículo especificado.
- `discontinued` indica si el artículo especificado ha dejado de fabricarse:
 - Y=ha dejado de fabricarse y puede solicitarse si hay suficiente inventario pero no podrán servirse en los pedidos en espera.
 - N=activo y pueden solicitarse en pedidos en espera si no quedan existencias.

Nota: Debe crear un artículo especificado para cada artículo que cree en el catálogo. Cada vez que define un artículo especificado, los números `itemspc_id="@itemspc_id_107"`, `baseitem_id="@baseitem_id_102"` y `partnumber="T0000107"` cambian para crear un nuevo artículo especificado. Por ejemplo, un nuevo artículo especificado podría contener `itemspc_id="@itemspc_id_108"`, `baseitem_id="@baseitem_id_102"` y `partnumber="T0000108"` como entradas, mientras que otro podría contener `itemspc_id`, `baseitem_id` y `partnumber` como entradas, etc.

3. Utilice el siguiente ejemplo de la tienda de ejemplo ToolTech como guía y añada a la base de datos la información siguiente sobre la relación entre la versión de un artículo y el artículo base. Complete esta tarea para cada relación de este tipo del catálogo:

```
<itemversn
itemversn_id="@itemversn_id_102"
baseitem_id="@baseitem_id_102"
expirationdate="2010-01-01 00:00:00.000000"
versionname="version"
/>
```

donde

- `itemversn_id` es un número de referencia generado que identifica la versión del artículo.
- `baseitem_id` es el artículo base.
- `expirationdate` es la hora en que caduca la versión del artículo.
- `versionname` identifica de forma exclusiva la versión del artículo para su artículo base.

Nota: Cada vez que crea una relación entre la versión de un artículo y el artículo base, se modifican los números `itemversn_id` y `baseitem_id` para crear una nueva relación. `baseitem_id` coincide con un artículo base existente. Por ejemplo, una relación nueva puede contener como entradas `itemversn_id="@itemversn_id_107"` y `baseitem_id="@baseitem_id_107"`, mientras que otra relación puede contener como entradas `itemversn_id="@itemversn_id_108"` y `baseitem_id="@baseitem_id_108"`, etc.

4. Utilizando el siguiente ejemplo de la tienda ToolTech como muestra, añada la información siguiente sobre la relación entre la versión de un producto y el artículo especificado en la base de datos. Complete esta tarea para cada relación de este tipo del catálogo:

```
<versionspc
versionspc_id="@versionspc_id_106"
itemspc_id="@itemspc_id_106"
itemversn_id="@itemversn_id_102"
/>
```

donde

- `versionspc_id` es el identificador exclusivo generado.
- `itemspc_id` es el artículo especificado con el que está relacionada esta entrada del catálogo.
- `itemversn_id` identifica la versión del artículo.

Nota: Cada vez que crea una relación entre la versión de un producto y el artículo especificado, los números `versionspc_id` e `itemspc_id` cambian para crear una nueva relación. `itemspc_id` coincide con un artículo especificado existente. Por ejemplo, una nueva relación podría contener como entradas `versionspc_id="@versionspc_id_107"` e

itemspc_id="@itemspc_id_107" , mientras que otra podría contener como entradas versionpc_id="@versionpc_id_108" e itemspc_id="@itemspc_id_108", etc.

5. Utilice el siguiente ejemplo de la tienda de ejemplo ToolTech como guía y añada el acuerdo de la distribución a la base de datos. Un acuerdo de distribución permite que una tienda venda su propio inventario. Complete esta tarea para cada acuerdo de distribución del catálogo:

```
<distarrang
distarrang_id="@distarrang_id_102"
wholesalestore_id="@storeent_id_1"
merchantstore_id="@storeent_id_1"
baseitem_id="@baseitem_id_102"
pickingmethod="F"
startdate="2000-12-25 00:00:00.000000"
enddate="2010-01-01 00:00:00.000000"
/>
```

donde

- distarrang_id es el número de referencia del acuerdo de distribución.
- wholesalestore_id es la tienda de ventas al mayor propietaria del inventario que la tienda del comerciante puede vender. Este ID debe ser igual que merchantstore_id.
- merchantstore_id es la tienda del comerciante que puede vender el inventario de la tienda de ventas al mayor. Este ID debe ser igual que wholesalestore_id.
- baseitem_id es el producto que cubre el acuerdo de distribución.
- pickingmethod determina la secuencia en la que se elige el inventario de la tabla RECEIPT según este acuerdo:
 - F = FIFO (primero en entrar primero en salir) : el inventario recibido con más anterioridad.
 - L = LIFO (último en entrar primero en salir) : el inventario recibido más recientemente.
- startdate es el hora en que este acuerdo de distribución empieza a entrar en vigor.
- enddate es la hora en que este acuerdo de distribución deja de estar en vigor.

Nota: Cada vez que crea un acuerdo de distribución, se modifican los números distarrang_id y baseitem_id para crear un nuevo acuerdo de distribución. Por ejemplo, un segundo acuerdo de distribución puede contener los valores distarrang_id="@distarrang_id_147" y baseitem_id="@baseitem_id_147", mientras que un tercero puede contener distarrang_id="@distarrang_id_192" y baseitem_id="@baseitem_id_192", etc.

6. Utilice el siguiente ejemplo de la tienda de ejemplo ToolTech como guía y añada a la base de datos los atributos que afectan cómo una tienda determinada asigna inventario para los artículos especificados de un artículo base determinado. Complete esta tarea para cada artículo base del catálogo:

```
<storeitem
baseitem_id="@baseitem_id_102"
storeent_id="@storeent_id_1"
trackinventory="Y"
forcebackorder="N"
releaseseparately="N"
returnnotdesired="N"
```

```
backorderable="Y"
creditable="Y"
minqtyforsplit="0"
/>
```

donde

- `baseitem_id` es el artículo base.
- `storeent_id` es la tienda del grupo de tiendas.
- `trackinventory` controla si hace un seguimiento o no del inventario en la tabla RECEIPT:
 - N = no se hace un seguimiento del inventario y no hay entradas en la tabla RECEIPT.
 - Y = se hace un seguimiento del inventario en la tabla RECEIPT.
- `forcebackorder` suspende temporalmente la asignación de los artículos especificados para el artículo base:
 - N = se puede asignar el inventario (comportamiento normal)
 - Y = no se puede asignar el inventario incluso si hay inventario suficiente.
- `releaseseperately` controla cómo se liberan los artículos de un pedido especificados para el artículo base:
 - N = se pueden liberar los artículos junto con los artículos de pedido.
 - Y = los artículos de pedido se deben liberar de forma independiente (en sus propias cajas)
- `returnnotdesired` indica que no se desea la devolución de un artículo (por ejemplo, los alimentos perecederos), incluso si el cliente está dispuesto o desea devolverlo:
 - N = se evalúa la petición de abono en base a la intención del cliente de devolver el artículo, aunque no se espera la devolución.
 - Y = se evalúa la petición de crédito como si se esperara la devolución.
- `backorderable` indica que no se puede suspender la asignación de los artículos especificados para el artículo base:
 - N = no se puede suspender la asignación de los artículos.
 - Y = se puede suspender la asignación de los artículos.
- `creditable` indica si el comerciante abonará este artículo, sin ninguna alteración temporal:
 - N = se vende como está.
 - Y = se puede abonar.
- `minqtyforsplit` indica que los artículos del pedido no se dividirán automáticamente durante la asignación de inventario si la cantidad restante que no se ha asignado en el nuevo artículo de pedido es menor que la cantidad mínima especificada.

Nota: Cada vez que define las normas de asignación de inventario para un artículo de tienda, se cambia el número de `baseitem_id` para que represente un nuevo artículo base. Por ejemplo, una asignación nueva puede contener `baseitem_id="@baseitem_id_147"` y una tercera asignación puede contener `baseitem_id="@baseitem_id_192"`, etc.

7. Utilice el siguiente ejemplo de la tienda de ejemplo ToolTech como guía y añada al archivo XML específico del entorno nacional la descripción del artículo base. Complete esta tarea para cada descripción de artículo base del catálogo:

```
<baseitmdsc
baseitem_id="@baseitem_id_102"
language_id="&es_ES;"
```

```

shortdescription="Sierra circular"
longdescription="Sierra de poco peso y gran calidad. La sierra circular
pesa como máximo 4,5 Kg. y tiene un motor de 12 ó 14 amp con una
velocidad de hasta 600 rpms. La base de aleación de aluminio de 220 V,
de fricción reducida, permite que el
trabajo se efectúe con gran rapidez."
/>

```

donde

- `baseitem_id` es la clave exclusiva generada.
- `language_id` es el idioma de esta información.
- `shortdescription` es una descripción breve del artículo base.
- `longdescription` es una descripción detallada del artículo base.

Parte 5: Creación de entradas de catálogo

1. Utilizando el siguiente ejemplo de la tienda ToolTech como muestra, cree entradas de catálogo añadiendo información a las tablas `CATENTRY` y `CATENTDESC`. Cada tipo de entrada de catálogo — productos, artículos, paquetes, paquetes compuestos y kits dinámicos — representa los artículos de mercancía a la venta en el catálogo. Tendrá que definir un artículo base para cada entrada de catálogo de producto. Complete esta tarea para cada entrada de catálogo de producto de su catálogo:

```

<catentry
catentry_id="@product_id_102"
baseitem_id="@baseitem_id_102"
member_id="@seller_b2b_mbr_id"
catenttype_id="ProductBean"
partnumber="T0000102"
mfpartnumber="Torsión-Herramientas-102"
mfname="Herramientas de torsión"
markfordelete="0"
buyable="1"
/>

```

donde

- `catentry_id` es el número de referencia interno de la entrada de catálogo de productos.
- `baseitem_id` es el artículo base con el que esta entrada de catálogo está relacionada.
- `member_id` es el número de referencia que identifica la entrada de catálogo.
- `catenttype_id` identifica el tipo de entrada de catálogo:
 - `ItemBean` = identifica un artículo.
 - `ProductBean` = identifica un producto.
 - `PackageBean` = identifica un paquete.
 - `BundleBean` = identifica un paquete compuesto.
 - `DynamicKitBean` = identifica un kit dinámico.
- `partnumber` es el número de referencia que identifica el número de pieza de la entrada de catálogo.
- `mfpartnumber` es el número de pieza que utiliza el fabricante para identificar la entrada de catálogo.
- `mfname` es el nombre del fabricante de la entrada de catálogo.
- `markfordelete` indica si la entrada de catálogo se ha marcado para que se suprima.
 - 0 = no.
 - 1 = sí.
- `buyable` indica si se puede adquirir la entrada de catálogo individualmente:

- 0 = no.
- 1 = sí.

Nota: Cada vez que añade un artículo base a una entrada de catálogo de producto, se cambia la secuencia de `catentry_id` y `baseitem_id` para representar una nueva entrada de catálogo. `catenttype_id` cambia según el tipo de la entrada de catálogo.

- Utilice el siguiente ejemplo de la tienda de ejemplo ToolTech como guía y defina un artículo especificado para cada entrada de catálogo. Complete esta tarea para cada entrada de catálogo:

```
<catentry
catentry_id="@catentry_id_106"
itemspc_id="@itemspc_id_106"
member_id="@seller_b2b_mbr_id"
catenttype_id="ItemBean"
partnumber="T0000106"
mfpartnumber="Torsión-Herramientas-106"
mfname="Herramientas de torsión"
markfordelete="0"
buyable="1"
/>
```

donde

- `catentry_id` es el número de referencia interno de la entrada de catálogo.
- `itemspc_id` es el artículo especificado al que pertenece esta entrada de catálogo.
- `member_id` es el número de referencia que identifica la entrada de catálogo.
- `catenttype_id` identifica el tipo de entrada de catálogo:
 - `ItemBean` = identifica un artículo.
 - `ProductBean` = identifica un producto.
 - `PackageBean` = identifica un paquete.
 - `BundleBean` = identifica un paquete compuesto.
 - `DynamicKitBean` = identifica un kit dinámico.
- `partnumber` es el número de referencia que identifica el número de pieza de la entrada de catálogo.
- `mfpartnumber` es el número de pieza que utiliza el fabricante para identificar la entrada de catálogo.
- `mfname` es el nombre del fabricante de la entrada de catálogo.
- `markfordelete` indica si la entrada de catálogo se ha marcado para que se suprima.
 - 0 = no.
 - 1 = sí.
- `buyable` indica si se puede adquirir la entrada de catálogo individualmente:
 - 0 = no.
 - 1 = sí.

Nota: Cada vez que añade un artículo especificado a una entrada de catálogo, se cambia la secuencia de `catentry_id` e `itemspc_id` para representar una nueva entrada de catálogo. `catenttype_id` cambia según el tipo de la entrada de catálogo. Debido a la limitación de la estructura del catálogo maestro, una entrada de catálogo no puede

pertenecer a más de una categoría. Para colocar una entrada de catálogo en más de una categoría, deberá utilizar un catálogo de ventas.

- Utilizando el siguiente ejemplo de la tienda ToolTech como muestra, añada la descripción al archivo XML específico del entorno nacional. Complete esta tarea para cada descripción de entradas de su catálogo:

```
<catentdesc
catentry_id="@product_id_102"
language_id="&es_ES"
name="Circular"
shortdescription="Sierra circular"
longdescription="Sierra de poco peso y gran calidad. La sierra circular
pesa como máximo 4,5 Kg. y tiene un motor de 12 ó 14 amp con una
velocidad de hasta 600 rpms. La base de aleación de aluminio de 220 V,
de fricción reducida, permite que el
trabajo se efectúe con gran rapidez."
thumbnail="images/circular_saw_sm.gif"
fullimage="images/circular_saw.gif"
available="1"
published="1"
/>
```

donde

- `catentry_id` es el número de referencia interno que indica la entrada de catálogo a la que hace referencia esta información específica del idioma.
- `language_id` es el identificador del idioma.
- `name` es el nombre dependiente del idioma de la entrada de catálogo.
- `shortdescription` es una descripción breve de la entrada de catálogo.
- `longdescription` es una descripción detallada de la entrada de catálogo.
- `thumbnail` es la vía de acceso de la imagen reducida.
- `fullimage` es la vía de acceso de la imagen de tamaño normal.
- `available` indica el periodo de tiempo que la entrada de catálogo estará disponible.
- `published` indica si esta entrada de catálogo debe mostrarse para el idioma indicado por `language_id`
 - 0 = visualizar.
 - 1 = no visualizar.

Parte 6: Creación de atributos y valores de atributo

1. Utilice el siguiente ejemplo de la tienda de ejemplo ToolTech como guía y cree atributos y valores de atributo para sus productos añadiendo información a las tablas `ATTRIBUTE` y `ATTRVALUE` en el archivo XML específico del entorno nacional para fines de traducción. Cada producto del catálogo tiene un conjunto de atributos específicos, como por ejemplo, la talla y el color de una camisa o unos pantalones. Los artículos se definen con los valores de atributo. Por ejemplo, una camisa es un producto, pero una camisa negra de talla pequeña es un artículo. Complete esta tarea para cada atributo del catálogo:

```
<attribute
attribute_id="@attribute_id_103"
language_id="&es_ES"
attrtype_id="STRING"
name="Amps"
sequence="0"
description="Amps"
catentry_id="@product_id_102"
description2="Amps"
/>
```


donde

- `attribute_id` es el número de referencia interno del atributo.
- `language_id` es el idioma al que pertenece este atributo.
- `attrtype_id` es el tipo del valor de atributo correspondiente.
- `name` es el nombre del atributo.
- `sequence` es un número de secuencia que determina el orden de visualización de los atributos de un producto específico.
- `description` es la descripción del atributo.
- `catentry_id` es el número de referencia del producto al que pertenece este atributo.
- `description2` es una descripción adicional del atributo.

Nota: Cada vez que añada un atributo a un producto definido mediante `catentry_id`, la secuencia de `attribute_id` cambia para representar un nuevo atributo.

2. Utilizando el siguiente ejemplo de la tienda ToolTech como muestra, añada los valores de atributo. Complete esta tarea para cada valor de atributo del catálogo:

```
<attrvalue
attrvalue_id="@attrvalue_id_114"
language_id="&es_ES"
attribute_id="@attribute_id_103"
name="14 amp"
attrtype_id="STRING"
stringvalue="14 amp"
sequence="0"
usage="1"
catentry_id="@catentry_id_106"
/>
```

donde

- `attrvalue_id` es el número de referencia interno del valor de atributo
- `language_id` es el idioma al que pertenece este valor de atributo
- `attribute_id` es el número de referencia interno del atributo asociado al valor
- `name` es el nombre del valor del atributo
- `attrtype_id` es el tipo del valor del atributo
- `stringvalue` es el valor del atributo
- `sequence` es un número de secuencia que determina el orden de visualización de los valores de un atributo específico.
- `usage` es el tipo de atributo:
 - 1 identifica un atributo de definición utilizado para la resolución de códigos de artículo.
 - 0 (u otro valor) identifica un atributo descriptivo.
- `catentry_id` es el ID de artículo que describe este valor de atributo

Nota: Cada vez que añada un valor a un atributo, la secuencia de `attrvalue_id` cambia para representar valores diferentes. La secuencia de `attribute_id` cambia para representar un atributo diferente. `sequence` aumenta con cada nuevo valor de atributo. Por ejemplo, los valores de atributo subsiguientes serían `sequence="1"`, `sequence="2"`, `sequence="3"`, etc.

Parte 7: Creación de relaciones entre productos y artículos

1. Después de crear productos y artículos para el catálogo, debe definir las relaciones entre ellos añadiendo información en la tabla CATENTREL. Utilice el siguiente ejemplo de la tienda ToolTech como muestra. Complete esta tarea para cada valor de relación entre producto y artículo del catálogo:

```
<catentrel
  catentry_id_parent="@product_id_147"
  catreltype_id="PRODUCT_ITEM"
  catentry_id_child="@catentry_id_152"
  sequence="2"
  quantity="1"
/>
```

donde

- `catentry_id_parent` es el número de referencia de la entrada de catálogo origen de esta relación, es decir, el producto.
- `catreltype_id` es el tipo de relación: `PRODUCT_ITEM`
- `catentry_id_child` es el número de referencia de la entrada de catálogo destino de esta relación, es decir, el artículo.
- `sequence` es el número de secuencia que se utiliza para determinar el orden de visualización.
- `quantity` es una cantidad que se puede asociar con la relación.

Nota: Cada vez que añade una relación entre un producto y un artículo, los números `catentry_id_parent` y `catentry_id_child` cambian para crear relaciones distintas, basados en `catreltype_id`. Con cada nueva relación, el número de `sequence` es distinto. Por ejemplo, si `sequence="2"`, el valor para la relación siguiente será `sequence="3"`, después `sequence="4"`, etc.

Parte 8: Creación de paquetes y paquetes compuestos

1. Una vez haya creado los productos y los artículos, debe crear los paquetes y paquetes compuestos añadiendo información a las tablas CATENTRY, CATENTDESC y CATENTREL. Por ejemplo, utilice el siguiente código de ejemplo para crear un paquete o un paquete compuesto añadiendo información a la tabla CATENTRY. Complete esta tarea para cada paquete y paquete compuesto de su catálogo:

```
<catentry
  catentry_id="@package_id_102"
  member_id="@seller_b2b_mbr_id"
  catenttype_id="PackageBean"
  partnumber="sku-@package_id_102"
  mfpartnumber="sku-@package_id_102"
  mfname="ToolTech"
  markfordelete="0"
  buyable="1"
/>
```

donde

- `catentry_id` es el número de referencia de la entrada de catálogo.
- `member_id` es el número de referencia que identifica al propietario de la entrada de catálogo.
- `catenttype_id` identifica el tipo de entrada de catálogo:
 - `PackageBean` = identifica un paquete.
 - `BundleBean` = identifica un paquete compuesto.
- `partnumber` es el número de referencia que identifica el número de pieza de la entrada de catálogo.

- `mfpnumber` es el número de pieza que utiliza el fabricante para identificar la entrada de catálogo.
- `mfname` es el nombre del fabricante de la entrada de catálogo.
- `markfordelete` indica si la entrada de catálogo se ha marcado para que se suprima.
 - 0 = no.
 - 1 = sí.
- `buyable` indica si la entrada de catálogo puede adquirirse individualmente:
 - 0 = no.
 - 1 = sí.

Nota: Cada vez que crea un paquete o un paquete compuesto, se cambian los números de `catentry_id`, `partnumber` y `mfpnumber` para crear un paquete o paquete compuesto diferente. Por ejemplo, para crear un paquete nuevo puede utilizar `catentry_id="@package_id_103"`, `partnumber="sku-@package_id_103"` y `mfpnumber="sku-@package_id_103"`, incluyendo `catenttype_id="PackageBean"` para identificar la entrada como un paquete. Para crear un paquete compuesto nuevo, puede utilizar `catentry_id="@package_id_110"`, `partnumber="sku-@package_id_110"` y `mfpnumber="sku-@package_id_110"` incluyendo `catenttype_id="BundleBean"` para identificar la entrada como un paquete compuesto, etc.

- Por ejemplo, utilice el siguiente código de ejemplo para añadir la descripción del paquete o paquete compuesto añadiendo información a la tabla `CATENTDESC` del archivo XML específico del entorno nacional para fines de traducción. Complete esta tarea para cada descripción de paquete o paquete compuesto del catálogo:

```
<catentdesc
catentry_id="@catentry_id_102"
language_id="-1"
name="PC"
shortdescription="PC"
longdescription="Equipo formado por una unidad central de proceso,
un monitor,
un disco duro y una impresora a color. Un PC ideal
para principiantes."
thumbnail="images/package_system_sm.gif"
fullimage="images/package_system.gif"
available="1"
published="1"
/>
```

donde

- `catentry_id` es el número de referencia interno que indica la entrada de catálogo a la que hace referencia esta información específica del idioma.
- `language_id` es el identificador del idioma.
- `name` es el nombre dependiente del idioma de la entrada de catálogo.
- `shortdescription` es una descripción breve de la entrada de catálogo.
- `longdescription` es una descripción detallada de la entrada de catálogo.
- `thumbnail` es la vía de acceso a la imagen reducida de la entrada de catálogo.
- `fullimage` es la vía de acceso a la imagen de tamaño normal de la entrada de catálogo.
- `available` indica el periodo de tiempo que la entrada de catálogo estará disponible.

- published indica si la entrada de catálogo debe mostrarse para el idioma indicado por language_id
 - 0 = no mostrar la entrada de catálogo.
 - 1 = mostrar la entrada de catálogo.
- Por ejemplo, utilice el siguiente código de ejemplo para crear relaciones entre paquetes o paquetes compuestos añadiendo información a la tabla CATENTREL. Complete esta tarea para cada relación de componente de paquete y paquete compuesto del catálogo:

```
<catentrel
catentry_id_parent="@catentry_id_102"
catreltype_id="PACKAGE_COMPONENT"
catentry_id_child="@catentry_id_97"
sequence="1.0"
quantity="1.0"
/>
```

donde

- catentry_id_parent es el número de referencia de la entrada de catálogo origen de esta relación, es decir, el paquete o paquete compuesto.
- catreltype_id es el tipo de relación:
 - PACKAGE_COMPONENT representa una relación entre un paquete y sus componentes.
 - BUNDLE_COMPONENT representa una relación entre un paquete compuesto y sus componentes.
- catentry_id_child es el número de referencia de la entrada de catálogo destino de esta relación, es decir, el componente.
- sequence es el número de secuencia que se utiliza para determinar el orden de visualización.
- quantity es una cantidad que se puede asociar con la relación.

Nota: Cada vez que crea una relación entre un paquete y un paquete compuesto, los números de catentry_id_parent y catentry_id_child cambian para que coincidan con entradas de catálogo existentes. Con cada relación nueva, el número de sequence es diferente. Por ejemplo, si empieza con sequence="1.0", el valor de la relación siguiente será sequence="2.0", después sequence="3.0", etc.

Parte 9: Creación de relaciones entre grupos de catálogo y entradas de catálogo

1. Después de crear grupos de catálogo y entradas de catálogo, debe definir las relaciones entre ellos, añadiendo información a la tabla CATGPENREL. Debido a la limitación de la estructura del catálogo maestro, una entrada de catálogo no puede pertenecer a más de una categoría. Para colocar una entrada de catálogo en más de una categoría, deberá utilizar un catálogo de ventas. Utilice el siguiente ejemplo de la tienda ToolTech como muestra. Complete esta tarea para cada relación entre un grupo de catálogo y una entrada de catálogo:

```
<catgpenrel
catgroup_id="@catgroup_id_11"
catalog_id="@catalog_id_1"
catentry_id="@product_id_102"
sequence="0"
/>
```

donde

- catgroup_id es el grupo de catálogo origen de esta relación.
- catalog_id es el catálogo dentro del cual se encuentra esta relación.

- catentry_id es la entrada de catálogo destino de esta relación.
- sequence es el número de secuencia que determina el orden de visualización del contenido del grupo de catálogo.

Nota: Cada vez que crea una relación entre grupos de catálogo y entradas de catálogo, los números catgroup_id y catentry_id cambian para formar nuevas relaciones con distintos grupos de catálogo y entradas de catálogo. Con cada nueva relación, el número de sequence es distinto. Por ejemplo, si empieza con sequence="0", el valor para la relación siguiente será sequence="1", después sequence="2", etc.

Parte 10: Creación de asociaciones de comercialización

1. Por ejemplo, utilice el siguiente código de ejemplo para crear entradas de asociaciones de comercialización añadiendo información a la tabla MASSOCECE. Complete esta tarea para cada asociación de comercialización del catálogo:

```
<massoccece
massoccece_id="@relationship_id_100"
massoctype_id="X-SELL"
catentry_id_from="@product_id_1"
catentry_id_to="@product_id_15"
massoc_id="REQUIRES"
quantity="2.0"
rank="1.00000"
/>
```

donde

- massoccece_id es el número de referencia de esta entrada.
- massoctype_id es el identificador del tipo de asociación:
 - X-SELL = venta cruzada.
 - UPSELL = venta ascendente.
 - ACCESSORY = accesorio.
 - REPLACEMENT = sustitución.
- catentry_id_from es la entrada de catálogo origen de la asociación.
- catentry_id_to es la entrada de catálogo destino de la asociación.
- massoc_id es el identificador del especificador semántico:
 - REQUIRED
 - OPTIONAL
 - COMES WITH
- quantity es la cantidad relacionada con esta asociación.
- rank es el número de secuencia que se utiliza para el orden de visualización.

Nota: Cada vez que añade una asociación de comercialización, el número de massoccece_id cambia para representar una nueva relación. Los números de catentry_id_from y catentry_id_to varían para crear nuevo contenido de mercancía para la asociación.

Parte 11: Asociación del catálogo a una tienda

1. Asocie su catálogo a una tienda, asignando el catálogo, sus grupos de catálogo y entradas de catálogo a una tienda en la base de datos, utilizando como muestra el archivo store-catalog.xml existente de la tienda de ejemplo ToolTech. También debe asignar páginas de visualización a los grupos de catálogo y a las entradas de catálogo. Añada esta información a las tablas STORECAT, STORECENT, STORECGRP, DISPCGPREL y DISPENTREL. Si crea un catálogo

globalizado, cree un archivo XML de relación entre el catálogo y la tienda para cada entorno nacional al que la tienda dé soporte:

```
<storecat  
catalog_id="@catalog_id_1"  
storeent_id="@storeent_id_1"  
mastercatalog="1"  
>
```

donde

- catalog_id es el número de referencia del catálogo.
- storeent_id es el número de referencia de la entidad de tienda en la base de datos.
- mastercatalog especifica un catálogo maestro para la tienda. El valor 1 indica que este catálogo está designado como catálogo maestro.

2. Utilizando como guía el siguiente ejemplo de la tienda de ejemplo ToolTech, añada entradas de catálogo a la relación entre tienda y catálogo. Complete esta tarea para cada entrada de catálogo:

```
<storecent  
storeent_id="@storeent_id_1"  
catentry_id="@product_id_102"  
>
```

donde

- storeent_id es el número de referencia de la entidad de tienda en la base de datos.
- catentry_id es el número de referencia de la entrada de catálogo.

Nota: Cada vez que añada un valor para catentry_id a la entidad de tienda, el número de referencia cambia para que coincida con una entrada de catálogo existente.

3. Utilizando el siguiente ejemplo de la tienda ToolTech como muestra, añada grupos de catálogo a la entidad de tienda. Complete esta tarea para cada grupo de catálogo en su catálogo:

```
<storecgrp  
storeent_id="@storeent_id_1"  
catgroup_id="@catgroup_id_1"  
>
```

donde

- storeent_id es el número de referencia de la entidad de tienda en la base de datos.
- catgroup_id es el número de referencia del grupo de catálogo.

Nota: Cada vez que añada un valor para catgroup_id a la entidad de tienda, el número de referencia cambia para que coincida con un grupo de catálogo existente.

Parte 12: Asociación de impuestos al catálogo

Asocie impuestos a los productos y servicios del catálogo para una tienda específica. Debe asociar un código de cálculo a las entradas de catálogo, añadiendo esta información a la tabla CATENCALCD. Para obtener más información, consulte "Creación de elementos de impuestos en WebSphere Commerce" en la página 268.

Parte 13: Asociación de métodos de envío al catálogo

Para asociar métodos de envío a los productos y servicios del catálogo, deberá asociar un código de cálculo de envío a las entradas de catálogo. Añada esta

información a la tabla CATENCALCD. Para obtener más información, consulte “Creación de elementos de envío en WebSphere Commerce” en la página 251.

Parte 14: Asociación de un centro de despacho de pedidos al catálogo

Para enviar productos a los clientes, debe asociar el catálogo a un centro de despacho de pedidos. Un centro de despacho gestiona las existencias y el envío de productos para una tienda. Añada esta información a la tabla FFMCCENTER. Para obtener más información, consulte “Creación de elementos de despacho de pedidos en WebSphere Commerce” en la página 218.

Parte 15: Creación de precios para las entradas de catálogo

Cree los precios para las entradas de catálogo. La fijación de precios representa el rango de precios para una entrada de catálogo y los criterios que deben satisfacerse para utilizar ese precio. Para crear un catálogo operativo, debe añadir información sobre ofertas a la base de datos. Añada esta información a las tablas TRADEPOSCN, TDPSCNCNTR, MGPTRDPSCN, OFFER y OFFERPRICE. Para obtener más información, consulte “Creación de elementos de fijación de precios en WebSphere Commerce” en la página 191. Alternativamente, puede crear o actualizar los precios para una entrada de catálogo utilizando las herramientas de Gestión de productos de WebSphere Commerce Accelerator.

Parte 16: Carga del archivo XML

Después de crear los datos, cargue el archivo XML en la base de datos utilizando Loader package o mediante el programa de utilidad de publicación. Para obtener más información sobre Loader package, consulte la Parte 10, “Publicación de la tienda”, en la página 345.

Nota: También puede utilizar las herramientas de Gestión de productos de WebSphere Commerce Accelerator para crear elementos de catálogo para su catálogo maestro. Para obtener información más detallada sobre las herramientas de Gestión de productos, consulte la ayuda en línea de WebSphere Commerce.

Visualización de los elementos de catálogo de la tienda

Después de asociar un catálogo, grupos de catálogo y entradas de catálogo a una tienda, asigne plantillas JSP para visualizar las entradas de catálogo y los grupos de catálogo creando estas relaciones en la base de datos. Cree estas relaciones en formato de archivos XML que se pueden cargar en la base de datos utilizando Loader package.

El archivo store-catalog.xml de la tienda de ejemplo ToolTech se encuentra en su archivo archivador de tienda. Para ver el archivo store-catalog.xml, descomprima el archivador de tienda utilizando un programa ZIP. El archivo store-catalog.xml se encuentra en el directorio de datos siguiente:

- *dir_instalación_WC/samplestores*

El archivo store-catalog.dtd se encuentra en el directorio siguiente:

- *dir_instalación_WC/xml/sar*

Antes de crear las relaciones entre tienda y catálogo, debe asegurarse de que ha creado los elementos de tienda. Complete las tareas siguientes, cada una de las cuales crea entradas en el archivo store-catalog.xml:

1. Para poder visualizar los grupos de catálogo (categorías) en la tienda, debe asignar plantillas JSP a los grupos de catálogo. Puede asignar una plantilla específica de página de visualización a un grupo de catálogo o una plantilla

por omisión para visualizar todos los grupos de catálogo. Utilizando el siguiente ejemplo de la tienda ToolTech como muestra, asigne plantillas de grupo de catálogo añadiendo información a la tabla DISPCGPREL. Complete esta tarea para cada plantilla que desee asignar a los grupos de catálogo:

```
<dispcgprel
catgroup_id="@catgroup_id_1"
devicefmt_id="-1"
dispcgprel_id="@dispcgprel_id_1"
mbrgrp_id="0"
pagename="CategoryDisplay.jsp"
storeent_id="@storeent_id_1"
rank="0"/>
```

donde

- `catgroup_id` es el número de referencia del grupo de catálogo para el que se utilizará este nombre de página. El valor 0 indica que este nombre de página se utilizará para todos los grupos de catálogo.
- `devicefmt_id` es el número de referencia del tipo de dispositivo en el que se visualizará la página. El valor -1 indica que esta página de plantilla la utilizará un navegador HTTP.
- `dispcgprel_id` es el número de referencia de esta entrada.
- `mbrgrp_id` es el número de referencia del grupo de miembros para el que se utilizará esta página de plantilla. El valor 0 indica que esta página de plantilla se utilizará para todos los grupos de miembros.
- `pagename` es el nombre de la página de la plantilla de visualización.
- `rank` es un número de secuencia que se utiliza cuando más de una página satisface los criterios de selección.

Nota: Cada vez que asigna una plantilla JSP a un grupo de catálogo, `catentry_id` cambia la secuencia para que coincida con una entrada de catálogo existente.

2. Para poder visualizar las entradas de catálogo (productos, artículos, paquetes, paquetes compuestos y kits dinámicos) en la tienda, debe asignar plantillas JSP a las entradas de catálogo. Puede asignar una plantilla por omisión para visualizar todas las entradas de catálogo o una plantilla por omisión para visualizar cada tipo de entrada de catálogo; por ejemplo, una plantilla para los productos y otra para los artículo, o una plantilla específica para una entrada de catálogo específica. Utilizando el siguiente ejemplo de la tienda ToolTech como muestra, asigne plantillas añadiendo información a la tabla DISPENTREL. Complete esta tarea para cada plantilla que desee asignar a las entradas de catálogo:

```
<dispentrel
auctionstate="0"
catentry_id="0"
catenttype_id="ProductBean"
devicefmt_id="-1"
dispentrel_id="@dispentrel_id_1"
mbrgrp="0"
pagename="ProductDisplay.jsp"
storeent_id="@storeent_id_1"
rank="0"/>
```

donde

- `auctionstate` indica que esta página de plantilla muestra una entrada de catálogo que está en una subasta:
 - 0 = no es una plantilla de subasta.
 - 1 = plantilla de subasta.

- `catentry_id` es el número de referencia de la entrada de catálogo para la que se utilizará este nombre de página. El valor 0 indica que este nombre de página se utilizará para todas las entradas de catálogo.
- `catenttype_id` es el tipo de entrada de catálogo para el que se utilizará esta página para su visualización:
 - `ProductBean` = visualiza un producto.
 - `ItemBean` = visualiza un artículo.
 - `PackageBean` = visualiza un paquete.
 - `BundleBean` = visualiza un paquete compuesto.
 - `DynamicKitBean` = visualiza un kit dinámico.
- `devicefmt_id` es el número de referencia del tipo de dispositivo en el que se visualizará la página. El valor -1 indica que esta página de plantilla la utilizará un navegador HTTP.
- `dispentrel_id` es el número de referencia de la entrada de catálogo.
- `mbrgrp` es el número de referencia del grupo de miembros para el que se utilizará esta página de plantilla. El valor 0 indica que esta página de plantilla se utilizará para todos los grupos de miembros.
- `pagename` es el nombre de la página de la plantilla de visualización.
- `storeent_id` es el número de referencia de la tienda para la que se utilizará esta página.
- `rank` es un número de secuencia que se utiliza cuando más de una página satisface los criterios de selección.

Nota: Cada vez que asigna una plantilla JSP a una entrada de catálogo, `catentry_id` cambia la secuencia para que coincida con una entrada de catálogo existente.

Creación de un catálogo de ventas

Una tienda de WebSphere Commerce permite dos tipos de catálogos: maestro y de ventas. Los catálogos de ventas no han de cumplir las restricciones estructurales que sí deben cumplir los catálogos maestros. Los catálogos de ventas están diseñados para proporcionar una estructura de visualización flexible que le permita crear un catálogo que se adapte a los requisitos de la tienda.

En particular, los catálogos de ventas no necesitan cumplir las restricciones siguientes que se imponen a los catálogos maestros:

- Un catálogo maestro debe ser un árbol apropiado, lo que significa que no hay ciclos y no se puede utilizar la estructura siguiente: La categoría padre A tiene una subcategoría B. Es importante que B y cualquiera de las subcategorías de B no sean la categoría padre de A.
- Un producto no puede pertenecer a más de una categoría.

La tarea siguiente crea un catálogo de ventas modificando el catálogo de la tienda de ejemplo FashionFlow. El catálogo resultante no puede seguir clasificándose como catálogo maestro puesto que los pasos siguientes clasifican algunos productos en varias categorías. Un catálogo de ventas clásico se crea añadiendo información a las tablas de relaciones de categoría: `CATGRPREL`, que contiene las relaciones entre subcategorías y `CATGPENREL`, que contiene las relaciones entre categorías y productos. Aunque estos ejemplos incluyen FashionFlow, puede seguir

estos pasos básicos con su propio catálogo maestro, realizando los ajustes necesarios para que coincidan con la información, la estructura y los diseños de su catálogo.

Adición de un producto a una segunda categoría

Este ejemplo le muestra cómo copiar productos de una categoría a otra al mismo tiempo que se conserva la estructura original. La categoría **Promociones de la página de presentación** contiene el producto **Camisón de verano**, que también podría pertenecer a la subcategoría **Pijamas** de la categoría superior **Moda para señoras**. Estas instrucciones le mostrarán cómo copiar el producto **Camisón de verano** y sus códigos de artículo en la categoría **Pijamas**.

Para cambiar el catálogo maestro de la tienda de ejemplo FashionFlow y convertirlo en un catálogo de ventas añadiendo un producto a una segunda categoría, realice lo siguiente:

1. Publique el archivador de tienda de FashionFlow para crear la tienda de ejemplo FashionFlow. FashionFlow está disponible en inglés de EE.UU. y en uno de los nueve idiomas nacionales que se entregan con WebSphere Commerce. Elija uno de los archivos `FashionFlow_entorno_nacional.sar` para la publicación.
2. Abra el archivo `catalog.xml` en un editor. El archivo se encuentra en el siguiente directorio de WebSphere Commerce:

- `dir_instalación_WC/samplestores/FashionFlow/entorno_nacional/data`

3. Localice la sección de datos `CATGPENREL` del archivo `catalog.xml`. Cree una entrada de producto nueva para **Camisón de verano**, que originalmente era un producto bajo la categoría **Promociones de la página de presentación**. Bajo la sección `CATGPENREL`, añada el fragmento siguiente para incluir el producto:

```
<catgpenrel
catgroup_id="@catgroup_id_18"
catalog_id="@catalog_id_1"
catentry_id="@product_id_2692"
sequence="2"
/>
```

donde

- `catgroup_id` es el número de referencia interno de grupo de catálogo definido por la tienda de ejemplo FashionFlow. En este ejemplo, `@catgroup_id_18` es la categoría **Moda para señoras**.
 - `catalog_id` es el número de referencia interno del catálogo definido por la tienda de ejemplo FashionFlow.
 - `catentry_id` es el número de referencia interno de entrada de catálogo definido por la tienda de ejemplo FashionFlow. En este ejemplo, `@catentry_id_2692` es el producto **Camisón de verano**.
 - `sequence` es el número que determina el orden de visualización del contenido del grupo de catálogo definido por la tienda de ejemplo FashionFlow. En este ejemplo, el producto **Camisón de verano** se visualizará en último lugar.
4. Después de añadir la entrada de producto **Camisón de verano**, añada las entradas de código de artículo para el producto bajo la sección `CATGPENREL`, tal como se han definido en la tienda de ejemplo FashionFlow. Actualmente, el producto **Camisón de verano** contiene diez códigos de artículo definidos. Bajo la sección `CATGPENREL`, añada los fragmentos siguientes para incluir los códigos de artículo:

```
<catgpenrel
catgroup_id="@catgroup_id_18"
catalog_id="@catalog_id_1"
```

```
catentry_id="@catentry_id_2695"  
sequence="2"  
</>
```

```
<catgpenrel  
catgroup_id="@catgroup_id_18"  
catalog_id="@catalog_id_1"  
catentry_id="@catentry_id_2696"  
sequence="2"  
</>
```

```
<catgpenrel  
catgroup_id="@catgroup_id_18"  
catalog_id="@catalog_id_1"  
catentry_id="@catentry_id_2697"  
sequence="2"  
</>
```

```
<catgpenrel  
catgroup_id="@catgroup_id_18"  
catalog_id="@catalog_id_1"  
catentry_id="@catentry_id_2698"  
sequence="2"  
</>
```

```
<catgpenrel  
catgroup_id="@catgroup_id_18"  
catalog_id="@catalog_id_1"  
catentry_id="@catentry_id_2699"  
sequence="2"  
</>
```

```
<catgpenrel  
catgroup_id="@catgroup_id_18"  
catalog_id="@catalog_id_1"  
catentry_id="@catentry_id_2700"  
sequence="2"  
</>
```

```
<catgpenrel  
catgroup_id="@catgroup_id_18"  
catalog_id="@catalog_id_1"  
catentry_id="@catentry_id_2701"  
sequence="2"  
</>
```

```
<catgpenrel  
catgroup_id="@catgroup_id_18"  
catalog_id="@catalog_id_1"  
catentry_id="@catentry_id_2702"  
sequence="2"  
</>
```

```
<catgpenrel  
catgroup_id="@catgroup_id_18"  
catalog_id="@catalog_id_1"  
catentry_id="@catentry_id_2703"  
sequence="2"  
</>
```

```
<catgpenrel  
catgroup_id="@catgroup_id_18"  
catalog_id="@catalog_id_1"  
catentry_id="@catentry_id_2704"  
sequence="2"  
</>
```

donde

- `catgroup_id` es el número de referencia interno de grupo de catálogo definido por la tienda de ejemplo FashionFlow. En este ejemplo, `@catgroup_id_18` es la categoría **Moda para señoras**.
 - `catalog_id` es el número de referencia interno del catálogo definido por la tienda de ejemplo FashionFlow.
 - `catentry_id` es el número de referencia interno de entrada de catálogo definido por la tienda de ejemplo FashionFlow. En este ejemplo, `@catentry_id_2695` a `@catentry_id_2704` representan los diez códigos de artículo que se han definido para el producto **Camisión de verano**.
 - `sequence` es el número que determina el orden de visualización del contenido del grupo de catálogo definido por la tienda de ejemplo FashionFlow. En este ejemplo, los códigos de artículo de **Camisión de verano** se visualizarán en último lugar.
5. Guarde el archivo `catalog.xml`.
 6. Para ver los cambios, realice una de las acciones siguientes: publique el archivador de tienda modificado de FashionFlow mediante Loader package o cargue el archivo `catalog.xml` con Loader package siguiendo las instrucciones del apartado “Carga de grupos de elementos de base de datos” en la página 413.

Gestión de elementos de catálogo en WebSphere Commerce

Con el tiempo, tendrá que actualizar la información de elementos de base de datos en el catálogo maestro. El mantenimiento del catálogo es un proceso continuo, ya que constantemente tendrá que añadir y eliminar mercancías, crear y asociar categorías o grupos de catálogo y actualizar información sobre los productos, como la descripción y el precio.

Puede cambiar los elementos de catálogo editando los datos XML de WebSphere Commerce utilizando las entradas de base de datos existentes y los archivos `catalog.xml` de la tienda. Como referencia, utilice los archivos XML de la tienda de ejemplo de WebSphere Commerce que se encuentran en el siguiente directorio de datos:

- `dir_instalación_WC/samplestores`

Nota: Estos ejemplos proceden de la tienda de ejemplo FashionFlow e identifican los elementos XML que deben modificarse para cambiar la información de los elementos de catálogo.

Grupos de catálogo

Los grupos de catálogo se crean en un catálogo de WebSphere Commerce utilizando las tablas de base de datos `CATGROUP` y `CATGRPDESC`. En el archivo `catalog.xml`, un grupo de catálogo típico se parece al extracto siguiente:

```
<catgroup
  catgroup_id="@catgroup_id_1"
  member_id="%MEMBER_ID"
  identifier="Accesorios"
  markfordelete="0"
/>
```

`catgroup_id` es el número de referencia interno del grupo de catálogo. A cada grupo de catálogo se le asigna un número de referencia interno en WebSphere Commerce, que identifica el grupo cuando se añaden entradas de catálogo. `identifier` es un nombre externo para el grupo de catálogo. Ambos elementos son exclusivos en los elementos de base de datos y no se pueden duplicar.

Los nombres y las descripciones pertenecen al archivo catalog.xml específico de entorno nacional; se necesita un archivo para cada entorno nacional soportado por la tienda. Un grupo de catálogo típico que contenga información traducible se parece al extracto siguiente:

```
<catgrpdesc
language_id="&es_ES"
catgroup_id="@catgroup_id_1"
name="Accesorios"
shortdescription="Accesorios"
longdescription="Accesorios"
published="1"
/>
```

language_id identifica el idioma de la información del catálogo. Este identificador debe cambiar y coincidir con cada uno de los idiomas soportados por la tienda. El cliente ve el elemento name, al igual que los elementos shortdescription y longdescription, que pueden contener descripciones breves y detalladas del grupo de catálogo.

Cuando cree un nuevo grupo de catálogo, siga la estructura anterior para la información.

Notas:

1. Aunque los elementos identifier y name sean idénticos en el ejemplo anterior, el contenido puede variar. Por ejemplo, quizá quiera renombrar el grupo de catálogo a **Adiciones complementarias**. En ese caso, no es necesario cambiar la información en identifier, sólo name.
2. Al suprimir grupos de catálogo, asegúrese de actualizar las apariciones de catgroup_id. Por ejemplo, si también quiere suprimir las entradas de catálogo bajo el grupo de catálogo, elimine todas las entradas XML. Sin embargo, si piensa mantener las entradas de catálogo, deberá cambiar catgroup_id al grupo correcto.

Entradas de catálogo

Las entradas de catálogo se crean en un catálogo de WebSphere Commerce utilizando la información de las tablas de base de datos CATENTRY y CATENTDESC. Una entrada de catálogo puede ser un producto, un artículo, un paquete, un paquete compuesto, un kit estático o un kit dinámico. En el archivo catalog.xml, una entrada de catálogo típica es similar al extracto siguiente:

```
<catentry
catentry_id="@product_id_102"
baseitem_id="@baseitem_id_102"
member_id="MEMBER_ID"
catentype_id="ProductBean"
partnumber="product-sku-nf-102"
mfpartnumber="product-sku-nf-102"
mfname="FashionFlow"
markfordelete="0"
buyable="1"
/>
```

catentry_id es el número de referencia interno de la entrada de catálogo de productos. baseitem_id es el artículo base con el que esta entrada de catálogo está relacionada, a efectos de inventario. partnumber es el número de referencia que identifica el número de pieza de la entrada de catálogo. mfpartnumber es el número de pieza que utiliza el fabricante para identificar la entrada de catálogo. Estos elementos son exclusivos en los elementos de base de datos y no se pueden duplicar.

catenttype_id identifica el tipo de entrada de catálogo: ItemBean, ProductBean, PackageBean, StaticBean, BundleBean o DynamicKitBean.

Los nombres y las descripciones pertenecen al archivo catalog.xml específico de entorno nacional; se necesita un archivo para cada entorno nacional soportado por la tienda. Las imágenes de los productos también se incluyen en este archivo. Un grupo de catálogo típico que contenga información traducible se parece al extracto siguiente:

```
<catentdesc
catentry_id="@product_id_102"
language_id="&es_ES"
name="Cinturón"
shortdescription="Cinturón clásico"
longdescription="Este cinturón clásico es un complemento ideal para sus vaqueros favoritos o para ir elegante al trabajo. En piel, de 3,5 cm de ancho y con hebilla metálica."
thumbnail="images/mens_accessories_belt_sm.gif"
fullimage="images/mens_accessories_belt.gif"
available="1"
published="1"
/>
```

language_id identifica el idioma en que está la información del catálogo. Este identificador debe cambiar y coincidir con cada uno de los idiomas soportados por la tienda. El cliente ve el elemento name, al igual que los elementos shortdescription y longdescription, que pueden contener descripciones breves y detalladas de la entrada de catálogo.

Cuando cree una nueva entrada de catálogo, siga la estructura anterior para la información.

Notas:

1. Al suprimir entradas de catálogo, asegúrese de actualizar cada aparición de los elementos exclusivos. Por ejemplo, si también quiere suprimir las entradas de catálogo bajo el grupo de catálogo, elimine todas las entradas XML. Sin embargo, si piensa mantener las entradas de catálogo, deberá cambiar catgroup_id al grupo correcto.
2. Los productos deben crearse antes que otros tipos de entradas de catálogo.

Si no desea cambiar manualmente los archivos XML, puede utilizar las herramientas de Gestión de productos.

Herramientas de Gestión de productos

Las herramientas de Gestión de productos de WebSphere Commerce Accelerator le permiten gestionar los productos del catálogo maestro de la tienda utilizando los diferentes asistentes y cuadernos. También puede utilizar la tabla dinámica Gestión de productos, que le permite actualizar directamente la información de la entrada de catálogo. Puede actualizar el contenido del catálogo o crear nuevos datos de catálogo; por ejemplo, puede:

- Crear, actualizar y eliminar productos y detalles de productos mediante el asistente o el cuaderno. Los productos actúan como plantillas para los códigos de artículo, los elementos individuales que finalmente se venden al cliente. Los detalles del producto incluyen el código del producto (que identifica de forma exclusiva el producto), el nombre del producto y la descripción, las opciones de comercialización (por ejemplo, mostrar un producto a los clientes o indicar que

forma parte de una promoción especial), las imágenes del producto, las especificaciones de impuestos y envío, los descuentos asignados a los productos y la información del fabricante.

- Generar, actualizar y eliminar los códigos de artículo. Los códigos de artículo representan cada uno de los artículos que pueden solicitarse de la mercancía que está en venta. Todos los códigos de artículo que están relacionados con un producto determinado presentan el mismo conjunto de atributos y se distinguen por los valores de sus atributos. Las modificaciones o adiciones que se pueden realizar en los códigos de artículo incluyen la misma información que la de los productos, excepto en lo relacionado con los pedidos.
- Crear, actualizar y eliminar categorías (o grupos de catálogo), que son un grupo de objetos con propiedades similares que se utilizan para organizar los productos o servicios que ofrece la tienda. Puede gestionar la jerarquía de categorías de su catálogo maestro creando, modificando y eliminando categorías y los detalles de las categorías como, por ejemplo, el código, el nombre y la descripción de la categoría, incluida la categoría padre y las imágenes.
- Asociar productos y códigos de artículo con categorías, seleccionando la categoría padre o trasladando los productos y códigos de artículo de una categoría a otra.
- Crear atributos y valores de atributos para los productos. Cada combinación de atributos y valores de atributos posibles es igual a un nuevo código de artículo. Deberá predefinir previamente los valores de los atributos antes de asignarlos a los códigos de artículo. Después de crear atributos y sus valores, puede crear o actualizar información como, por ejemplo, el nombre, la descripción, (el texto, números enteros o números decimales) y la secuencia en la que aparecerán los atributos y los valores de atributos.
- Crear, actualizar, suprimir y asociar la fijación de precios del catálogo con los productos. Puede definir un precio para un producto o código de artículo, en una o varias monedas, junto con un conjunto de condiciones como, por ejemplo, establecer un precio para una cantidad unitaria o para cantidades generales, que deben satisfacerse para poder utilizar el pedido.

Puede consultar el apartado de Gestión de productos de la ayuda en línea si desea obtener instrucciones detalladas acerca de cada tarea.

Notas:

1. Las herramientas de Gestión de productos se recomiendan solamente para cambios ligeros. Para las actualizaciones importantes del catálogo, por ejemplo, para añadir o suprimir los productos de temporada o para preparar una venta de liquidación total, utilice Loader package.
2. Las modificaciones que se realicen en los datos del catálogo no se pueden mostrar en la tienda a menos que inhabilite el almacenamiento en antememoria o suprima las páginas JSP que actualmente hay en la antememoria. Para obtener más información, consulte el mandato CacheDelete en la ayuda en línea de WebSphere Commerce. El mandato CacheDelete inicia la limpieza remota de la antememoria de páginas dinámicas y permite gestionar la antememoria sin que sea necesario acceder directamente al sistema de archivos. Antes de utilizar este mandato compruebe que la invalidación de página automática esté habilitada. Tenga en cuenta que para utilizar este mandato debe haber iniciado la sesión como administrador.

Loader Package

También puede efectuar el mantenimiento del catálogo utilizando Loader Package o Web Editor, dos programas de utilidad que están disponibles en Catalog

Manager. Loader package resulta ideal para importar a la base de datos grandes cantidades de información de productos existentes. En WebSphere Commerce, esta es la herramienta principal para crear y gestionar información sobre catálogos. Este paquete consta principalmente de utilidades de mandatos para preparar y cargar datos en la base de datos de WebSphere Commerce. Loader Package también permite extraer datos de una base de datos como un documento XML, transformar los datos XML en formatos XML alternativos y transformar datos entre un formato de variables delimitadas por caracteres y un formato de datos XML.

Consulte la ayuda en línea a la producción de WebSphere Commerce para obtener más información.



Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce Server. Para obtener más información sobre el modelo de información, consulte “Modelo de información de los datos de la tienda” en la página 105. Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, “Descripción de UML”, en la página 463.

Oferta

Las *ofertas*, o fijación de precios, son precios diferentes del mismo producto o artículo para clientes u organizaciones diferentes. Una oferta representa el precio de una entrada de catálogo y los criterios, como la cantidad que va a comprar, que el cliente debe satisfacer para pagar dicho precio. Por ejemplo, a menudo, la mercancía o los servicios tienen precios distintos para los niños, los estudiantes, los adultos o los jubilados. En WebSphere Commerce, una oferta también se denomina propuesta de comercio y forma parte de un contenedor de propuestas de comercio.

Precio de oferta

El *precio de oferta* es el precio al que se ofrecen las entradas de catálogo de una tienda, mediante contratos o acuerdos comerciales. Una oferta puede tener uno o varios precios de oferta definidos en varias monedas.

Contenedor de propuestas de comercio

Una oferta forma parte de un *contenedor de propuestas de comercio*, que es propiedad de un miembro. Un contenedor de propuestas de comercio contiene las propuestas de comercio. Puede ponerse a disposición de todos los clientes o sólo de clientes de determinados grupos mediante acuerdos comerciales o contratos y los términos y condiciones de los contratos. Bajo un contrato, un contenedor de propuestas de comercio es un objeto de negocio de precio al que pueden hacer referencia varias políticas de negocio de precios y que pueden compartir una o todas las tiendas de un grupo de tiendas. Un contenedor de propuestas de comercio también se denomina *lista de precios*.

Términos y condiciones

Los *términos y condiciones* definen el comportamiento y las propiedades de un acuerdo comercial. Muchos términos y condiciones hacen referencia a políticas de negocio ya que éstas definen varios aspectos del funcionamiento de una tienda.

Tipos de términos y condiciones de fijación de precios

Los *términos y condiciones de fijación de precios* definen qué productos están disponibles bajo un contrato y qué precios pagará el cliente por los productos. En un contrato, es necesario como mínimo uno de los términos de fijación de precios siguientes: Los siguientes términos y condiciones de fijación de precios están disponibles en WebSphere Commerce:

Lista de precios personalizada

Este término especifica que tanto la lista de productos en venta como sus precios están personalizados para la venta en un contrato. Los artículos no están limitados a una sección del catálogo de tienda, pueden pertenecer a cualquier parte del catálogo de tienda.

Catálogo entero con ajuste

Este término ofrece todos los productos disponibles en un catálogo de tienda para la venta con un ajuste por porcentaje (aumento o descuento)

respecto al precio base definido en el catálogo de tienda. Si no se especifica ningún ajuste, los artículos se venden al precio base.

Lista de precios con ajuste

Este término ofrece todos los productos disponibles en una lista de precios para la venta con un ajuste por porcentaje (aumento o descuento) respecto al precio base definido en el catálogo de tienda. Si no se especifica ningún ajuste, los artículos se venden al precio base.

Lista de precios con ajuste selectivo

Este término es similar a la lista de precios con ajuste excepto en que el ajuste no se aplica a toda la lista de precios. El ajuste se realiza en un subconjunto de la lista de precios. El subconjunto de la lista de precios puede ser una política de negocio de conjunto de productos o un conjunto de productos personalizado. Para obtener información sobre las diferencias entre los tipos de conjuntos de productos, consulte el tema "Términos y condiciones de los contratos" en la ayuda en línea de WebSphere Commerce.

Catálogo con filtrado

Este término ofrece todos los productos disponibles en un catálogo de tienda para la venta con un ajuste por porcentaje (aumento o descuento) respecto al precio base definido en el catálogo de tienda. Este término también ofrece todos los productos disponibles en una categoría, o una lista de productos y artículos especificados, para la venta con un ajuste por porcentaje (aumento o descuento) respecto al precio base definido en la lista de precios a la que hace referencia este término. Este término también puede indicar qué categorías, productos y artículos están a la venta y no están a la venta en un contrato. Los conjuntos de productos de categoría se comportarán como una política de negocio de conjunto de productos. Los conjuntos de productos de artículo serán conjuntos de productos personalizados.

Acuerdo comercial

Un *acuerdo comercial* puede ser un contrato, una RFQ, una cuenta de negocio o una subasta. Un acuerdo comercial es un acuerdo que se negocia entre un vendedor y un comprador bajo el cual el comprador puede comprar determinados artículos bajo los términos y condiciones especificados y las políticas de negocio estipuladas en el contrato. Por ejemplo, permite al cliente comprar productos de una tienda al precio especificado durante un periodo de tiempo especificado, bajo los términos y condiciones de fijación de precios. En WebSphere Commerce, todos los clientes deben comprar en una tienda bajo un contrato. Una tienda puede utilizar uno o varios contratos y se puede designar uno de ellos como el contrato por omisión. Un contrato por omisión contiene un conjunto de términos y condiciones asociados a un conjunto de políticas por omisión de una tienda. Un acuerdo comercial puede contener cero o más participantes con roles diferentes.

Participante

Un *participante* puede formar parte de un acuerdo comercial o de términos y condiciones. Un participante es un miembro y puede tratarse de un grupo de miembros, de una organización, etcétera. Si en un contrato se especifica un participante cuyo rol es el de comprador, para poder comprar bajo ese contrato el comprador debe ser miembro del participante comprador. Los términos y condiciones del contrato también pueden contener cero o varios participantes.

Rol de participante

Un participante puede tener uno de los siguientes *roles de participante*:

- Creador
- Vendedor
- Comprador
- Proveedor
- Aprobador
- Titular de la cuenta
- Persona de contacto del comprador
- Persona de contacto del vendedor
- Representante legal
- Administrador.

Contrato

Un *contrato* contiene el precio de oferta del producto. En WebSphere Commerce, todos los clientes deben comprar bajo un contrato. Un contrato permite al cliente adquirir productos de una tienda al precio especificado durante un periodo de tiempo especificado, bajo los términos y condiciones y las políticas de negocio estipulados en el contrato. Una tienda puede poseer varios contratos o no poseer ninguno pero, como mínimo, debe poseer un contrato por omisión.

Política de negocio

Las *políticas de negocio* son conjuntos de normas seguidas por una tienda o grupo de tiendas que definen los procesos de negocio, las prácticas del sector y el ámbito y las características de las ofertas de una tienda o grupo de tiendas. Las políticas de negocio se imponen con una combinación de uno o varios mandatos de política de negocio que implementan las normas de la política de negocio, una referencia a un objeto de negocio sobre el que actúan las normas y un conjunto de propiedades que configuran el funcionamiento de los mandatos de política de negocio.

Política de precios

Una *política de precios* contiene una referencia a una lista de precios y puede estar asociada con varios mandatos de política de negocio que definen cómo se implementarán las políticas de negocio en las listas de precios. La política se puede definir para una tienda o un grupo de tiendas. Si la política se registra para un grupo de tiendas, entonces todas las tiendas de dicho grupo podrán utilizar la política.

Envío de entrada de catálogo

La información de *envío de entrada de catálogo* incluye información acerca de cómo se empaqueta el producto para su envío. Cada entrada de catálogo puede tener definidos diferentes tipos de información de envío. Por ejemplo, la altura, el peso y la longitud del producto una vez empaquetado.

Otros elementos asociados a la fijación de precios

Los elementos siguientes están asociados a la fijación de precios:

- Un *miembro* que es el propietario del contenedor de propuestas de comercio. Un contenedor de propuestas de comercio sólo tiene un propietario.
- Una *entidad de tienda* representa una tienda en la base de datos de WebSphere Commerce Server.

- Un *catálogo* contiene entradas de catálogo a las que se hará referencia en un contrato. El catálogo contiene toda la información jerárquica y de navegación para el catálogo en línea y es una colección de grupos de catálogo y de entradas de catálogo que se pueden visualizar y adquirir en una tienda en línea.
- Un *grupo de catálogo*, o categoría, son agrupaciones genéricas de entradas de catálogo, que se crean para facilitar la navegación y partición del catálogo. Un grupo de catálogo pertenece a un catálogo y puede contener más de un grupo de catálogo o entradas de catálogo. Puede asociar grupos de catálogo a más de un catálogo.
- Una *entrada de catálogo* representa mercancía que se puede solicitar en un catálogo en línea. Las entradas de catálogo pertenecen a grupos de catálogo. Una oferta siempre está asociada a una entrada de catálogo.



Para obtener información más detallada sobre la estructura de los elementos de fijación de precios en WebSphere Commerce Server, consulte los modelos de datos y de objetos de fijación de precios en la ayuda en línea de WebSphere Commerce.

Creación de elementos de fijación de precios en WebSphere Commerce

Tiene dos opciones para crear los elementos de fijación de precios:

- Crear precios utilizando las herramientas de Gestión de productos de WebSphere Commerce Accelerator. El uso de las herramientas de WebSphere Commerce Accelerator es más conveniente en la creación de precios para catálogos muy pequeños.
- Crear precios en un archivo XML, que se puede cargar mediante Loader package de WebSphere Commerce, o como parte de un archivador de tienda, que se puede publicar mediante la Consola de administración. Este método es más adecuado para crear grandes cantidades de datos.

Para obtener más información sobre la creación de precios utilizando las herramientas de Gestión de productos de WebSphere Commerce Accelerator, consulte la ayuda en línea de WebSphere Commerce. Para obtener más información sobre la creación de precios en un archivo XML, consulte “Creación de elementos de fijación de precios en un archivo XML”.

Creación de elementos de fijación de precios en un archivo XML

Cree los elementos de fijación de precios en formato de archivos XML que se puedan cargar en la base de datos utilizando Loader package. Para obtener más información sobre Loader package, consulte la Parte 10, “Publicación de la tienda”, en la página 345.

1. Revise los archivos XML que se utilizan para crear elementos de fijación de precios para las tiendas de ejemplo. Todos los archivos para las tiendas de ejemplo se encuentran en el archivo archivador de tienda correspondiente.

Los archivos de archivador de tienda se encuentran en el directorio siguiente:

- *dir_instalación_WC/samplestores*

Nota: La ayuda en línea de WebSphere Commerce contiene información sobre cada uno de los elementos de datos contenidos en las tiendas de ejemplo.

Cada tienda de ejemplo incluye dos archivos `offering.xml`, que incluyen la información sobre la fijación de precios. Para ver los archivos `offering.xml` en el archivador de tienda, descomprímalo utilizando un programa ZIP. Los archivos `offering.xml` se encuentran en el directorio de datos. El archivo `offering.xml` específico de un idioma está situado en un subdirectorío específico del entorno nacional del directorio de datos.

2. Revise la información del Apéndice B, “Creación de los datos”, en la página 465.
3. Cree un archivo `offering.xml`, copiando uno de los archivos `offering.xml` de los archivadores de tienda de ejemplo o creando uno nuevo. Para obtener más información, consulte el archivo DTD que corresponde a `offering.xml`. Los archivos DTD se encuentran en el directorio siguiente:
 - `dir_instalación_WC/xml/sar`
4. Cree un contenedor de propuestas de comercio. Para ofrecer los precios de los artículos de su tienda, primero debe crear un contenedor de propuestas de comercio. Para crearlo, añada información a la tabla TRADEPOSCN.
 - a. Utilice el ejemplo siguiente como guía y cree un contenedor de propuestas de comercio en su archivo XML, en la tabla TRADEPOSCN:

```
<tradeposcn
tradeposcn_id="@tradeposcn_id_101"
member_id="@seller_b2b_mbr_id"
markfordelete="0"
name="ToolTech"
precedence="0"

/>
```

donde

- `tradeposcn_id` es una clave exclusiva generada
 - `@seller_b2b_mbr_id` es el propietario contenedor de propuestas de comercio. En la tienda de ejemplo FashionFlow, sustituya esto por `@Member_ID`;
 - `markfordelete` indica lo siguiente:
 - 0 = puede utilizarse el contenedor de propuestas de comercio
 - 1 = el contenedor de propuestas de comercio se ha marcado para suprimirlo (consulte el programa de utilidad DBClean) y no se debe utilizar
 - `name` es un nombre nemotécnico para el contenedor de propuestas de comercio, exclusivo de un propietario particular.
 - `precedence` se utiliza cuando se habilita más de un contenedor de propuestas de comercio en un momento determinado; en ese caso, se utiliza la propuesta cuyo valor de PRECEDENCE sea mayor.
5. Asocie el catálogo maestro a un contenedor de propuestas de comercio, añadiendo información a la tabla CATGRPTPC. Cuando asocia el catálogo maestro a un contenedor de propuestas de comercio, todas las entradas del catálogo maestro deben tener un precio estándar. Para obtener más información sobre la creación de catálogos maestros, consulte “Visualización de los elementos de catálogo de la tienda” en la página 177.
 - a. Utilizando el ejemplo siguiente como muestra, asocie el catálogo maestro al contenedor de propuestas de comercio añadiendo información a la tabla CATGRPTPC:

```
<catgrptpc
catalog_id="@catalog_id_1"
tradeposcn_id="@tradeposcn_id_101"
/>
```

donde

- catalog_id es el catálogo maestro.
- tradeposcn_id es el contenedor de propuestas de comercio.

6. Cree ofertas y precios de oferta para las entradas de catálogo añadiendo información en las tablas OFFER y OFFERPRICE

- a. Utilizando el ejemplo siguiente como muestra, cree una oferta para una entrada de catálogo añadiendo información a la tabla OFFER. Recuerde que para crear precios, debe haber creado entradas de catálogo. Para obtener más información sobre la creación de entradas de catálogo, consulte “Visualización de los elementos de catálogo de la tienda” en la página 177.

```
offer
offer_id="@offer_id_138"
startdate="2000-06-19 00:00:00.000000"
catentry_id="@product_id_102"
precedence="0"
published="1"
identifier="1"
flags="1"
tradeposcn_id="@tradeposcn_id_101"
/>
```

donde

- offer_id es una clave exclusiva generada.
 - startdate es el inicio del rango de fecha y hora durante el cual estará en vigor esta oferta.
 - catentry_id es la entrada de catálogo que se ofrece a la venta.
 - precedence: cuando hay más de una oferta en vigor en un momento determinado, se utiliza la que tiene el valor de PRECEDENCE más elevado.
 - published puede ser
 - 0 = no publicado (inhabilitado temporalmente)
 - 1 = publicado
 - 2 = marcado para suprimirlo (y no publicado).
 - identifier es un número que identifica de forma exclusiva a esta oferta junto con su entrada de catálogo especificada y contenedor de propuestas de comercio.
 - flags son
 - 1 = shiptoAddressRequired - si es 1, OrderPrepare devolverá un error si un artículo de pedido hace referencia a esta oferta pero no tiene una dirección de envío.
 - tradeposcn_id es el contenedor de propuestas de comercio del que forma parte esta oferta.
- b. Utilizando el ejemplo siguiente como muestra, cree un precio de oferta para una entrada de catálogo añadiendo información a la tabla OFFERPRICE. El precio de oferta es el precio real al que una entrada de catálogo se pone a la venta. Recuerde que para crear precios, debe haber creado entradas de catálogo. Para obtener más información sobre la creación de entradas de catálogo, consulte “Visualización de los elementos de catálogo de la tienda” en la página 177.

```
<offerprice
offer_id="@offer_id_138"
currency="EUR"
price="590.00"
/>
```

donde

- offer_id es la oferta asociada a este precio.
- currency es la moneda en que se ofrece el precio.
- price es el precio para la cantidad nominal (vea CATENTSHIP.NOMINALQUANTITY) del producto al que hace referencia la oferta.

Nota: Para mostrar varias monedas en su tienda, cree una entrada XML aparte en la tabla OFFERPRICE para cada moneda. Por ejemplo, para mostrar la moneda en dólares canadienses, utilice currency="CAD" en una entrada XML nueva. El valor de price cambiará para reflejar el precio en dólares canadienses. O puede utilizar una conversión, que permite a los clientes ver diferentes tipos de conversión según la moneda que seleccionen. Para obtener más información, consulte "Creación de elementos de moneda utilizando un archivo XML" en la página 238.

c. Repita los pasos a y b para todas las entradas de su catálogo.



Para obtener más información sobre cómo utilizar @ y &, consulte el Apéndice B, "Creación de los datos", en la página 465.

Capítulo 18. Elementos de contrato

En WebSphere Commerce, todos los clientes deben comprar bajo un contrato. Un contrato permite a los clientes adquirir productos de una tienda a un precio especificado durante un periodo de tiempo especificado, bajo condiciones específicas. Al examinar el catálogo de una tienda, los clientes sólo verán los productos cubiertos por los contratos que estén autorizados a utilizar en la tienda.

Si desea que los clientes que no tienen ningún contrato con la tienda (por ejemplo, los compradores invitados) también puedan comprar en la tienda, o si desea que los clientes puedan comprar productos que no estén cubiertos por sus contratos, su tienda necesitará un *contrato por omisión*.

Importante

WebSphere Commerce Professional Edition solamente da soporte al contrato por omisión de tienda.

Los contratos que no sean el contrato por omisión de tienda solamente están soportados por WebSphere Commerce Business Edition.

Para que todos los clientes puedan comprar en una tienda creada con WebSphere Commerce, ésta debe incluir lo siguiente:

- Políticas de negocio
- Contrato por omisión

El contrato por omisión hace referencia a las políticas de negocio, permitiendo que todos los clientes puedan comprar en una tienda.

Importante: Las cuentas de negocio sólo están soportadas por WebSphere Commerce Business Edition.

Contratos

Hay dos tipos de contratos activos asociados a las tiendas: los contratos desplegados y los contratos por omisión. Los contratos desplegados autorizan a organizaciones compradoras específicas o compradores individuales y se pueden crear utilizando WebSphere Commerce Accelerator después de crear la tienda. Un contrato desplegado se asocia a una cuenta de negocio. Un contrato por omisión define el comportamiento por omisión de la tienda para los compradores que no tienen ningún otro contrato con la tienda. Un contrato por omisión solamente puede crearse utilizando XML y sólo puede definirse un contrato por omisión para una tienda. Para obtener más información sobre los contratos, consulte la ayuda en línea a la producción y el desarrollo de WebSphere Commerce. Para obtener información sobre la creación de un elemento de contrato por omisión, consulte el apartado “Creación de un elemento de contrato por omisión en WebSphere Commerce” en la página 203.

Un contrato típico consta de los elementos siguientes:

Perfil El perfil del contrato contiene la información de identificación para el contrato. Esta información incluye un nombre exclusivo para el contrato, una descripción breve y un periodo de tiempo durante el cual el contrato es válido.

Participantes

Los participantes del contrato son las organizaciones que toman parte en el contrato. Hay una organización compradora, una organización vendedora y personas de contacto en ambas organizaciones.

Términos y condiciones

Los términos y condiciones del contrato son las normas que cubren la implementación real del contrato. Los términos y condiciones del contrato cubren información tal como la fijación de precios de productos, las devoluciones y los reembolsos, el pago, el envío y la aprobación de pedidos.

Adjuntos

Los adjuntos del contrato cubren cualquier información no cubierta por los elementos anteriores, por ejemplo archivos adjuntos que proporcionan información adicional sobre el contrato y cualquier observación general acerca del contrato. WebSphere Commerce almacena los URI (Universal Resource Identifier) para los adjuntos de contratos y no los adjuntos reales.

Referencia

Un contrato puede hacer referencia a otro contrato para compartir sus términos y condiciones. Por ejemplo, el contrato A puede hacer referencia al contrato B. De este modo, un comprador que esté autorizado para el contrato A estará autorizado a utilizar todos los términos y condiciones del contrato A, así como todos los términos y condiciones del contrato B.

Acuerdos comerciales

Un contrato es un tipo de acuerdo comercial. WebSphere Commerce proporciona varios mecanismos de comercio que rigen las interacciones entre compradores y vendedores. Las diferentes ediciones de WebSphere Commerce dan soporte a los mecanismos de comercio siguientes:

- Subastas (soportadas tanto por Business Edition como por Professional Edition)

- Cuentas de negocio (soportadas solamente por Business Edition)
- Contratos (vea las limitaciones descritas anteriormente en este capítulo)
- Solicitud de presupuesto (RFQ) (soportada solamente por Business Edition)

Todos estos mecanismos de comercio tienen propiedades comunes. Por ejemplo, todos los mecanismos de comercio tienen participantes y todos tienen normas que rigen el comportamiento del mecanismo de comercio. Las normas que rigen el comportamiento de los mecanismos de comercio se conocen como *términos y condiciones* en WebSphere Commerce.

Un acuerdo comercial representa una instancia de un mecanismo de comercio y registra las propiedades de esa instancia de un mecanismo de comercio. Cada contrato, cuenta de negocio y RFQ en WebSphere Commerce es un acuerdo comercial. Hay un único acuerdo comercial que rige todas las subastas en WebSphere Commerce.

Un acuerdo comercial consta de un perfil almacenado en la tabla TRADING; de participantes almacenados en la tabla PARTICIPANT; de términos y condiciones almacenados en la tabla TERMCOND; y de adjuntos opcionales almacenados como URI (Identificador de recursos universal) en la tabla ATTACHMENT. Puesto que un acuerdo comercial puede tener múltiples adjuntos, éstos se relacionan con el acuerdo comercial a través de la tabla TRDATTACH. Tenga en cuenta que las RFQ no dan soporte a los adjuntos.

Además del acuerdo comercial general, cada tipo de acuerdo comercial almacena información adicional específica del tipo de acuerdo comercial en su propia tabla: CONTRACT almacena información específica de contratos; RFQ almacena información específica de RFQ; y ACCOUNT almacena información específica de cuentas.

Participantes

Los participantes del contrato adoptan roles específicos en cada contrato. Los participantes pueden ser una persona de contacto de una organización compradora y de una organización vendedora. Si un contrato especifica que el participante comprador sea nulo, entonces todos los usuarios, incluidos los invitados, tienen derecho al contrato. Cualquier contrato puede especificar un participante comprador que sea nulo.

Términos y condiciones

Los términos y las condiciones definen el comportamiento y las propiedades de un acuerdo comercial. En los contratos, los términos y condiciones definen cómo se implementa un contrato para una organización compradora. Definen lo que se está vendiendo bajo el contrato, el precio de los artículos en venta, cómo se envían los artículos, cómo se pagan los pedidos, cómo se manejan las devoluciones, cómo se aprueban los pedidos y desde dónde se envían los pedidos.

Algunos términos y condiciones hacen referencia a políticas de negocio ya que éstas definen muchos aspectos del funcionamiento de una tienda. Los términos y condiciones proporcionan parámetros para las políticas de negocio a las que hacen referencia. El suministro de parámetros a las políticas de negocio le permite modificar el comportamiento de esas políticas para cada contrato. WebSphere Commerce da soporte a los términos y condiciones siguientes (los términos y condiciones que hacen referencia a las políticas de negocio están marcados con un asterisco (*)).

Centro de despacho de pedidos

Este término opcional le permite especificar la lista de centros de despacho de pedidos desde los que se deben despachar los pedidos formalizados bajo el contrato. Esta lista debe ser un subconjunto de los centros de despacho de pedidos definidos para la tienda. La prioridad de los centros de despacho de pedidos la define la tienda y los términos y condiciones de un contrato no la pueden alterar temporalmente.

Aprobación de pedidos

Este término especifica si, antes de despachar los pedidos, la organización cliente debe aprobarlos. Puede especificar un importe opcional, que incluya impuestos y costes de envío, que permita que los pedidos con un valor inferior al importe se despachen sin la aprobación de la organización cliente. Si el total de un pedido supera este importe, se necesita la aprobación. Si un comprador está formalizando un pedido con artículos de pedido que están bajo varios contratos y un artículo del pedido tiene un contrato que especifica este término, el pedido entero estará sujeto al término de aprobación de pedidos que se aplica al artículo.

Método de pago*

Este término opcional especifica los métodos de pago que se aceptarán para los pedidos realizados bajo el contrato. El método de pago puede ser tan general como un tipo de pago, por ejemplo un tipo de tarjeta de crédito, o tan específico como un número de tarjeta de crédito que se deberá utilizar para el pago. Si no se especifica ningún término de método de pago en un contrato, se aceptará el pago en todos los métodos aceptados por la tienda para los pedidos realizados bajo el contrato.

Términos y condiciones de fijación de precios

Los términos y condiciones de fijación de precios definen qué productos están disponibles bajo un contrato y qué precios pagará el cliente por los productos. En un contrato, se necesita como mínimo un término de fijación de precios. En WebSphere Commerce están disponibles los términos y condiciones de fijación de precios siguientes:

Lista de precios personalizada

Este término especifica que tanto la lista de productos en venta como sus precios están personalizados para la venta en un contrato. Los artículos no están limitados a una sección del catálogo de tienda, pueden pertenecer a cualquier parte del catálogo de tienda.

Catálogo entero con ajuste

Este término ofrece todos los productos disponibles en un catálogo de tienda para la venta con un ajuste por porcentaje (aumento o descuento) respecto al precio base definido en el catálogo de tienda. Si no se especifica ningún ajuste, los artículos se venden al precio base.

Lista de precios con ajuste*

Este término ofrece todos los productos disponibles en una lista de precios para la venta con un ajuste por porcentaje (aumento o descuento) respecto al precio base definido en el catálogo de tienda. Si no se especifica ningún ajuste, los artículos se venden al precio base.

Lista de precios con ajuste selectivo*

Este término es similar a la lista de precios con ajuste excepto en que el ajuste no se aplica a toda la lista de precios. El ajuste se

realiza en un subconjunto de la lista de precios. El subconjunto de la lista de precios puede ser una política de negocio de conjunto de productos o un conjunto de productos personalizado. Para obtener información sobre las diferencias entre los tipos de conjuntos de productos, consulte la ayuda en línea al desarrollo de WebSphere Commerce.

Catálogo con filtrado *

Este término ofrece todos los productos disponibles en un catálogo de tienda para la venta con un ajuste por porcentaje (aumento o descuento) respecto al precio base definido en el catálogo de tienda. Este término también ofrece todos los productos disponibles en una categoría, o una lista de productos y artículos especificados, para la venta con un ajuste por porcentaje (aumento o descuento) respecto al precio base definido en la lista de precios a la que hace referencia este término. Este término también puede indicar qué categorías, productos y artículos están a la venta y no están a la venta en un contrato. Los conjuntos de productos de categoría se comportarán como una política de negocio de conjunto de productos. Los conjuntos de productos de artículo serán conjuntos de productos personalizados.

Términos y condiciones de restricción de productos

Los términos y condiciones de restricción de productos controlan qué productos se incluyen o excluyen para la venta bajo un contrato. Los términos de restricción de productos son opcionales. Si no se especifican términos y condiciones de restricción de productos en un contrato, todos los productos especificados en los términos y condiciones de precio del contrato están disponibles para la venta bajo el contrato. En WebSphere Commerce están disponibles los siguientes términos y condiciones de restricción de productos:

Exclusión de conjunto de productos personalizado

Este término indica que los artículos de un conjunto de productos personalizado no están a la venta en un contrato.

Inclusión de conjunto de productos personalizado

Este término indica que los artículos de un conjunto de productos personalizado están a la venta en un contrato.

Exclusión de conjunto de productos*

Este término indica que los artículos de una política de negocio de conjunto de productos no están a la venta en un contrato.

Inclusión de conjunto de productos*

Este término indica que los artículos de una política de negocio de conjunto de productos están a la venta en un contrato.

Los términos de exclusión tienen prioridad sobre los términos de inclusión. Esto significa que si un producto aparece en un término de inclusión y en un término de exclusión del contrato, no se podrá comprar el producto bajo el contrato. Para obtener información sobre las diferencias entre un conjunto de productos personalizado y una política de negocio de conjunto de productos, consulte la ayuda en línea al desarrollo de WebSphere Commerce.

Términos y condiciones de devoluciones

Los términos y condiciones de devoluciones especifican cómo se manejan las devoluciones bajo este contrato. Si no se especifican términos y

condiciones de devoluciones, no se pueden crear devoluciones. Si se especifican, los términos y condiciones de devoluciones deben ser sólo un conjunto que se aplique al contrato entero. En WebSphere Commerce están disponibles los términos y condiciones de devoluciones siguientes:

Método de pago de reembolso*

Este término especifica el método de pago utilizado para pagar los reembolsos a un cliente. Si se especifica un término de cargo por devolución, deberá especificarse también un método de pago de reembolso como mínimo. Puede que no se especifique este término si no se permiten devoluciones bajo el contrato.

Cargo por devolución*

Este término especifica cómo se aprueban automáticamente las devoluciones y las deducciones del reembolso que se realizan por el manejo de la devolución, por ejemplo, los cargos por realmacenaje.

Importe de derecho de compra

Este término pone un límite en el valor combinado de todos los pedidos, incluidos los impuestos y el envío, formalizados bajo un contrato. El valor de todos los pedidos realizados bajo el contrato debe ser inferior o igual a un importe especificado. Si se excede este límite al formalizar un pedido, fallará la autorización de pago del pedido.

Términos y condiciones de envío

Los términos y condiciones de envío especifican cómo se enviarán los pedidos, a dónde se enviarán y quién pagará el envío. En WebSphere Commerce están disponibles los términos y condiciones de envío siguientes:

Modalidad de envío*

Este término opcional define cómo se envían los pedidos creados bajo un contrato. Si no se especifica este término en un contrato, los pedidos pueden enviarse mediante cualquier modalidad disponible en una tienda. Una modalidad de envío se conoce también como proveedor de envío. Un proveedor de envío es la combinación de una empresa de transportes y su servicio de envío. Por ejemplo, Mensajería XYZ, Urgente es un proveedor de envío.

Dirección de envío

Este término opcional especifica a dónde se envían los productos comprados bajo un contrato. La especificación de este término y condición le permite limitar las ubicaciones a las que se pueden enviar pedidos. Si no se especifica el término y condición de dirección de envío, se deberá especificar una dirección de envío cada vez que se realice un pedido bajo un contrato. Si se especifica este término, el comprador no puede especificar una dirección de envío nueva al formalizar un pedido, sino que debe seleccionar una dirección de envío en una lista de direcciones de envío.

Tipo de costes de envío*

Este término define quién paga por el envío de pedidos. Se soportan los siguientes tipos de costes de envío:

- El comprador paga los costes de envío al vendedor. El vendedor calcula los costes de envío cuando se captura el pedido y los costes de envío pasan a formar parte del total del pedido.
- El comprador paga los costes de envío a la empresa de transportes. La empresa de transportes calcula los costes de

envío y asume la responsabilidad de cobrar el pago al comprador. Los costes de envío no se calculan cuando se captura el pedido.

Interfaz de referencia*

Este término especifica la relación entre una tienda y una tienda remota. Define las funciones soportadas por la tienda remota y los parámetros que se deben utilizar en los mensajes enviados a la tienda remota.

Políticas de negocio

Las políticas de negocio son conjuntos de normas seguidas por una tienda o un grupo de tiendas. Las políticas de negocio definen los procesos de negocio, las prácticas del sector y el ámbito y las características de las ofertas de una tienda o grupo de tiendas. Son la plantilla de origen y referencia central para todas las prácticas permitidas y soportadas en una tienda o un grupo de tiendas.

En WebSphere Commerce, las políticas de negocio se imponen con una combinación de uno o más mandatos de política de negocio que implementan las normas de la política de negocio, una referencia a un objeto de negocio sobre el que actúan las normas y un conjunto de propiedades que configuran el funcionamiento de los mandatos de política de negocio. Los términos y condiciones pueden proporcionar parámetros para las políticas de negocio a las que hacen referencia. Esto permite modificar el comportamiento de la política de negocio según el término y condición que haga referencia a la política de negocio.

Las políticas de negocio son un recurso que se puede compartir. Cuando se listan las políticas de negocio que se pueden utilizar en un contrato, se listan las políticas de negocio propiedad de la tienda en la que se está creando el contrato y las políticas de negocio propiedad de cualquier tienda con la que haya una relación entre tiendas de tipo `com.ibm.commerce.businessPolicy`. Para obtener más información acerca de cómo se comparten los elementos entre las tiendas de un sitio, consulte el Capítulo 14, "Relaciones entre tiendas", en la página 143.

En WebSphere Commerce se proporcionan las siguientes categorías de políticas de negocio:

Políticas de negocio de catálogo

Las políticas de negocio de catálogo definen el ámbito y las características del catálogo de productos a la venta en una tienda, incluidos los precios y las categorías de productos del catálogo de una tienda.

Políticas de negocio de pago

Las políticas de negocio de facturación, pago y reembolso definen el modo en que una tienda acepta los pagos, paga los reembolsos y el formato de las facturas de la tienda.

Políticas de negocio de devoluciones

Las políticas de negocio de devoluciones definen si se aceptan devoluciones, el periodo de tiempo durante el cual se aceptan y los cargos por realmacenaje, si los hay, que se aplican a las devoluciones.

Políticas de negocio de envío

Las políticas de negocio de envío definen los proveedores de envío que puede utilizar una tienda y los costes asociados con cada tipo.

Políticas de negocio de interfaz de referencia

Las políticas de negocio de interfaz de referencia definen la relación entre una tienda proxy y una tienda remota.

Muchos términos y condiciones de contrato hacen referencia a políticas de negocio. Esto proporciona una medida de control sobre la naturaleza de los contratos en los que participa una tienda y al mismo tiempo sigue proporcionando flexibilidad en la creación de términos y condiciones de contrato. Para obtener más información sobre las políticas de negocio, consulte la publicación *WebSphere Commerce, Guías de programación y aprendizaje*.

Adjunto

Un adjunto proporciona información adicional acerca de un acuerdo comercial que no queda cubierta por los demás elementos del acuerdo comercial. Un ejemplo es un archivo que proporciona información adicional acerca de las requisitos de RFQ y cualquier observación general acerca de la RFQ. Un acuerdo comercial puede tener varios adjuntos. Los adjuntos se almacenan fuera de WebSphere Commerce y el acuerdo comercial almacena los URI (Universal Resource Identifiers) asociados a los adjuntos. A continuación se muestran algunos ejemplos de URI:

- <http://www.miempresa.com/information/documento1.txt>
- <archivo:///home/usuariojoan/misdocs/documento1>
- <ftp://ftp.miempresa.com/information/adjunto.txt>

Se puede asignar un uso a todos los adjuntos que indique la finalidad del adjunto. El uso de adjunto es una propiedad opcional de un adjunto.

Artículo de pedido

Un artículo de pedido es un producto o artículo que se incluye en un pedido. En un mismo pedido, se pueden comprar diferentes artículos de pedido bajo distintos acuerdos comerciales de contrato. En función del diseño de la tienda, los compradores pueden seleccionar el acuerdo comercial de contrato bajo el que comprarán, al principio del flujo de compra o cuando añaden un artículo al pedido. Cuando se compran artículos bajo acuerdos comerciales de contrato diferentes, se aplican las normas siguientes:

- Los acuerdos comerciales de contrato para todos los artículos de un pedido deben tener en común, como mínimo, un método de pago. Si el contrato para un artículo no tiene en común un método de pago, el comprador no puede añadir dicho artículo al pedido. Para pagar el pedido sólo se pueden utilizar los métodos de pago comunes a todos los artículos de un pedido.
- Todos los artículos de un pedido deben proceder de acuerdos comerciales de contrato que pertenezcan a la misma cuenta de negocio o al contrato por omisión de la tienda.



Para obtener información más detallada sobre la estructura de los elementos de contrato en WebSphere Commerce, consulte el modelo de datos de contrato en la ayuda en línea al desarrollo de WebSphere Commerce.

Creación de un elemento de contrato por omisión en WebSphere Commerce

El contrato por omisión define el comportamiento por omisión de una tienda. Al igual que con todos los contratos, puede establecer los productos disponibles, los métodos de pago, los métodos de envío y otro tipo de comportamiento de la tienda.

Los contratos por omisión de tienda que se proporcionan con las tiendas de ejemplo de WebSphere Commerce contienen términos y condiciones que especifican lo siguiente:

- Los clientes pueden comprar todos los productos disponibles en el catálogo maestro de la tienda a los precios estándar establecidos en dicho catálogo (sin descuentos ni recargos).
- Los costes de envío se deben pagar al vendedor (la tienda).
- Los clientes pueden devolver los artículos comprados sin recargos de penalización dentro de un periodo de tiempo especificado.
- Los clientes pueden recibir reembolsos utilizando el mismo método de pago usado para la compra original.

Además, la versión más general del contrato por omisión de una tienda omite los términos y condiciones que limitan los métodos de pago y envío que pueden utilizar los compradores. Al omitir estos términos los compradores pueden pagar sus compras utilizando cualquiera de los métodos de pago por omisión soportados por la tienda y utilizar cualquiera de los métodos de envío que haya disponibles en la tienda.

Las propiedades del contrato por omisión se definen en sus términos y condiciones. Algunos de los términos y condiciones hacen referencia a políticas de negocio. Para obtener más información sobre las políticas de negocio y los términos y condiciones, consulte la ayuda en línea al desarrollo de WebSphere Commerce.

Para crear un elemento de contrato por omisión, efectúe lo siguiente:

1. Revise la información en línea sobre términos y condiciones, contratos, contratos por omisión y políticas de negocio.
2. Revise las políticas de negocio definidas en el archivo `wcs.bootstrap.xml`. Para obtener información sobre el archivo `wcs.bootstrap.xml`, consulte la información en línea.
3. Revise los archivos utilizados para crear elementos de contrato por omisión para las tiendas de ejemplo. Todos los archivos de las tiendas de ejemplo se encuentran en el archivador de tienda correspondiente. Cada tienda de ejemplo incluye un archivo `businesspolicy.xml` y un archivo `contract.xml`, que incluyen información adicional sobre políticas de negocio e información sobre el contrato por omisión. Los archivos de archivador de tienda se encuentran en el directorio `dir_instalación_WC/samplestores`.

Notas:

- a. La ayuda en línea de WebSphere Commerce contiene información sobre cada uno de los elementos de datos contenidos en las tiendas de ejemplo.
 - b. Para ver los archivos `businesspolicy.xml` y `contract.xml` del archivador de tienda, descomprímalos utilizando un programa ZIP. Los archivos están situados en el directorio de datos.
 - c. Los archivos de elementos de contrato de la tienda de ejemplo ToolTech que se proporciona con WebSphere Commerce Business Edition incluyen información para contratos que no son el contrato por omisión de la tienda.
4. Revise la información del Apéndice B, “Creación de los datos”, en la página 465.
 5. Cree un archivo `businesspolicy.xml`, copiando uno de los archivos `businesspolicy.xml` de los archivadores de tienda de ejemplo o bien creando un archivo nuevo. Encontrará las instrucciones sobre cómo crear un archivo nuevo en el apartado “Creación de archivos XML de políticas de negocio” en la página 205

página 205. Si desea crear diferentes políticas de negocio a partir de las descritas, consulte el archivo DTD correspondiente a `businesspolicy.xml`. Los archivos DTD están situados en el directorio `dir_instalación_WC/xml/sar`.

6. Cargue el archivo `businesspolicy.xml` utilizando `Loader package`. Para obtener más información sobre `Loader package`, consulte la Parte 10, "Publicación de la tienda", en la página 345. Si crea una tienda multicultural, tal vez desee crear archivos XML independientes para cada entorno nacional al que la tienda dé soporte. El archivo específico del entorno nacional debe especificar toda la información de las descripciones, para que se pueda traducir fácilmente.
7. Cree un archivo `contract.xml`, copiando uno de los archivos `contract.xml` de los archivadores de tienda de ejemplo o bien creando un archivo nuevo. Encontrará las instrucciones sobre cómo crear un archivo nuevo en el apartado "Creación de un archivo de contrato por omisión" en la página 206. Si desea crear un contrato por omisión más complejo, revise el archivo `B2BTrading.dtd` o `Package.xsd` que define la estructura de un archivo de contrato. El archivo `B2BTrading.dtd` lo puede encontrar en el directorio `dir_instalación_WC/xml/trading/dtd`. El archivo `Package.xsd` lo encontrará en el directorio `dir_instalación_WC/xml/trading/xsd`.
8. Publique el contrato utilizando el mandato `ContractImportApprovedVersion`. Para obtener más información, consulte el Capítulo 39, "Publicación de cuentas de negocio y contratos", en la página 419. También puede encontrar información sobre el mandato `ContractImportApprovedVersion` en la ayuda en línea de WebSphere Commerce.

Los usuarios de WebSphere Commerce Business Edition pueden definir contratos para clientes específicos utilizando WebSphere Commerce Accelerator. Para obtener más información acerca de cómo crear contratos para clientes específicos, consulte la ayuda en línea a la producción de WebSphere Commerce.

Creación de archivos XML de políticas de negocio

Aunque WebSphere Commerce proporciona varias políticas de negocio a las que pueden hacer referencia los términos y condiciones del contrato por omisión de la tienda, deberá definir, de todos modos, algunas políticas de negocio. Debe definir las políticas de negocio de cargo por devolución, de aprobación de devoluciones y de fijación de precios a las que hagan referencia los términos del contrato por omisión de la tienda. Se proporcionan los mandatos para estas políticas de negocio y se pueden utilizar sin tener que modificarlos. Si desea crear sus propias políticas de negocio, consulte la publicación *WebSphere Commerce, Guías de programación y aprendizaje*.

Para crear políticas de negocio para la tienda, debe crear una política de negocio y asociar uno o más mandatos a la política de negocio. Para crear una política de negocio, añada información a la tabla `POLICY`. Para asociar un mandato a una política de negocio, añada información a la tabla `POLICYCMD`.

Para crear una política de negocio y asociar mandatos a ella, haga lo siguiente:

1. Cree una política de negocio en el archivo XML de políticas de negocio, añadiendo información a la tabla `POLICY`. Utilice el ejemplo siguiente como muestra:

```
<policy
policy_id="@policy_id_10"
policyname="MasterCatalogPriceList"
```

```
policytype_id="Price"
storeent_id="@storeent_id_1"
properties="name=&STORE_IDENTIFIER;&orgentity_dn=ORGANIZATION_DN
/>
```

donde

- `policy_id` es el identificador numérico y exclusivo de la política de negocio.
- `policyname` es un nombre exclusivo de la política de negocio.
- `policytype_id` es el tipo de política que se está definiendo. Los valores de `policytype_id` válidos son:
 - InvoiceFormat
 - Payment
 - Price
 - ProductSet
 - ReturnApproval
 - ReturnCharge
 - ReturnPayment
 - ShippingCharge
 - ShippingPayment
 - ReferralInterface
- `storeent_id` es la tienda o grupo de tiendas.
- `properties` es una lista de parejas de nombre/valor que se envía al mandato de política de negocio.

2. Asocie un mandato a la política de negocio en el archivo XML de políticas de negocio, añadiendo información a la tabla POLICYCMD. Utilice el ejemplo siguiente como muestra:

```
<polycmd
policy_id="@policy_id_10"
businesscmdclass=
"com.ibm.com.commerce.price.commands.RetrievePricesCmdImpl"
/>
```

donde

- `policy_id` es el identificador numérico de la política de negocio con la que se asocia el mandato.
- `businesscmdclass` es el nombre de la clase Java que implementa la política de negocio.

Las particiones de línea en el atributo `businesscmdclass` sólo se muestran a efectos de presentación.



Para obtener más información sobre cómo utilizar @ y &, consulte el Apéndice B, "Creación de los datos", en la página 465.

Creación de un archivo de contrato por omisión

Para crear un contrato por omisión, debe definir el contrato, el propietario del contrato, la descripción del contrato, los participantes del contrato y los términos y condiciones del contrato. La información sobre un contrato se almacena en cuatro tablas: CONTRACT, PARTICIPNT, TRADING y TERMCOND.

El contrato por omisión se asocia a la tienda mediante la tabla de base de datos STOREDEF. Para los usuarios de WebSphere Commerce Business Edition, los contratos distintos al contrato por omisión se asocian a una tienda mediante la tabla de base de datos STORECNTR.

Puede crear un contrato por omisión en XML, basándose en dos formatos: XSD o DTD. Consulte los apartados siguientes para obtener información detallada acerca de cómo crear cada uno de los tipos.

Creación de un archivo de contrato por omisión en XSD

Para crear un contrato por omisión en formato XSD, efectúe lo siguiente:

1. Defina el contrato por omisión en el archivo XML. El contrato por omisión se define al principio del archivo XML, tal como se muestra a continuación:

```
<?xml version="1.0" encoding="UTF-8"?>
<Package xmlns="http://www.ibm.com/WebSphereCommerce"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ibm.com/WebSphereCommerce Package.xsd">
<BuyerContract state="Active" contractUsage="Default" comment="">
<ContractUniqueKey majorVersionNumber="1" minorVersionNumber="0"
  name="&STORE_IDENTIFIER; Default Contract" origin="Manual">
<ContractOwner>
<OrganizationRef distinguishName="ou=&ORGENTITYNAME; ,&ORGANIZATION_DN;" />
</ContractOwner>
</ContractUniqueKey>
```

Tenga en cuenta que las particiones de línea de los elementos Package y ContractUniqueKey solo aparecen a efectos de presentación.

2. Defina los participantes del contrato en el archivo XML de contrato. Utilice el ejemplo siguiente como muestra:

```
<Participant role="Buyer">
</Participant>
<Participant role="Seller">
  <ParticipantMember>
    <OrganizationRef distinguishName=""ou=&ORGENTITYNAME; ,&ORGANIZATION_DN;" />
  </ParticipantMember>
</Participant>
```

donde distinguishName es el nombre del usuario que es el vendedor para este contrato con el formato de nombre distinguido LDAP. Por ejemplo, uid=johnsmith,ou=People,o=ibm,o=com. En muchos casos, será igual que el propietario del contrato.

Nota: No se especifican miembros en el rol de participante de comprador porque el contrato está disponible para todos los usuarios con el rol de comprador.

3. Defina la descripción del contrato en el archivo XML de contrato. Utilice el ejemplo siguiente como muestra:

```
<ContractDescription title="Contrato por omisión de la tienda." locale="es_ES"/>
```

donde

- title es una descripción de texto del contrato.
- locale es el entorno nacional del idioma de title. Para locale se han definido previamente los valores siguientes:
 - en_US (Inglés de EE.UU.)
 - fr_FR (Francés)
 - de_DE (Alemán)

- it_IT (Italiano)
- es_ES (Español)
- pt_BR (Portugués de Brasil)
- zh_CN (Chino simplificado)
- zh_TW (Chino tradicional)
- ko_KR (Coreano)
- ja_JP (Japonés)

Se pueden definir valores adicionales para locale actualizando los elementos de idioma de la tienda. Para obtener más información sobre los elementos de idioma, consulte el Capítulo 22, "Elementos de idioma", en la página 233.

4. Defina los términos y condiciones en el archivo XML de contrato. Los elementos XML y los atributos son distintos para los diversos tipos de términos y condiciones. Utilice el archivo Package.xsd para saber qué elementos XML y atributos se han de utilizar para cada tipo de término. Al definir términos y condiciones, comúnmente se utilizan los atributos siguientes:

policyName

El nombre de la política de negocio a la que hace referencia el término y condición. El nombre se almacena en POLICY.POLICYNAME.

Referencias de políticas

El tipo de política de negocio al que hace referencia el término y condición. Los valores válidos son:

- PricePolicyRef
- ProductSetPolicyRef
- InvoiceFormatPolicyRef
- PaymentPolicyRef
- ReturnApprovalPolicyRef
- ReturnChargePolicyRef
- ReturnPaymentPolicyRef
- ShippingChargePolicyRef
- ShippingModePolicyRef

storeRef

La tienda o grupo de tiendas para el término y condición.

distinguishName

El nombre del usuario que es el propietario de la tienda o grupo de tiendas. El nombre debe tener el formato de nombre distinguido LDAP. Por ejemplo, uid=wcsadmin,o=Root Organization.

Los siguientes términos y condiciones de ejemplo van precedidos de una descripción de lo que definen:

- Todos los compradores pueden adquirir todos los artículos del catálogo maestro de la tienda, a los precios establecidos en dicho catálogo maestro:

```
<PriceTCMasterCatalogWithOptionalAdjustment>
</PriceTCMasterCatalogWithOptionalAdjustment>
```

- Los compradores pagan al vendedor los costes de envío:

```
<ShippingTCShippingCharge>
<ShippingChargePolicyRef policyName="StandardShippingChargeBySeller">
  <StoreRef name="&STORE_IDENTIFIER;">
    <Owner>
      <OrganizationRef distinguishName="ou=&ORGENTITYNAME;,
&ORGANIZATION_DN;" />
```

```

    </Owner>
  </StoreRef>
</ShippingChargePolicyRef>
</ShippingTCShippingCharge>

```

- Los compradores pueden devolver los productos sin sufrir ningún cargo por devolución. Los productos deben devolverse dentro del número de días definido en la política de negocio ApprovalByDays:

```

<ReturnTCReturnCharge>
  <ReturnChargePolicyRef policyName="NoCharges">
    <StoreRef name="&STORE_IDENTIFIER;">
      <Owner>
        <OrganizationRef distinguishName="ou=&ORGENTITYNAME;,
          &ORGANIZATION_DN;" />
      </Owner>
    </StoreRef>
  </ReturnChargePolicyRef>
  <ReturnApprovalPolicyRef policyName="ApprovalByDays">
    <StoreRef name="&STORE_IDENTIFIER;">
      <Owner>
        <OrganizationRef distinguishName="ou=&ORGENTITYNAME;,
          &ORGANIZATION_DN;" />
      </Owner>
    </StoreRef>
  </ReturnApprovalPolicyRef>
</ReturnTCReturnCharge>

```

Nota para usuarios de WebSphere Commerce Business Edition:

Omitir estos términos y condiciones en el contrato por omisión de la tienda indica que, por omisión, la tienda no acepta devoluciones. Sin embargo, otros contratos pueden permitir que los compradores hagan devoluciones, si se definen los términos y condiciones de las devoluciones.

Nota para usuarios de WebSphere Commerce Professional Edition:

Omitir estos términos y condiciones en el contrato por omisión de la tienda indica que la tienda no acepta devoluciones.

- Los reembolsos se pagan utilizando el mismo método de pago que ha utilizado el comprador para formalizar el pedido:

```

<ReturnTCRefundPaymentMethod>
<ReturnPaymentPolicyRef policyName="UseOriginalPayment">
<StoreRef name="&STORE_IDENTIFIER;">
  <Owner>
    <OrganizationRef distinguishName="ou=&ORGENTITYNAME;,
      &ORGANIZATION_DN;" />
  </Owner>
</StoreRef>
</ReturnPaymentPolicyRef>
</ReturnTCRefundPaymentMethod>

```

5. Cierre el archivo XML del modo siguiente:

```

</BuyerContract>
</Package>

```



Para obtener más información sobre cómo utilizar @ y &, consulte el Apéndice B, "Creación de los datos", en la página 465.

Creación de un archivo de contrato por omisión en formato DTD

Para crear un contrato por omisión, debe definir el contrato, el propietario del contrato, la descripción del contrato, los participantes del contrato y los términos y condiciones del contrato. La información sobre un contrato se almacena en cuatro tablas: CONTRACT, PARTICIPNT, TRADING y TERMCOND.

El contrato por omisión se asocia a la tienda mediante la tabla de base de datos STOREDEF. Para los usuarios de WebSphere Commerce Business Edition, los contratos distintos al contrato por omisión se asocian a una tienda mediante la tabla de base de datos STORECNTR.

Para crear un contrato por omisión en formato DTD, efectúe lo siguiente:

1. Defina el contrato por omisión en el archivo XML. El contrato por omisión se define al principio del archivo XML, tal como se muestra a continuación:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Trading SYSTEM "B2BTrading.dtd">
<Trading>
<Contract state="Active" origin="Manual"
  name="&STORE_IDENTIFIER; Default Contract" majorVersionNumber="1"
  minorVersionNumber="0" contractUsage="Default">
```

Las particiones de línea en el elemento Contract sólo aparecen a efectos de presentación.

2. Defina el propietario del contrato. Utilice el ejemplo siguiente como muestra:

```
<ContractOwner>
  <Member>
    <Organization distinguishName="ou=&ORGENTITYNAME;,&ORGANIZATION_DN;" />
  </Member>
</ContractOwner>
```

donde distinguishName es el nombre del usuario propietario del contrato con el formato de nombre distinguido LDAP. Por ejemplo, uid=johnsmith,ou=People,o=ibm,o=com.

3. Defina la descripción del contrato en el archivo XML de contrato. Utilice el ejemplo siguiente como muestra:

```
<ContractDescription title="Contrato por omisión de la tienda." languageId="-5">
</ContractDescription>
```

donde

- title es una descripción de texto del contrato.
- languageId es el idioma del título. Para languageId se han definido previamente los valores siguientes:
 - -1 (Inglés de EE.UU.)
 - -2 (Francés)
 - -3 (Alemán)
 - -4 (Italiano)
 - -5 (Español)
 - -6 (Portugués de Brasil)
 - -7 (Chino simplificado)
 - -8 (Chino tradicional)
 - -9 (Coreano)
 - -10 (Japonés)

Se pueden definir valores adicionales para languageId actualizando los elementos de idioma de la tienda. Para obtener más información sobre los elementos de idioma, consulte el Capítulo 22, "Elementos de idioma", en la página 233.

4. Defina los participantes del contrato en el archivo XML de contrato. Utilice el ejemplo siguiente como muestra:


```

<Participant role="Buyer">
</Participant>
<Participant role="Seller">
  <Member>
    <Organization distinguishName="ou=&ORGENTITYNAME;,&ORGANIZATION_DN;"/>
  </Member>
</Participant>

```

donde distinguishName es el nombre del usuario que es el vendedor para este contrato con el formato de nombre distinguido LDAP. Por ejemplo, uid=erickoeck,ou=People,o=ibm,o=com. En muchos casos, será igual que el propietario del contrato.

Nota: No se especifican miembros en el rol de participante de comprador porque el contrato está disponible para todos los usuarios con el rol de comprador.

- Defina los términos y condiciones en el archivo XML de contrato. Los elementos XML y los atributos son distintos para los diversos tipos de términos y condiciones. Utilice el archivo B2BTrading.dtd para saber qué elementos XML y atributos se han de utilizar para cada tipo de término. Al definir términos y condiciones, comúnmente se utilizan los atributos siguientes:

policyName

El nombre de la política de negocio a la que hace referencia el término y condición. El nombre se almacena en POLICY.POLICYNAME.

policyType

El tipo de política de negocio al que hace referencia el término y condición. Los valores válidos son:

- Price
- ProductSet
- InvoiceFormat
- Payment
- ReturnApproval
- ReturnCharge
- ReturnPayment
- ShippingCharge
- ShippingMode

storeIdentity

La tienda o grupo de tiendas para el término y condición.

distinguishName

El nombre del usuario que es el propietario de la tienda o grupo de tiendas. El nombre debe tener el formato de nombre distinguido LDAP. Por ejemplo, uid=wcsadmin,o=Root Organization.

Los siguientes términos y condiciones de ejemplo van precedidos de una descripción de lo que definen:

- Todos los compradores pueden adquirir todos los artículos del catálogo maestro de la tienda, a los precios establecidos en dicho catálogo maestro:

```

<TermCondition>
  <PriceTC>
    <PriceTCMasterCatalogWithOptionalAdjustment>
  </PriceTCMasterCatalogWithOptionalAdjustment>
  </PriceTC>
</TermCondition>

```

- Los compradores pagan al vendedor los costes de envío:

```
<TermCondition>
  <ShippingTC>
    <ShippingTCShippingCharge>
      <PolicyReference policyName="StandardShippingChargeBySeller"
        policyType="ShippingCharge" storeIdentity="&STORE_IDENTIFIER;">
        <Member>
          <Organization distinguishName="ou=&ORGENTITYNAME;,
            &ORGANIZATION_DN;" />
        </Member>
      </PolicyReference>
    </ShippingTCShippingCharge>
  </ShippingTC>
</TermCondition>
```

Las particiones de línea en el elemento PolicyReference sólo aparecen a efectos de presentación.

- Los compradores pueden devolver los productos sin sufrir ningún cargo por devolución. Los productos deben devolverse dentro del número de días definido en la política de negocio ApprovalByDays:

```
<TermCondition>
  <ReturnTC>
    <ReturnTCReturnCharge>
      <ReturnChargePolicyReference>
        <PolicyReference policyName="NoCharges"
          policyType="ReturnCharge"
          storeIdentity="&STORE_IDENTIFIER;">
          <Member>
            <Organization distinguishName="ou=&ORGENTITYNAME;,
              &ORGANIZATION_DN;" />
          </Member>
        </PolicyReference>
      </ReturnChargePolicyReference>
      <ReturnApprovalPolicyReference>
        <PolicyReference policyName="ApprovalByDays"
          policyType="ReturnApproval"
          storeIdentity="&STORE_IDENTIFIER;">
          <Member>
            <Organization distinguishName="ou=&ORGENTITYNAME;,
              &ORGANIZATION_DN;" />
          </Member>
        </PolicyReference>
      </ReturnApprovalPolicyReference>
    </ReturnTCReturnCharge>
  </ReturnTC>
</TermCondition>
```

Las particiones de línea en los elementos PolicyReference sólo aparecen a efectos de presentación.

Nota para usuarios de WebSphere Commerce Business Edition:

Omitir estos términos y condiciones en el contrato por omisión de la tienda indica que, por omisión, la tienda no acepta devoluciones. Sin embargo, otros contratos pueden permitir que los compradores hagan devoluciones, si se definen los términos y condiciones de devoluciones.

Nota para usuarios de WebSphere Commerce Professional Edition:

Omitir estos términos y condiciones en el contrato por omisión de la tienda indica que la tienda no acepta devoluciones.

- Los reembolsos se pagan utilizando el mismo método de pago que ha utilizado el comprador para formalizar el pedido:

```

<TermCondition>
  <ReturnTC>
    <ReturnTCRefundPaymentMethod>
      <PolicyReference policyName="UseOriginalPayment"
        policyType="ReturnPayment" storeIdentity="&STORE_IDENTIFIER;">
        <Member>
          <Organization distinguishName="ou=&ORGENTITYNAME;,
            &ORGANIZATION_DN;" />
        </Member>
      </PolicyReference>
    </ReturnTCRefundPaymentMethod>
  </ReturnTC>
</TermCondition>

```

Las particiones de línea en el elemento PolicyReference sólo aparecen a efectos de presentación.

6. Cierre el archivo XML del modo siguientes:

```

</Contract>
</Trading>

```



Para obtener más información sobre cómo utilizar @ y &, consulte el Apéndice B, "Creación de los datos", en la página 465.

Capítulo 19. Elementos de despacho de pedidos

Las tiendas utilizan los centros de despacho de pedidos como almacenes de inventario y como centros de envío y recepción. Un centro de despacho de pedidos representa un lugar desde el que se envían productos a los clientes. Las cuentas de inventario se mantienen por separado para cada centro de despacho de pedidos. Una tienda puede tener uno o muchos centros de despacho de pedidos asociados a ella. Un centro de despacho gestiona el inventario de productos y el envío de productos de una tienda. El despacho de pedidos incluye la selección, el empaquetado y el envío. La selección de productos puede efectuarse para una o más salidas de un centro de despacho de pedidos, el empaquetado consiste en poner estos productos en contenedores de envío y el envío es la acción de enviarlos a los clientes.

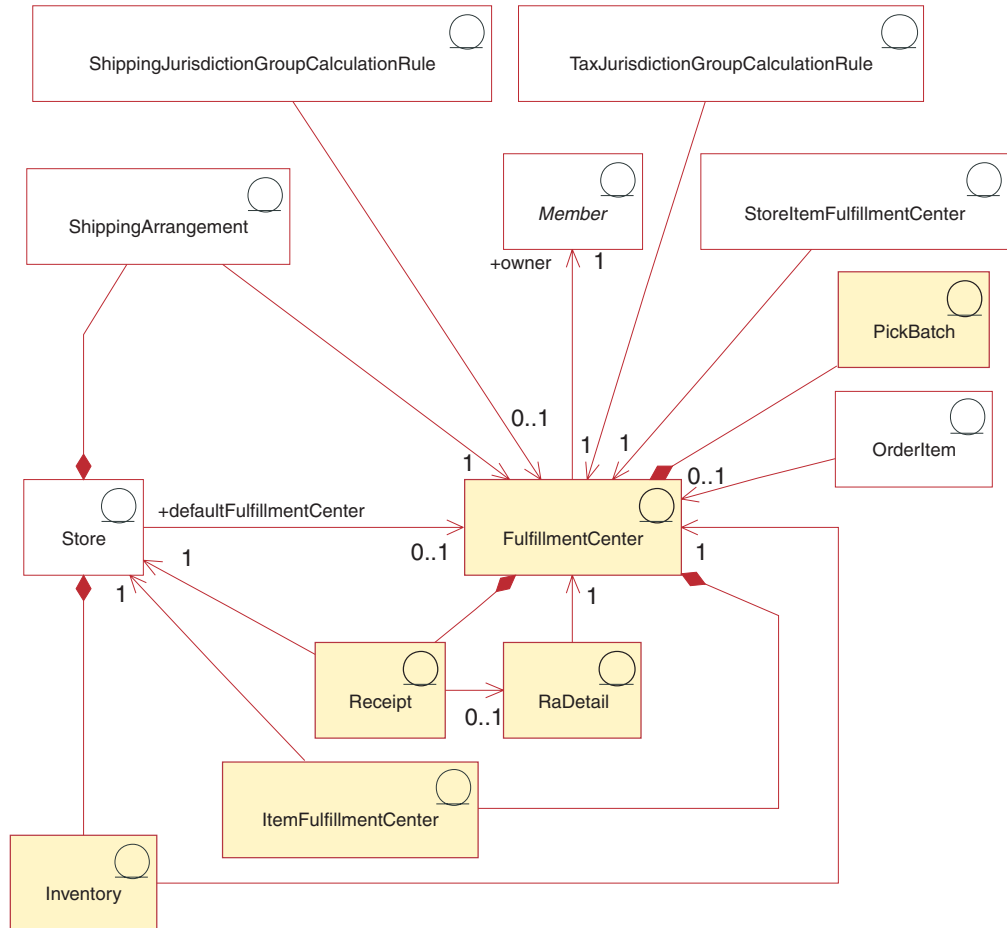
Los productos se configuran para el despacho de pedidos con el asistente Producto y el cuaderno Producto. La configuración de productos proporciona opciones para hacer un seguimiento del inventario, permitir pedidos en espera, forzar pedidos en espera, despachar el producto por separado y especificar que el producto no se debe devolver.

Normalmente hay varias personas trabajando a la vez en un centro de despacho de pedidos, cada una con una tarea o tareas diferentes a realizar. WebSphere Commerce Accelerator divide las tareas más comunes en roles, y estos roles se asignan a usuarios. En WebSphere Commerce Accelerator, deberá seleccionar un centro de despacho de pedidos en el momento de realizar la conexión, si se le han asignado uno o más roles pertenecientes al despacho de pedidos.

Nota: Para obtener más información sobre el despacho de pedidos, los centros de despacho de pedidos y los roles, consulte la ayuda en línea de WebSphere Commerce.

Los elementos de despacho de pedidos en WebSphere Commerce

Para comprender los elementos de despacho de pedido, es necesario comprender las relaciones entre el despacho de pedidos y la tienda. Esto puede explicarse mediante el uso de un modelo de información. Las secciones siguientes describen las relaciones que tiene el inventario con una tienda y otros elementos.



Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce Server. Para obtener más información sobre el modelo de información, consulte “Modelo de información de los datos de la tienda” en la página 105. Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, “Descripción de UML”, en la página 463.

Centro de despacho de pedidos

En el diagrama anterior, el *centro de despacho de pedidos* es el centro del proceso de despacho de pedidos. Un centro de despacho de pedidos tiene un propietario, definido en la tabla MEMBER. Cada tienda se puede asociar con varios centros de despacho de pedidos y un centro de despacho de pedidos puede tener varias tiendas asociadas. Tal como se indica en el diagrama, existen varias interacciones entre la tienda y el centro de despacho de pedidos. Para obtener más información sobre los elementos de tienda, consulte el apartado “Los elementos de tienda en WebSphere Commerce” en la página 135.

Recepciones

Los centros de despacho de pedidos reciben inventario para los artículos diaria, semanal o mensualmente. Cuando se recibe inventario para un artículo, se crea una *recepción* en la tabla RECEIPT, que registra la información sobre la cantidad recibida, así como la tienda que posee el inventario. A medida que se procesan los pedidos, la tabla RECEIPT se actualiza para reflejar los niveles actuales de inventario disponible. Para obtener más información sobre cómo crear recepciones, consulte el apartado “Creación de elementos de inventario en WebSphere Commerce” en la página 293.

RaDetail

RaDetail es la información detallada acerca de los artículos de un registro de inventario esperado. Esta información puede utilizarse para calcular cuándo se puede esperar recibir el inventario en un centro de despacho de pedidos y para proporcionar a los clientes las fechas de envío previstas para los artículos de pedidos en espera.

Inventario

Una tienda tiene *inventario* que está asociado con el centro de despacho de pedidos. El inventario incluye todo aquello que se puede contabilizar físicamente en un centro de despacho de pedidos. El inventario está asociado a una tienda y a un centro de despacho de pedidos. También se registra la información sobre el inventario que posee una tienda en el centro de despacho de pedidos, como por ejemplo, cantidades reservadas, cantidades en pedidos en espera y cantidades asignadas a los pedidos en espera. Esta información se almacena en la tabla ITEMFFMCTR. Para obtener más información sobre el inventario y los elementos de inventario, consulte el Capítulo 29, “Elementos de inventario”, en la página 289.

Acuerdos de envío

Un *acuerdo de envío* es una relación que permite que una tienda pueda utilizar un centro de despacho de pedidos. Los acuerdos de envío indican que un centro de despacho de pedidos puede enviar productos en nombre de una tienda utilizando una modalidad de envío. Cada tienda tiene un acuerdo de envío con un centro de despacho de pedidos y viceversa. Los acuerdos de envío se definen en la tabla SHPARRANGE. Para obtener más información sobre cómo crear acuerdos de envío, consulte el apartado “Creación de elementos de despacho de envío” en la página 261.

Otros elementos relacionados con el despacho de pedidos

Hay otras relaciones con un centro de despacho de pedidos que no están directamente relacionadas con una tienda. Un lote de requisición de artículos es un elemento que está asociado a un centro de despacho de pedidos. Un lote de requisición de artículos agrupa las salidas de los pedidos para su proceso como una unidad en un centro de despacho de pedidos y crea listas de requisición y listas de embalaje. Una vez que los artículos se han seleccionado y empaquetado, puede efectuarse el envío de la salida del pedido, y el envío puede confirmarse. La información sobre lotes de requisición se almacena en la tabla PICKBATCH. Un artículo de pedido también está asociado a un centro de despacho de pedidos. Un artículo es una instancia específica de un producto, definida por atributos. La información sobre cada artículo de un pedido se almacena en la tabla ORDERITEM. Para obtener más información sobre los elementos de pedido, consulte el Capítulo 30, “Elementos de pedido”, en la página 299.

Al igual que otras entidades, un centro de despacho de pedidos tiene normas que rigen algunas de sus acciones. Cada centro de despacho de pedidos tiene normas para los impuestos y los costes de envío. Estas normas están definidas en las tablas TAXJCRULE y SHPJCRULE, respectivamente. Para obtener más información sobre los elementos de envío y de impuestos, consulte el Capítulo 26, “Elementos de envío”, en la página 249 y el apartado “Los elementos de impuestos en WebSphere Commerce” en la página 265.



Para obtener información más detallada sobre la estructura de los elementos de despacho de pedidos en WebSphere Commerce Server, consulte los modelos de datos de despacho de pedidos en la ayuda en línea de WebSphere Commerce.

Creación de elementos de despacho de pedidos en WebSphere Commerce

Antes de que la tienda pueda enviar artículos a un cliente, debe definir el centro, o centros, de despacho de pedidos que suministrarán estos artículos. Esta información debe crearse en formato de archivos XML que se pueden cargar en la base de datos utilizando Loader package. Para obtener más información sobre Loader package, consulte la Parte 10, “Publicación de la tienda”, en la página 345.

Antes de crear elementos, debe familiarizarse con el material tratado en el Capítulo 38, “Carga de grupos de elementos de base de datos de WebSphere Commerce”, en la página 405.

Para crear elementos de despacho de pedidos para su tienda utilizando un archivo XML, haga lo siguiente:

1. Revise los archivos XML utilizados para crear elementos de despacho de pedidos para las tiendas de ejemplo. Todos los archivos para las tiendas de ejemplo se encuentran en el archivo archivador de tienda correspondiente.

Los archivos de archivador de tienda se encuentran en el directorio siguiente:

- `dir_instalación_WC/sampletores`

Nota: La ayuda en línea de WebSphere Commerce contiene información sobre cada uno de los elementos de datos contenidos en las tiendas de ejemplo.

Cada tienda de ejemplo incluye un archivo `fulfillment.xml`, que incluye la información de despacho de pedidos. Para ver el archivo `fulfillment.xml` del archivador de tienda, descomprímalo utilizando un programa ZIP. El archivo `fulfillment.xml` se encuentra en el directorio de datos.

2. Revise la información del Apéndice B, “Creación de los datos”, en la página 465.
3. Cree un archivo `fulfillment.xml`, copiando uno de los archivos `fulfillment.xml` de los archivadores de tienda de ejemplo o creando uno nuevo. Para obtener más información, consulte el archivo DTD que corresponde a `fulfillment.xml`. Los archivos DTD se encuentran en el directorio siguiente:
 - `dir_instalación_WC/xml/sar`
4. Defina el centro, o los centros, de despacho de pedidos que la tienda soporta:
 - a. Utilizando como guía el ejemplo siguiente, defina un centro de despacho de pedidos en el archivo XML, en la tabla FFMCENTER:


```
<ffmcenter
ffmcenter_id="@ffmcenter_id_1"
  member_id="@seller_b2b_mbr_id"
  name="ToolTech Home"
  defaultshipoffset="0"
  markfordelete="0"
/>
```

donde

- ffmcenter_id es una clave exclusiva generada
 - member_id es el propietario del centro de despacho de pedidos
 - name es una serie que, junto con el propietario, identifica de forma exclusiva el centro de despacho de pedidos.
 - defaultshipoffset es una estimación del número de segundos que tarda en enviarse un artículo desde este centro de despacho de pedidos. Este valor se puede modificar en la tabla STORITMFFC.
 - markfordelete indica si el centro de despacho de pedidos debe suprimirse: 0 = no suprimir. 1 = suprimir si ya no se usa. Para obtener más detalles, consulte la información sobre el programa de utilidad de Limpieza de base de datos en la ayuda en línea de WebSphere Commerce.
- b. Utilizando como guía el ejemplo siguiente, describa el centro de despacho de pedidos en el archivo XML, en la tabla FFMCENTDS. Si está creando una tienda multicultural, deberá incluir esta información en un archivo XML específico del entorno nacional.

```
<ffmcentds
ffmcenter_id="@ffmcenter_id_1"
  description="El centro de despacho de pedidos que proporciona
              productos a ToolTech."
  language_id="@es_ES"
  displayname="Centro de despacho de pedidos de ToolTech"
  staddress_id="@staddress_id_es_ES_1"
/>
```

donde

- ffmcenter_id es una clave exclusiva generada
 - description es una descripción del centro de despacho de pedidos adecuada para mostrarla a un cliente.
 - language_id es el idioma en que se visualiza esta información. (Para obtener más información acerca del soporte de idiomas diferentes, consulte el Capítulo 34, "Globalización", en la página 321.)
 - displayname es el nombre del centro de despacho de pedidos adecuado para mostrarlo a un cliente.
 - staddress_id_loc es la ubicación física del centro de despacho de pedidos.
- c. Repita los pasos a y b para todos los centros de despacho de pedidos que la tienda soporte.



Para obtener más información sobre cómo utilizar @ y &, consulte el Apéndice B, "Creación de los datos", en la página 465.

Creación de elementos de despacho de pedidos (no ATP)

Después de haber definido el centro o los centros de despacho de pedidos que proporcionarán artículos para la tienda, deberá asociar un centro de despacho de pedidos a cada producto. Es decir, deberá identificar qué centro de despacho de

pedidos suministrará cada uno de los productos. Para crear esta relación, añada información a la tabla INVENTORY. Esta información debe crearse en formato de archivos XML, que se pueden cargar en la base de datos utilizando Loader package. Para obtener más información sobre Loader package, consulte la Parte 10, "Publicación de la tienda", en la página 345.

Nota:

1. Para poder asociar una tienda con un centro de despacho de pedidos, deberá crear elementos de tienda. Para obtener más información sobre la creación de elementos de tienda, consulte "Creación de elementos de datos de tienda en un archivo XML" en la página 136. También debe crear los elementos de catálogo antes de crear los elementos de despacho de pedidos de la tienda. Para obtener más información, consulte "Visualización de los elementos de catálogo de la tienda" en la página 177.
2. Cree elementos de despacho de pedidos de tienda sólo si implementa el despacho de pedidos no ATP. La tabla INVENTORY no la utiliza una tienda que incluye las funciones ATP.

Para crear la relación entre una tienda y un centro de despacho de pedidos utilizando un archivo XML, haga lo siguiente:

1. Revise los archivos XML que se utilizan para crear elementos de despacho de pedidos de la tienda para las tiendas de ejemplo. Todos los archivos para las tiendas de ejemplo se encuentran en el archivo archivador de tienda correspondiente.

Los archivos de archivador de tienda se encuentran en el directorio siguiente:

- *dir_instalación_WC/samplestores*

Nota: La ayuda en línea de WebSphere Commerce contiene información sobre cada uno de los elementos de datos contenidos en las tiendas de ejemplo.

Cada tienda de ejemplo incluye un archivo *storefulfill.xml*, que incluye la información de despacho de pedidos de la tienda. Para ver el archivo *storefulfill.xml* del archivador de tienda, descomprímalo utilizando un programa ZIP. El archivo *storefulfill.xml* se encuentra en el directorio de datos.

2. Revise la información del Apéndice B, "Creación de los datos", en la página 465.
3. Cree un archivo *storefulfill.xml*, copiando uno de los archivos *storefulfill.xml* de los archivadores de tienda de ejemplo o creando uno nuevo. Para obtener más información, consulte el archivo *wcs.dtd* en el directorio *dir_instalación_WC/schema/xml* o la DTD incluida en el archivador de tienda.
4. Utilizando como guía el ejemplo siguiente, cree una relación de tienda y centro de despacho de pedidos en el archivo XML, añadiendo información en la tabla INVENTORY.

```
<inventory
catentry_id="@catentry_id_1470"
quantity="100"
ffmcenter_id="@ffmcenter_id_1"
store_id="@storeent_id_1"
quantitymeasure="C62"
inventoryflags="0"
/>
```

donde

- `catentry_id` es la entrada de catálogo que suministrará este centro de despacho de pedidos.
- `quantity` es la cantidad, en las unidades indicadas por `QUANTITYMEASURE`, disponible en este centro de despacho de pedidos.
- `ffmcenter_id` es el centro de despacho de pedidos que suministrará el inventario.
- `store_id` es la tienda para la que se suministra el inventario.
- `quantitymeasure` es la unidad de medida para `QUANTITY`.
- `inventoryflags` son los distintivos de bit que indican cómo se utiliza `QUANTITY`:
 - 1 = `noUpdate`. El mandato de tarea por omisión `UpdateInventory` no actualiza `QUANTITY`.
 - 2 = `noCheck`. Los mandatos de tarea por omisión `CheckInventory` y `UpdateInventory` no comprueban `QUANTITY`.

5. Repita el paso 3 para cada entrada de catálogo de su tienda.



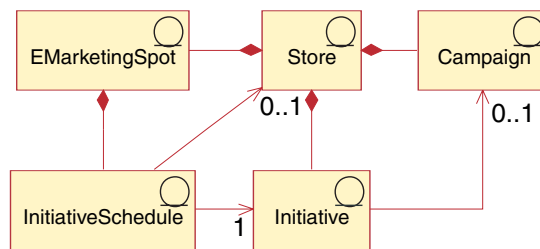
Para obtener más información sobre cómo utilizar @ y &, consulte el Apéndice B, "Creación de los datos", en la página 465.

Capítulo 20. Elementos de campaña

Las campañas sirven para organizar las actividades de marketing. Las campañas las crea normalmente un Director de marketing o un Director de comercialización. Generalmente están asociadas con un conjunto determinado de objetivos. Por ejemplo, una campaña de "Vuelta al cole" puede tener el objetivo de incrementar las ventas de ropa infantil durante la campaña.

Las campañas en WebSphere Commerce

El diagrama siguiente ilustra los elementos de campaña de WebSphere Commerce Server:

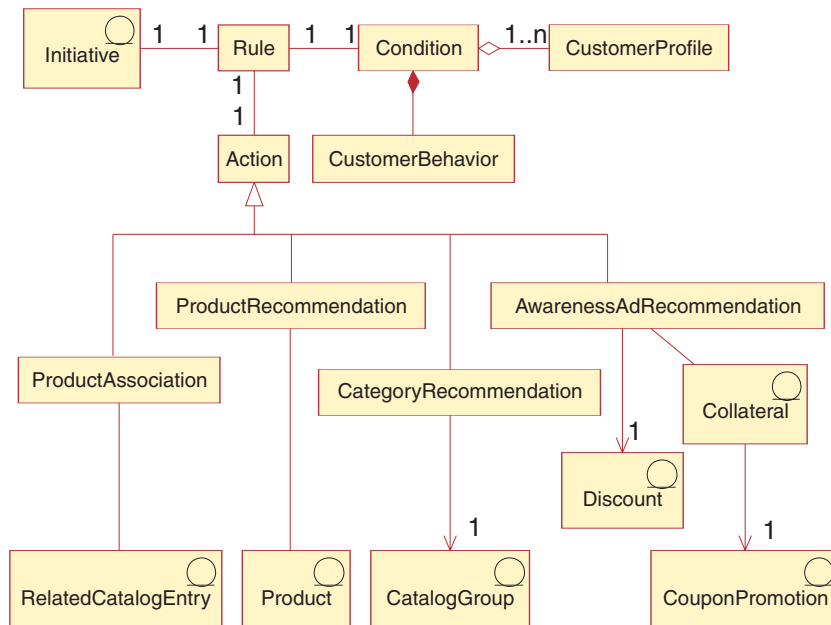


Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce Server. Para obtener más información sobre el modelo de información, consulte "Modelo de información de los datos de la tienda" en la página 105. Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, "Descripción de UML", en la página 463.

El ámbito de las campañas y los elementos asociados son las tiendas.

En WebSphere Commerce, las campañas contienen cualquier número de iniciativas de campaña, que definen una condición. Las iniciativas de campaña generan contenido destinado a los clientes, cuando se cumple la condición definida. El resultado es que una campaña es el elemento de marketing de alto nivel que organiza las iniciativas. A continuación se visualiza el modelo de información de

iniciativa de campaña:



Las iniciativas de campaña se asocian con una campaña que contenga un conjunto de iniciativas. Como ejemplo de esta relación, si una tienda de artículos de oficina tuviese una campaña de "Vuelta al cole", las iniciativas serían responsables de las acciones de nivel inferior, por ejemplo anunciar un descuento en los lápices o sugerir papel rayado a cualquier cliente que se haya registrado y haya indicado que es un estudiante.

Las iniciativas de campaña pueden mostrar cuatro tipos de contenido dinámico:

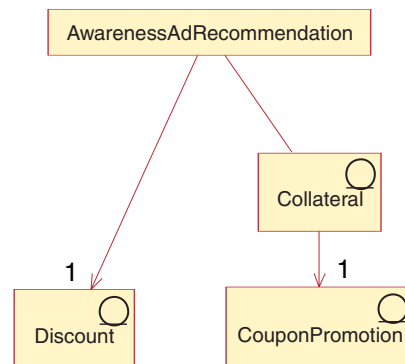
- Iniciativa de ventas por sugerencia
- Recomendación basada en filtrado colaborativo
- Anuncios publicitarios informativos
- Asociación de comercialización

El contenido de las ventas por sugerencia proporciona recomendaciones de productos y categorías según unas normas; estas sugerencias van destinadas a un público de clientes específico, basándose en el perfil del cliente y el comportamiento de otros clientes. Los comportamientos de los clientes a los que van destinadas son el valor total del carro de la compra, el contenido del carro de la compra y el contenido del historial de compras del cliente.

Las recomendaciones basadas en el filtrado colaborativo también crean recomendaciones de productos, pero utilizan un algoritmo de recomendación diferente, que presenta artículos basándose en el comportamiento general de los clientes, en lugar de hacerlo en normas predefinidas.

Los anuncios publicitarios informativos están diseñados para dirigir el contenido de un anuncio a un grupo de clientes específico, basándose en los mismos criterios que los utilizados para la venta por sugerencia, pero su fin es que el cliente tenga más información sobre las actividades de la tienda en línea, resaltar las ofertas especiales y aumentar el conocimiento de los tipos de tarjeta. Los anuncios

publicitarios siguen el modelo de información que se muestra aquí:



Las asociaciones de comercialización crean oportunidades de venta cruzada y venta ascendente basadas en las asociaciones estáticas definidas en el catálogo. Para crear una iniciativa de este tipo, se debe definir un método para seleccionar el producto fuente en la asociación, de modo que al invocar la zona de e-Marketing se utilice la fuente adecuada y se devuelvan los productos a los que va destinada la iniciativa. El método puede seleccionar la fuente basándose en el contenido de la página actual, el contenido del carro de la compra o el contenido del historial de compras del cliente como fuentes de la asociación.

Las iniciativas pueden incorporarse en cualquier página del sitio. Cuando se diseña el sitio, se colocan espacios reservados especiales, llamados Zonas de e-marketing, en el sitio. Cuando se muestran a un cliente, estos espacios reservados se sustituyen por el contenido específico deseado. Las ubicaciones de destino se asignan planificando la visualización de iniciativas en zonas de e-Marketing de las ubicaciones deseadas. Para obtener más información sobre cómo añadir zonas de e-marketing a su tienda, consulte el Capítulo 42, "Adición de zonas de e-marketing a la tienda", en la página 455.

Las iniciativas de campaña contienen una condición que determina cuándo y a quién se muestran. Esta condición se define cuando se crea la iniciativa y se puede cambiar durante el tiempo de vida de la iniciativa para ajustar la visibilidad de la iniciativa y el contenido visualizado. Para obtener más información acerca de los perfiles de cliente, consulte el Capítulo 32, "Perfiles de cliente", en la página 307.

Las iniciativas de campaña generan estadísticas sobre su uso. Estas estadísticas pueden verlas los comerciantes, los directores de marketing y los directores de comercialización utilizando WebSphere Commerce Accelerator. Las estadísticas ilustran el índice de pulsaciones de una iniciativa para cada zona de e-marketing donde se implementa. Estas estadísticas proporcionan información sobre la eficacia de la iniciativa, así como porcentajes comparativos de éxito entre las diversas ubicaciones en las que se visualiza.

Creación de elementos de campaña en WebSphere Commerce

Las campañas y las iniciativas de campaña las crea normalmente un Director de marketing o un Director de comercialización, mediante los asistentes de campaña o de iniciativa de campaña de WebSphere Commerce Accelerator. Para obtener más información, consulte la ayuda en línea de WebSphere Commerce.

Para obtener más información sobre cómo añadir zonas de e-marketing a su tienda, consulte el Capítulo 42, "Adición de zonas de e-marketing a la tienda", en la página 455.

Capítulo 21. Herramientas de Payments

WebSphere Commerce proporciona un componente opcional llamado WebSphere Commerce Payments (conocido anteriormente como IBM Payment Manager).

Si desea utilizar WebSphere Commerce Payments en la tienda, debe incluir un archivo de elementos de pago en el archivador de tienda. Antes de publicar el archivador de tienda, asegúrese de que se haya iniciado la instancia de Payments, a continuación, cuando se haya publicado el archivador de tienda, el archivo de elementos de pago (que se incluye como parte de los archivadores de tienda de ejemplo) establece la información siguiente en WebSphere Commerce Payments:

- El ID_comerciante de la base de datos de WebSphere Commerce Payments.
- El tipo de casete que se utiliza en la tienda.
- Una cuenta de la base de datos de WebSphere Commerce Payments para cada moneda soportada por la tienda y especificada en el archivo de elementos de pago. Si la tienda no da soporte a la moneda especificada en el archivo de elementos de pago, no se creará la cuenta.
- Un tipo o tipos de tarjeta para cada cuenta.

Para establecer un archivo de elementos de pago y configurar la tienda para que utilice WebSphere Commerce Payments, efectúe lo siguiente:

- Cree datos de pago con el formato de un archivo XML (`paymentinfo.xml`) que se carga durante la publicación de la tienda mediante la Consola de la administración. Esto configurará WebSphere Commerce Payments con el comerciante y los tipos de tarjetas especificados para la tienda que se está publicando. Para obtener más información, consulte el apartado “Creación de elementos de pago utilizando un archivo XML” en la página 228.

Nota: `paymentinfo.xml` no llena con datos las tablas de la base de datos de WebSphere Commerce Server. Configura WebSphere Commerce Payments. `paymentinfo.xml` sólo es aplicable si está utilizando como método de pago las tarjetas de crédito fuera de línea.

Una vez publicado el archivador de tienda, puede formalizar pedidos utilizando la información de pago configurada en el archivador de tienda de ejemplo. Si desea añadir nuevas marcas de tarjeta, debe configurar WebSphere Commerce Payments para que funcione con cada marca.

- Si va a utilizar un casete de pago de IBM distinto del casete OfflineCard o CustomOffline, modifique los elementos Web de tienda de ejemplo como se describe en el apartado “Personalización del entorno para un casete de pago” en la página 229.
- Complete la configuración de WebSphere Commerce Payments para la tienda mediante la Consola de administración o la interfaz de usuario de WebSphere Commerce Payments. Si utiliza la Consola de administración, los elementos de menú aparecen en el menú Payments. Si utiliza la interfaz de usuario de WebSphere Commerce Payments, los elementos de menú aparecen bajo Administración en el marco de navegación. Para obtener más información sobre las tareas de configuración, consulte el tema *Configuración de WebSphere Commerce Payments para la tienda* en la Ayuda en línea a la producción de WebSphere Commerce.

Si piensa utilizar un mecanismo de pago distinto de WebSphere Commerce Payments, los pasos que debe seguir para utilizar el mecanismo de pago son similares a los procedimientos siguientes.

Creación de elementos de pago utilizando un archivo XML

Para crear elementos de pago para la tienda utilizando un archivo XML, realice lo siguiente:

1. Revise los archivos XML utilizados para crear elementos de pago para las tiendas de ejemplo. Todos los archivos para las tiendas de ejemplo se encuentran en el archivo archivador de tienda correspondiente.

Los archivos de archivador de tienda están ubicados en el directorio siguiente:

- *dir_instalación_WC/samplestores*

Nota: La ayuda en línea de WebSphere Commerce contiene información sobre cada uno de los elementos de datos contenidos en las tiendas de ejemplo.

Cada tienda de ejemplo incluye un archivo `paymentinfo.xml`, que contiene la información de pago. Para ver el archivo `paymentinfo.xml` del archivador de tienda, descomprímalo utilizando un programa ZIP. Los archivos `paymentinfo.xml` se encuentran en el directorio de datos.

2. Cree un archivo `paymentinfo.xml`, copiando uno de los archivos `paymentinfo.xml` de los archivadores de tienda de ejemplo o creando uno nuevo. Para obtener más información, consulte el archivo DTD correspondiente a `paymentinfo.xml`. El archivo DTD está situado en el directorio siguiente:
dir_instalación_WC/xml/sar.
3. Habilite o inhabilite WebSphere Commerce Payments.
 - a. Utilizando como guía el ejemplo siguiente, habilite o inhabilite WebSphere Commerce Payments en el archivo XML y especifique los tipos de casete de pago, monedas y tipos de tarjeta que acepta la tienda.

```
<paymentinfo>
  <PaymentManager enable="yes"/>
  <Cassette type="OfflineCard">
    <Account currency="USD">
      <Brand type="MasterCard"/>
      <Brand type="VISA"/>
      <Brand type="American Express"/>
    </Account>
    <Account currency="EUR">
      <Brand type="MasterCard"/>
      <Brand type="VISA"/>
      <Brand type="American Express"/>
    </Account>
  </Cassette>
</paymentinfo>
```

donde:

- `enable` indica si WebSphere Commerce Payments está habilitado o inhabilitado. Cuando WebSphere Commerce Payments está inhabilitado, la tienda no podrá procesar las transacciones de pago mediante el componente Payments, aunque la interfaz de usuario de Payments continuará funcionando. Si inhabilita el componente Payments, no es necesario especificar otros elementos del elemento `paymentinfo`.
- `Cassette type` es el tipo de casete soportado.
- `Account currency` son las monedas a las que da soporte la tienda. Este valor es necesario si está utilizando el tipo de casete `OfflineCard`. La

moneda se debe identificar mediante un código de tres letras que se ajuste al estándar ISO 4217. Por ejemplo, "USD" para dólares de EE.UU.

- Brand type es el tipo de tarjeta de crédito soportado por Account currency.

Personalización del entorno para un casete de pago

WebSphere Commerce proporciona tiendas de ejemplo que pueden utilizar el casete OfflineCard o CustomOffline como el casete de pago para manejar transacciones de pago. Estos casetes se configuran automáticamente de modo que están disponibles para ser utilizados como método de pago para las tiendas de ejemplo. Para utilizar cualquier otro casete de pago se deben modificar los elementos Web de las tiendas de ejemplo. Las instrucciones de este apartado describen cómo puede personalizar el entorno para utilizar otros casetes de pago de IBM que se proporcionan con WebSphere Commerce.

Para que una tienda utilice un casete de pago de IBM, en primer lugar debe haber seleccionado el componente WebSphere Commerce Payments para su instalación. En la publicación *WebSphere Commerce, Guía de instalación* encontrará las instrucciones de instalación para su plataforma. El programa de instalación de WebSphere Commerce instala tanto la infraestructura de Payments como el software del casete. A continuación, debe utilizar el Gestor de configuración de WebSphere Commerce para realizar las tareas posteriores a la instalación necesarias, como por ejemplo crear una instancia de Payments y añadir un casete a una instancia. Consulte la publicación *WebSphere Commerce, Guía de instalación* y la ayuda en línea del Gestor de configuración para obtener instrucciones sobre cómo configurar una instancia de Payments.

Después de añadir un casete de pago a una instancia de Payments, revise los pasos de personalización siguientes para asegurarse de que la tienda de ejemplo de WebSphere Commerce puede procesar pagos mediante el casete de pago que ha seleccionado:

1. Modificar el archivo .jsp de la tienda para especificar el casete de pago.
2. Comprobar el perfil de cajero del casete.
3. Comprobar el archivo .jsp del casete que da soporte a la página de formalización de pedidos del encargado de los pedidos (pedidos de invitado).
4. Configurar los Valores de comerciantes.

En los apartados siguientes se describen estos pasos.

Modificar el archivo .jsp de la tienda

Si no va a utilizar el casete OfflineCard o CustomOffline con la tienda de ejemplo, debe modificar el archivo .jsp de la tienda. Por omisión, el archivo .jsp de la tienda se configura para que utilice el casete OfflineCard, por lo tanto, para utilizar cualquier otro casete debe modificar el archivo. La tienda FashionFlow también utiliza el casete CustomOffline.

Para modificar el archivo .jsp efectúe los pasos siguientes:

1. Cree una tienda en WebSphere Commerce utilizando una tienda de ejemplo como, por ejemplo, FashionFlow.
2. Vaya al directorio siguiente:

```
dir_instalación_WAS/installedApps/nombre_célula/  
nombre_instancia_WC.ear/Stores.war/
```

► 400 En iSeries, la vía de acceso es:

*dir_usuario_WAS/nombre_instancia_WAS/
installedApps/nombre_célula/nombre_instancia_WC.ear/Stores.war/La tienda
creada tendrá su propio directorio en el directorio war.*

3. En el directorio de la tienda, abra el archivo `OrderSubmitForm.jsp` con un editor de texto.

► **Business** La herramienta Contratos de WebSphere Commerce Accelerator da soporte a todos los casetes de pago. El archivo `OrderSubmitForm.jsp` debe respetar los términos y condiciones del contrato establecido entre la organización compradora y la vendedora.

4. Busque el texto siguiente en el archivo `OrderSubmitForm.jsp`:

```
if (info[i].getPolicyName().trim().equals("OfflineCard"))
```

Cambie el nombre de la política de pago de `OfflineCard` por uno de los siguientes, según corresponda:

```
CustomOffline  
BankServACH  
Paymentech  
VisaNet  
VisaNet_PCard
```

Para obtener más información, consulte la tabla de base de datos POLICY en la ayuda en línea al desarrollo de WebSphere Commerce.

La política para `CustomOffline` da soporte al proceso de transacciones de pago personalizadas como, por ejemplo, entrega contra reembolso o COD, Bill-Me-Later o cupones que se ejecutan generalmente fuera de WebSphere Commerce Payments.

La política para `BankServACH` da soporte al proceso de pagos de cheques electrónicos en línea utilizando la pasarela de pago `BankServ` que interactúa con ACH (Automated Clearing House Network).

La política para `Paymentech` da soporte a la autorización en línea y a los pagos con tarjetas de crédito y con tarjetas de crédito que no están basadas en el PIN.

La política para `VisaNet` da soporte al proceso de transacciones de tarjetas de crédito utilizando la red `Vital Processing Services` o `FHMS` (First Horizon Merchant Services).

Nota: Si está utilizando el casete para `VisaNet` con el soporte de tarjeta de compras, seleccione `VisaNet_PCard` en lugar de `VisaNet`.

Para obtener más información acerca de estos casetes, consulte los suplementos del casete.

Si la tienda utiliza la función Caja rápida, también debe cambiar el nombre de la política de pago en estos otros archivos:

```
ShoppingArea\CheckoutSection\QuickCheckoutSubsection\QuickCheckoutForm.jsp  
UserArea\AccountSection\QuickCheckoutProfileSubsection\QuickCheckoutProfileForm.jsp
```

5. (Opcional) Si está utilizando el método de pago por tarjeta de crédito y necesita añadir campos adicionales en la interfaz de usuario para recopilar información adicional del usuario, revise también el archivo `StandardCreditCard.jsp` para realizar posibles modificaciones. Este archivo está ubicado en la vía de acceso siguiente:

Comprobar el perfil de cajero para el casete

Debe haber disponible un perfil de cajero de WebSphere Commerce Payments para cada casete de pago de IBM que se proporciona con WebSphere Commerce. El perfil de cajero se utiliza para crear pedidos en el componente Payments.

Es posible que desee editar el perfil de cajero para establecer determinados parámetros como, por ejemplo, los parámetros APPROVEFLAG y DEPOSITFLAG. Dado que no todos los parámetros de casetes son similares, consulte las publicaciones adicionales relacionadas con el casete para obtener más información acerca de cómo establecer los parámetros:

- *WebSphere Commerce Payments, Suplemento del casete CustomOffline*
- *WebSphere Commerce Payments, Suplemento del casete OfflineCard*
- *WebSphere Commerce Payments Cassette for BankServACH Supplement*
- *WebSphere Commerce Payments Cassette for Paymentech Supplement*
- *WebSphere Commerce Payments Cassette for VisaNet Supplement*

Los perfiles de cajero asociados con los casetes de pago proporcionados por IBM incluyen los siguientes:

WC51_BankServACH.profile
WC51_CustomOffline_BillMe.profile
WC51_CustomOffline_COD.profile
WC51_OfflineCard.profile
WC51_VisaNet.profile
WC51_VisaNet_PCard.profile
WC_Paymentech.profile

El perfil de casete se debe almacenar en el directorio de perfiles de instancias de WebSphere Commerce.

Para localizar el directorio en el que se almacena el perfil, busque el archivo de configuración de WebSphere Commerce para la instancia que ha creado. Si ha utilizado el nombre de instancia por omisión "demo", el archivo de configuración será:

dir_instalación_WC/instances/demo/xml/demo.xml

▶ 400 Para iSeries, la vía de acceso debe ser:

dir_usuario_WC/instances/demo/xml/demo.xml

A continuación, busque el directorio que especifica el atributo ProfilePath del elemento Payment Manager del archivo de configuración. Este atributo especifica dónde debe ubicarse el perfil. Si ha utilizado el nombre de instancia por omisión "demo", la vía de acceso al directorio en el que se almacenará el perfil será la siguiente:

dir_instalación_WC/instances/demo/xml/payment

▶ 400 En iSeries, la vía de acceso es:

dir_usuario_WC/instances/demo/xml/payment

Si está editando el perfil de cajero del casete para establecer los parámetros, asegúrese de que guarda el perfil en el directorio *dir_instalación_WC/instancias/nombre_instancia/xml/payment*, donde *nombre_instancia* es el nombre de la instancia que está utilizando.

▶ 400 En iSeries, la vía de acceso es:

dir_usuario_WC/instancias/nombre_instancia/xml/payment

El perfil de cajero real que utiliza una política de negocio de pago se especifica mediante el valor de la propiedad *profileName* en el campo Propiedades de la política de negocio de pago. Consulte la tabla POLICY en la ayuda en línea para obtener más información acerca de las políticas de negocio.

Comprobar el archivo .jsp del casete

Los pagos se procesan mediante WebSphere Commerce Accelerator si un encargado de los pedidos formaliza un pedido de invitado en nombre de un cliente. Los datos de pago para el casete se recopilan mediante el archivo .jsp del casete.

El archivo .jsp del casete se denomina la “página de atributos de pago” en WebSphere Commerce. La página real se especifica mediante el valor de la propiedad *attrPageName* en el campo Propiedades de la política de negocio de pago. Para obtener más información, consulte la tabla de base de datos POLICY en la ayuda en línea al desarrollo de WebSphere Commerce. Tanto el flujo de tienda como el flujo de WebSphere Commerce Accelerator deben utilizar la página de atributos de pago.

El archivo .jsp del casete debe encontrarse ya en el directorio siguiente:

*dir_instalación_WC/wc.ear/CommerceAccelerator.war/tools/order/buyPages/
dir_instalación_WAS/installedApps/nombre_célula/WC_demo.ear/
CommerceAccelerator.war/tools/order/buypages*

▶ 400 En iSeries, la vía de acceso es:

*QIBM/userdata/webas5/base/nombre_instancia_WAS/installedApps/
nombre_célula/WC_demo.ear/CommerceAccelerator.war/tools/order/buypages*

Si desea personaliza la información de la “página de compras”, modifique el archivo .jsp como corresponda.

Configurar los valores de comerciantes en WebSphere Commerce Payments

Para configurar el comerciante para el casete de pago de IBM, siga las instrucciones que se proporcionan en las publicaciones adicionales para el casete. Los valores de comerciantes se pueden modificar mediante la Consola de administración de WebSphere Commerce o mediante la interfaz de usuario de Payments (http://nombre_sistema_principal:puerto/webapp/PaymentManager). Debe tener autorización de Administrador de Payments o de Administrador de comerciantes en WebSphere Commerce Payments para poder configurar los valores de comerciantes.

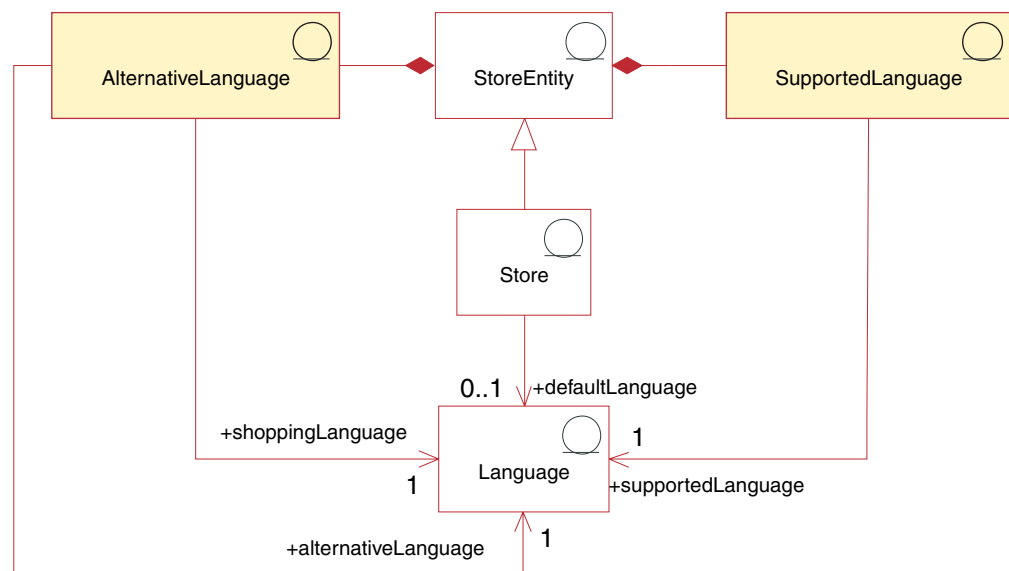
Capítulo 22. Elementos de idioma

En WebSphere Commerce, un sitio puede definir muchos idiomas que se pueden utilizar en el mismo. Durante la creación de instancias, la tabla LANGUAGE define los diez idiomas soportados, que incluyen alemán, chino tradicional y chino simplificado, japonés, coreano, italiano, francés, español, portugués de Brasil e inglés. Los sitios pueden definir idiomas adicionales o dialectos de las lenguas existentes para personalizar el modo en que se presenta la información a los clientes de culturas diferentes o cuyos datos estadísticos sean diferentes.

Los elementos de idioma en WebSphere Commerce

Para comprender los elementos de idioma, es necesario conocer las relaciones entre los idiomas y la tienda. Esto puede explicarse mediante el uso de un modelo de información, que se muestra a continuación. La sección siguiente describe las relaciones y asociaciones que tiene el idioma con una tienda y otros elementos.

El diagrama siguiente muestra el modelo de información de los elementos de idioma.



En WebSphere Commerce, los idiomas se clasifican en cuatro tipos. Estos son:

- Idioma por omisión,
- Idioma soportado,
- Idioma alternativo e
- Idioma de compras.

Cada uno de estos tipos de idioma lleva a cabo un rol distinto en la tienda. Todos los idiomas se almacenan en la tabla LANGUAGE.

Idioma por omisión

Cada tienda tiene asociado un *idioma por omisión*. Este es el idioma que la tienda ha elegido para utilizar como idioma principal y será el idioma que se mostrará en

primer lugar a los clientes que seleccionan de forma explícita un idioma de compras. El idioma por omisión de una tienda está soportado de forma implícita por la tienda, es decir, la tienda siempre debe poder mostrar la información en el idioma por omisión, o en uno de los idiomas alternativos, si hay alguno definido en la tabla LANGPAIR. Cuando la información no esté disponible en uno de los idiomas soportados o idiomas alternativos, la información se mostrará en el idioma por omisión.

Idiomas soportados

La tabla STORELANG indica los idiomas que soporta cada tienda. Una tienda debe poder mostrar información en sus *idiomas soportados* o en sus idiomas alternativos, si están definidos en la tabla LANGPAIR. Una tienda también debe soportar todos los idiomas soportados por su grupo de tiendas.

Para obtener más información acerca de cómo añadir un idioma soportado, consulte el apartado “Adición de un idioma a una tienda” en la página 334.

Idioma alternativo

Cuando no hay información disponible en uno de los idiomas soportados, la tienda intenta mostrar la información en un *idioma alternativo*, si está disponible. Una tienda puede especificar el orden en que se han de intentar sus idiomas alternativos. Los idiomas alternativos de una tienda incluyen los idiomas alternativos de su grupo de tiendas. Los idiomas alternativos pueden ser útiles cuando hay información que solamente está disponible en un idioma, pero debería estar disponible para los clientes que compran en un idioma diferente pero afín. Esto puede ser el caso cuando, por ejemplo, todavía no se ha traducido toda la información en todos los idiomas soportados o cuando, por ejemplo, se da soporte a dos dialectos muy similares de la misma lengua, a veces con la misma información.



Para obtener información más detallada sobre la estructura de los elementos de idioma en WebSphere Commerce Server, consulte los modelos de datos de idioma en la ayuda en línea de WebSphere Commerce.

Creación de elementos de idioma en WebSphere Commerce

Puede definir los idiomas a los que da soporte su tienda de una de las dos maneras siguientes:

- Utilizando las herramientas de tienda de WebSphere Commerce Accelerator
- En un archivo XML que cargará Loader package o mediante la herramienta de publicación de la Consola de administración
- Editando directamente la base de datos utilizando inserciones SQL
- Utilizando ediciones y actualizaciones SQL

Nota: Las herramientas trabajan con archivos XML con datos preinsertados, en forma de archivador de tienda.

Para obtener más información sobre cómo definir los idiomas soportados por la tienda utilizando las herramientas de tienda, consulte la ayuda en línea de WebSphere Commerce. Para obtener más información sobre cómo definir los idiomas soportados por la tienda en un archivo XML, consulte “Creación de elementos de datos de tienda en un archivo XML” en la página 136.

Capítulo 23. Elementos de moneda

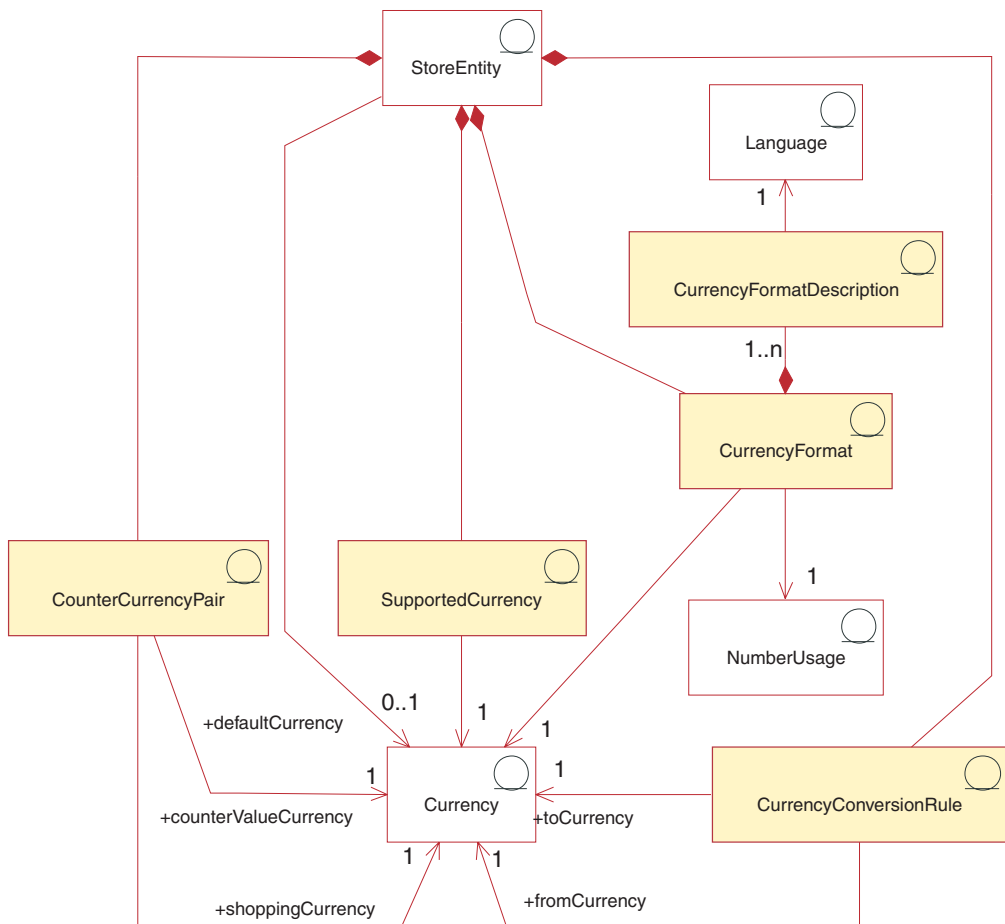
Puede mostrar los precios en su sitio Web en una sola moneda o puede utilizar varias monedas siguiendo las instrucciones que se proporcionan para el euro (consulte “Moneda de contravalor” en la página 237). Para un sitio con múltiples tiendas, puede utilizar monedas diferentes para las tiendas o puede asignar monedas al grupo de tiendas. En función de la naturaleza del sitio que esté creando, puede especificar qué monedas desea utilizar y cómo se visualizarán.

En WebSphere Commerce, puede permitir que los clientes seleccionen una moneda de compra. La moneda de compra es la moneda con la que los clientes pagan los productos en una tienda específica. Todos los importes monetarios de las páginas de la tienda se visualizan en esta moneda. Cuando los clientes cambian la moneda de compra, los precios de los artículos que han añadido a los carros de la compra y los totales de pedido se convierten, se vuelven a calcular y se visualizan automáticamente en la nueva moneda de compra.

Los clientes pueden comprar en muchas monedas, incluido el euro. El euro se convirtió en moneda legal para la Unión Europea el 1 de enero de 1999 y se utiliza actualmente en los mercados financieros. Los tipos de conversión entre el euro y las monedas de todos los países participantes son fijos.

Los elementos de moneda en WebSphere Commerce

El diagrama siguiente muestra la estructura de los elementos de moneda en WebSphere Commerce Server:



Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce Server. Para obtener más información sobre el modelo de información, consulte “Modelo de información de los datos de la tienda” en la página 105. Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, “Descripción de UML”, en la página 463.

En el diagrama anterior, la moneda está en el centro del modelo de información. Cada tienda, o grupo de tiendas, tiene una moneda por omisión.

Formato de moneda

Una entidad de tienda puede tener muchas normas de *formato de moneda*. Si una tienda no tiene una norma de formato para una moneda específica, utiliza la norma de formato de su grupo de tiendas. Los formatos de moneda se establecen en la tabla CURFORMAT.

El elemento de formato de moneda lo pueden utilizar otras tiendas como se describe en el Capítulo 14, “Relaciones entre tiendas”, en la página 143.

Utilización de números

Cada norma de formato de moneda está asociada con una *utilización de números*. Los números como, por ejemplo, los importes monetarios y las cantidades se pueden formatear y redondear de forma diferente según la utilización asociada.

Las tiendas pueden especificar distintas normas de redondeo y formato de los números que se visualizan, según la forma en que se utilicen; por ejemplo, una tienda puede redondear los precios por unidad a cuatro cifras decimales especificando la utilización de precio unitario, y redondear otros importes a dos cifras decimales especificando la utilización por omisión. La utilización de números se almacena en la tabla NUMBRUSG.

Descripción del formato de moneda

Una norma de formato de moneda puede tener muchas *descripciones de formato de moneda*. Una descripción de formato de moneda describe cómo se da formato (para su visualización) a un importe monetario en una moneda e idioma determinados. Cada descripción está asociada a un idioma en la tabla LANGUAGE. Para obtener más información sobre los elementos de idioma consulte el Capítulo 22, “Elementos de idioma”, en la página 233. Para obtener más información, consulte el Capítulo 34, “Globalización”, en la página 321. Las descripciones sobre los formatos de moneda se almacenan en la tabla CURFMTDESC.

Moneda soportada

Una entidad de tienda puede tener muchas *monedas soportadas*. Una moneda soportada es aquella en que se aceptan los pagos.

El elemento de moneda lo pueden utilizar otras tiendas como se describe en el Capítulo 14, “Relaciones entre tiendas”, en la página 143.

Norma de conversión de moneda

Todas las monedas tienen normas que rigen la conversión a y de otras monedas. Cada *norma de conversión de moneda* puede utilizarse para convertir un precio (almacenado en la base de datos en una moneda específica) al importe que se cobrará a los clientes en una moneda de compra soportada.

El elemento de norma de conversión de moneda lo pueden utilizar otras tiendas como se describe en el Capítulo 14, “Relaciones entre tiendas”, en la página 143.

Moneda de contravalor

Las *monedas de contravalor* son importes monetarios que se muestran junto con una moneda soportada. No se pueden utilizar para comprar sino que se utilizan para fines informativos. Si los clientes deciden comprar en euros, podrán visualizar los importes monetarios en euros y en otra moneda en la tienda. Los importes en la moneda de compra se convierten a todas las monedas de contravalor para esa moneda de compra. Las monedas de contravalor están emparejadas con una moneda soportada, como por ejemplo la peseta española y el euro. Las parejas de moneda de contravalor están almacenadas en la tabla CURCVLIST.

El elemento de contravalor de moneda lo pueden utilizar otras tiendas como se describe en el Capítulo 14, “Relaciones entre tiendas”, en la página 143.



Para obtener información más detallada sobre la estructura de los elementos de moneda en WebSphere Commerce Server, consulte el modelo de datos de moneda en la ayuda en línea de WebSphere Commerce.

Creación de elementos de moneda en WebSphere Commerce

La Consola de administración de WebSphere Commerce le permite añadir monedas soportadas a la tienda y seleccionar una moneda por omisión para la tienda. Para obtener más información sobre qué elementos puede editar con la Consola de administración, consulte el tema "*Cambiar los elementos de base de datos de la tienda*" de la ayuda en línea de WebSphere Commerce.

Nota: La Consola de administración trabaja con archivos XML con datos preinsertados, en forma de archivador de tienda.

También puede añadir a la tienda las monedas soportadas y una moneda por omisión mediante un archivo XML que se puede cargar en la base de datos utilizando Loader package. Este método también permite crear otros tipos de elementos de moneda, incluida la definición de las tasas de conversión de monedas y las monedas de contravalor.

Para obtener información acerca de cómo trabajar con monedas, consulte la ayuda en línea para el desarrollo de WebSphere Commerce. Para obtener información sobre cómo crear elementos de moneda nuevos con formato de archivo, consulte "Creación de elementos de moneda utilizando un archivo XML".

Creación de elementos de moneda utilizando un archivo XML

Cree los elementos de moneda de su tienda en formato de archivo XML que se pueda cargar en la base de datos mediante Loader package. Para obtener más información sobre Loader package, consulte la Parte 10, "Publicación de la tienda", en la página 345.

Antes de crear elementos, debe familiarizarse con el material tratado en el Capítulo 38, "Carga de grupos de elementos de base de datos de WebSphere Commerce", en la página 405.

Para crear elementos de moneda para su tienda utilizando un archivo XML, haga lo siguiente:

1. Revise los archivos XML que se utilizan para crear elementos de moneda para las tiendas de ejemplo. Todos los archivos para las tiendas de ejemplo se encuentran en el archivo archivador de tienda correspondiente.

Los archivos de archivador de tienda se encuentran en el directorio siguiente:

- *dir_instalación_WC/samplestores*

Nota: La ayuda en línea de WebSphere Commerce contiene información sobre cada uno de los elementos de datos contenidos en las tiendas de ejemplo.

Cada tienda de ejemplo incluye un archivo *currency.xml*, que contiene la información de moneda. Para ver los archivos *currency.xml* en el archivador de tienda, descomprímalo utilizando un programa ZIP. Los archivos *currency.xml* se encuentran en el directorio de datos.

2. Revise la información del Apéndice B, "Creación de los datos", en la página 465.
3. Cree un archivo *currency.xml*, copiando uno de los archivos *currency.xml* de los archivadores de tienda de ejemplo o creando uno nuevo. Para obtener más información, consulte el archivo *wcs.dtd* en el directorio *dir_instalación_WC/schema/xml* o la DTD incluida en el archivador de tienda.

4. Defina las monedas que soporta la tienda.
 - a. Utilizando el ejemplo siguiente como muestra, defina las monedas soportadas por la tienda en el archivo XML para la tabla CURLIST:

```
<curlist currstr="EUR" storeent_id="@storeent_id_1" />
```

donde:

- currstr es el código de moneda ISO 4217 de tres caracteres que representa la moneda soportada. Este código debe aparecer en la columna SETCCURR de la tabla SETCURRE. Una tienda debe poder aceptar el pago en todas las monedas soportadas.
- storeent_id es la entidad de tienda.

- b. Repita este paso para cada moneda soportada por la tienda.



La moneda por omisión para una tienda se define en la tabla STOREENT. Para obtener más información, consulte “Creación de elementos de datos de tienda en un archivo XML” en la página 136.

5. (Opcional) Los precios y la moneda que aparecen en la tienda dependen del modo en que defina los precios. Puede definir los precios para cada moneda que se utilice en la tienda o puede definirlos solamente para la moneda por omisión. Para obtener más información sobre cómo definir los precios, consulte el apartado “Creación de elementos de fijación de precios en WebSphere Commerce” en la página 191.

Si cuando define los precios solamente define los precios para la moneda por omisión y, sin embargo, desea mostrar los precios de la tienda en otras monedas soportadas, deberá añadir las tasas de conversión de la tienda. Utilice esta tasa de conversión para realizar la conversión desde la moneda por omisión a la moneda soportada.

- a. Determine la moneda desde la que realizará la conversión, por ejemplo, el dólar americano (USD) y la moneda o monedas a las que va a convertir los precios, por ejemplo, el Yen (JPY). Para determinar los códigos de moneda ISO para cada moneda, consulte los códigos ISO 4217 para las monedas internacionales.
 - b. Utilice el ejemplo siguiente como guía y añada la información de conversión a la tabla CURCONVERT:

```
<curconvert
storeent_id="@storeent_id_1"
fromcurr="USD"
tocurr="JPY"
factor="105.10"
multiplyordivide="M"
bidirectional="Y"
updatable="Y"
curconvert_id="@curconvert_id_1" />
```

donde:

- storeent_id es la entidad de tienda.
- fromcurr es la moneda desde la que va a realizar la conversión. Un importe en la moneda FROMCURRE normalmente forma parte de una norma o de otra información que se utiliza para determinar un precio, un descuento, un coste de envío o un importe similar asociado con un producto que se ha puesto en venta.
- tocurr es la moneda a la que va a realizar la conversión. Normalmente, TOCURRE es la moneda en la que el cliente piensa pagar. Normalmente, los importes en esta moneda forman parte de un artículo de pedido como, por ejemplo, un precio unitario, un coste de envío o un importe de impuestos.

- factor es el factor de conversión.
 - multiplyordivide es como se indica a continuación: para realizar la conversión de FROMCURR a TOCURR:
 - M = Multiplicar por FACTOR
 - D = Dividir por FACTOR

Para las normas bidireccionales, se permite la conversión de TOCURR a FROMCURR utilizando la operación inversa.
 - bidirecional indica que la norma es bidireccional o unidireccional:
 - Y = bidireccional
 - N = unidireccional
 - updatable es un distintivo que utiliza una interfaz de usuario que gestiona las normas de conversión de moneda. Valores válidos:
 - N = la tasa de conversión es irrevocable y no debe cambiarse nunca
 - Y = la tasa de conversión se puede cambiar
 - curconvert_id es una clave exclusiva generada.
- c. Repita los pasos a y b para todas las monedas en que desee mostrar los precios.



Incluso si ha definido precios para todas las monedas soportadas en su información de fijación de precios, es posible que desee definir las tasas de conversión de moneda para las monedas que soporta la tienda.

6. (Opcional) si desea mostrar precios en la moneda de compra y en la moneda de contravalor (por ejemplo, para mostrar precios en pesetas y en euros), debe añadir la información a la tabla CURCVLIST.
- a. Utilice el ejemplo siguiente como guía y añada la información de conversión a la tabla CURCVLIST:

```
<curcvlist
storent_id="@storent_id_1"
currstr="NLG"
countervaluecurr="EUR"
displayseq="1" />
```

donde:

- storent_id es la entidad de tienda.
- currstr es el código de moneda ISO 4217 de tres caracteres que representa la moneda. Este código debe aparecer en la columna SETCCURR de la tabla SETCURR y es la moneda a la que realizará la conversión. Un importe en la moneda FROMCURR normalmente forma parte de una norma o de otra información que se utiliza para determinar un precio, un descuento, un coste de envío o un importe similar asociado con un producto que se ha puesto en venta.
- countervaluecurr es el código de moneda ISO 4217 de tres caracteres que representa la moneda de contravalor. Este código debe aparecer en la columna SETCCURR de la tabla SETCURR.
- displayseq es el número que indica el orden de presentación de la moneda de contravalor. Las monedas de contravalor se muestran en orden ascendente según la secuencia de visualización del contravalor que se ha especificado en la columna DISPLAYSEQ de la tabla CURCVLIST.



Para obtener más información sobre cómo utilizar @ y &, consulte el Apéndice B, "Creación de los datos", en la página 465.

Otras tareas de moneda

Para obtener más información sobre la moneda en general y otras tareas de moneda, entre ellas:

- Añadir nuevas monedas que WebSphere Commerce no soporta actualmente
- Cambiar los formatos de las monedas existentes

consulte la ayuda en línea al desarrollo de WebSphere Commerce.

Capítulo 24. Elementos de unidad de medida

Los productos pueden venderse, y se puede hacer un seguimiento del inventario, utilizando diversas unidades de medida, como por ejemplo, kilos, pulgadas, litros, etc. Usando estas unidades, se pueden solicitar productos en cantidades mínimas y por múltiplos de cantidades específicas.

Los mandatos de controlador utilizan el parámetro UOM (unidad de medida) para especificar la unidad de medida. Si no se especifica un parámetro UOM, la cantidad especificada del cliente se multiplica por la cantidad nominal de la entrada de catálogo de la tabla de base de datos CATENTSHIP. El resultado se conoce como la cantidad solicitada.

La cantidad solicitada se redondea por exceso al siguiente múltiplo de cantidad más alto para la entrada de catálogo. Por ejemplo, si el múltiplo es 2 kilos y la cantidad solicitada es 4,1 kilos, el resultado del redondeo será 6 kilos. La cantidad redondeada se utiliza al comprobar el inventario, que tiene su propia unidad de medida. Si la unidad de medida del inventario y la unidad de medida de entrada de catálogo son diferentes, debe efectuarse una conversión entre las dos unidades.

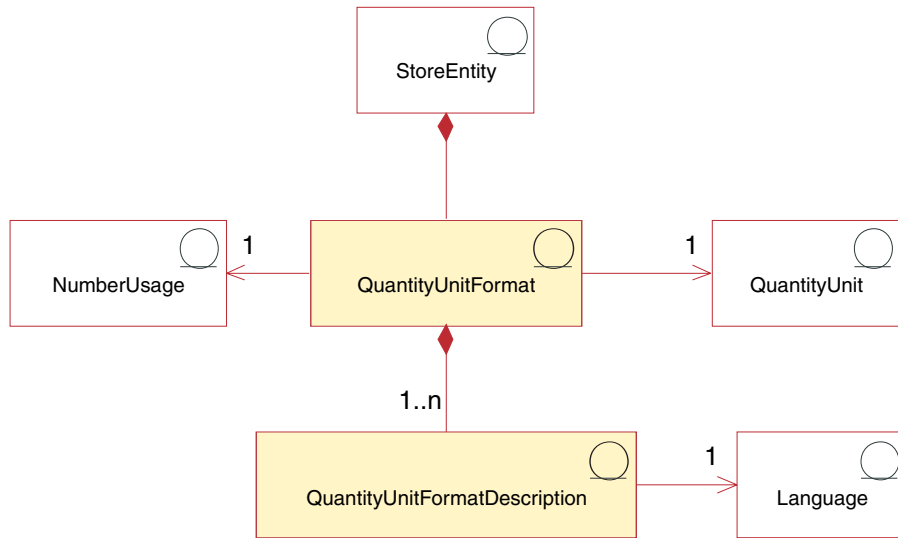
Cuando se habilita el inventario de disponibilidad asegurada (ATP) (consulte la columna ALLOCATIONGOODFOR de la tabla STORE), la unidad de medida de inventario se define en la columna QUANTITYMEASURE de la tabla BASEITEM. De lo contrario, se define en la columna QUANTITYMEASURE de la tabla INVENTORY.

La cantidad redondeada dividida por la cantidad nominal de la entrada de catálogo se conoce como cantidad normalizada. La cantidad normalizada se almacena en el artículo de pedido o el artículo de interés, en función del mandato que se esté ejecutando. Por ejemplo, si la cantidad redondeada es 6 kilos y la cantidad nominal es 2 kilos, la cantidad normalizada es 3.

Cuando se busca una oferta de una entrada de catálogo, la cantidad solicitada puede tener importancia para seleccionar la oferta que proporciona el mejor precio y, por consiguiente, determinar qué oferta se utilizará. Por ejemplo, si la cantidad redondeada es 6 kg y hay dos ofertas, una que especifica un precio de 4,00 euros para la cantidad nominal de 2 kilos y una cantidad mínima de 10 kilos y otra que especifica un precio de 4,50 euros para la cantidad nominal de 2 kilos y una cantidad mínima de 2 kilos, sólo se podrá utilizar la segunda oferta.

Las unidades de medida en WebSphere Commerce

El diagrama siguiente ilustra la estructura de las unidades de medida en WebSphere Commerce Server:



Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce Server. Para obtener más información sobre el modelo de información, consulte “Modelo de información de los datos de la tienda” en la página 105. Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, “Descripción de UML”, en la página 463.

Unidad de cantidad y formato de la unidad de cantidad

Una *unidad de cantidad* es la unidad de medida que se utiliza en la tienda, por ejemplo, kilogramos, libras, metros, pulgadas, litros, etc. El formato de la unidad de cantidad indica cómo se formatea esta unidad de cantidad, por ejemplo, cuántas cifras decimales se utilizan al mostrar la unidad de cantidad.

Cada *formato de unidad de cantidad* forma parte de una sola entidad de tienda, pero cada entidad de tienda puede tener varios formatos de unidad de cantidad.

Cada formato de unidad de cantidad tiene una unidad de cantidad y utilización de números, pero puede tener una o más descripciones de formato de unidad de cantidad, dependiendo del número de idiomas que la tienda soporte.

Otras tiendas pueden utilizar las unidades de cantidad definidas en una tienda. Para que una tienda pueda utilizar las unidades de cantidad definidas en otra tienda, debe crearse una relación entre tiendas de tipo `com.ibm.commerce.measurement`. Para obtener más información, consulte el Capítulo 14, “Relaciones entre tiendas”, en la página 143.

Descripción del formato de unidad de cantidad

La *descripción del formato de unidad de cantidad* describe cómo dar formato (a efectos de visualización) a una cantidad en una unidad de cantidad y un idioma específicos.

Utilización de números

La *utilización de números* define el modo en que se utiliza un número en una aplicación. Por ejemplo, si utiliza códigos de utilización de números en el código de WebSphere Commerce, podrá elegir el modo en que desea que se formatee o redondee dicho número (moneda o cantidad). Estos códigos (definidos en la tabla

NUMBRUSG) permiten formatear el número de acuerdo con las normas especificadas para dicho tipo de utilización de números en las tablas CURFORMAT, CURFMTDESC, QTYFORMAT y QTYFMTDESC. Esto permite a las tiendas formatear los números de varias formas, a fin de satisfacer los requisitos de diversas situaciones.



Para obtener información más detallada sobre la estructura de los elementos de unidad de medida en WebSphere Commerce Server, consulte el modelo de datos de la unidad de cantidad en la ayuda en línea de WebSphere Commerce.

Creación de unidades de medida en WebSphere Commerce

Cuando se crea una instancia, las unidades de medida están ya insertadas en la base de datos de WebSphere Commerce Server. Para obtener más información, consulte el Capítulo 11, "Elementos de sitio", en la página 119.

También puede definir nuevas unidades de medida en WebSphere Commerce para utilizarlas en la tienda o suprimir unidades de medida que no desee utilizar en la tienda.

Para definir nuevas unidades de medida para utilizarlas en la tienda, añada información a las siguientes tablas de base de datos:

- QTYUNIT
- QTUNITDSC
- QTYFORMAT
- QTYFMTDESC
- QTYUNITMAP
- QTYCONVERT

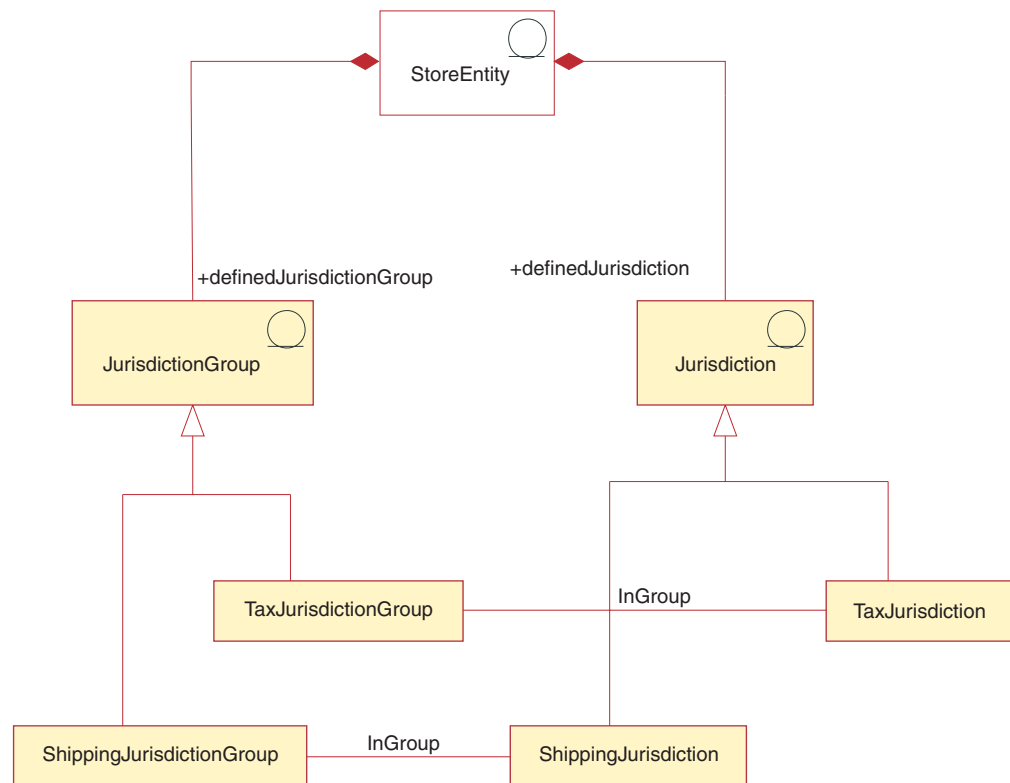
Capítulo 25. Elementos de jurisdicción

Las *jurisdicciones* son regiones o zonas geográficas que representan un país o una región, una provincia o un territorio, o un rango de códigos postales, en los que se venden mercancías. Las jurisdicciones se pueden agrupar para formar *grupos de jurisdicciones*.

Los grupos de jurisdicciones se utilizan en el cálculo de los costes de envío y los impuestos de los pedidos. Es decir, un grupo de jurisdicciones se puede utilizar para calificar las normas de cálculo de costes de envío e impuestos que se utilizan. Estas normas de cálculo calificadas se aplican a los artículos de un pedido solamente si el artículo se va a enviar a una dirección dentro de una de las jurisdicciones de un grupo de jurisdicciones que está asociado con la norma de cálculo. Como resultado, los costes de envío y los impuestos se pueden calcular de forma diferente según las direcciones de envío de los diferentes artículos del pedido.

Los elementos de jurisdicción en WebSphere Commerce

El siguiente diagrama ilustra la relación entre las jurisdicciones y los grupos de jurisdicciones en WebSphere Commerce Server.





Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce Server. Para obtener más información sobre el modelo de información, consulte “Modelo de información de los datos de la tienda” en la página 105. Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, “Descripción de UML”, en la página 463.

En WebSphere Commerce una jurisdicción o un grupo de jurisdicciones es una parte de la tienda y es exclusiva de la tienda para la que se ha creado. Por ejemplo, si crea tres jurisdicciones para una tienda y, posteriormente, suprime la tienda, también se suprimen las jurisdicciones. No las puede utilizar ninguna tienda existente ni ninguna tienda que cree en el futuro.

Sin embargo, si crea jurisdicciones para un grupo de tiendas, las jurisdicciones no se suprimirán cuando se supriman las tiendas de dicho grupo. Las jurisdicciones estarán disponibles para las nuevas tiendas creadas en dicho grupo de tiendas.

WebSphere Commerce da soporte a dos tipos de jurisdicciones: jurisdicciones de envío y jurisdicciones fiscales. Las jurisdicciones de envío se pueden agrupar para formar grupos de jurisdicciones que califican las normas de cálculo de costes de envío. Del mismo modo, las jurisdicciones fiscales se pueden agrupar para formar grupos de jurisdicciones fiscales que calificarán las normas de cálculo de impuestos.



Para obtener información más detallada sobre la estructura de los elementos de jurisdicción en WebSphere Commerce Server, consulte el modelo de datos de jurisdicción en la ayuda en línea de WebSphere Commerce.

Creación de elementos de jurisdicción en WebSphere Commerce

Debe crear elementos de jurisdicción para la tienda para poder aplicar los impuestos y los costes de envío. Para obtener más información sobre la creación de jurisdicciones, consulte “Creación de elementos de impuestos en WebSphere Commerce” en la página 268 o “Creación de elementos de envío en WebSphere Commerce” en la página 251.

Una vez haya creado jurisdicciones para la tienda, puede editarlas o crear nuevas, utilizando los Cuadernos Impuestos y Envíos de las herramientas de tienda de WebSphere Commerce Accelerator.

Nota: Se crea automáticamente un grupo de jurisdicciones para cada jurisdicción creada. Las jurisdicciones se crean para las tiendas pero no para los grupos de tiendas.



Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce Server. Para obtener más información sobre el modelo de información, consulte “Modelo de información de los datos de la tienda” en la página 105. Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, “Descripción de UML”, en la página 463.

Modalidades de envío

La *modalidad de envío* es una forma de enviar las mercancías. Más específicamente, una modalidad de envío es la combinación de una empresa de transportes (que es una empresa que proporciona servicios de envío desde un centro de despacho de pedidos a un cliente) y el servicio de envío ofrecido por dicha empresa de transportes. Por ejemplo, Compañía de transportes ABC, Servicio 12 horas y Compañía de transportes ABC, Entrega urgente son modalidades de envío.

Una modalidad de envío pertenece a una entidad de tienda. Si se suprime la entidad de tienda, también se suprimirán las modalidades de envío definidas en dicha entidad de tienda. No es necesario que una tienda tenga una modalidad de envío por omisión pero es recomendable.

Acuerdos de envío

Un *acuerdo de envío* es un acuerdo entre la tienda y el centro de despacho de pedidos, que indica que un centro de despacho de pedidos enviará mercancías para una tienda determinada utilizando las modalidades de envío especificadas. Se pueden imponer determinadas restricciones en un acuerdo de envío, incluyendo el periodo de tiempo durante el cual el acuerdo de envío está en vigor y las jurisdicciones de envío.

Si un acuerdo de envío está asociado con una modalidad de envío, solamente se aplica a dicha modalidad de envío. De lo contrario, el acuerdo de envío se aplica a todas las modalidades de envío disponibles. Un acuerdo de envío forma parte de una tienda y se suprimirá si se suprime la tienda.

Códigos de cálculo

Los *códigos de cálculo* se utilizan para calcular los costes de envío, es decir, un código de cálculo de costes de envío indica cómo se calculan los costes de envío para los artículos de pedido. Para calcular los costes de envío de un artículo de pedido, debe asignar códigos de cálculo de costes de envío a una entrada de catálogo o a un grupo de entradas de catálogo.

Un código de cálculo forma parte de una entidad de tienda. Un código de cálculo sólo se puede asociar a una entidad de tienda, pero una entidad de tienda puede tener varios códigos de cálculo. Si se suprime la entidad de tienda, también se suprimirán los códigos de cálculo asociados a dicha entidad de tienda.



Para obtener más información acerca de cómo utilizar los códigos de cálculo, consulte la publicación *WebSphere Commerce, Guía de infraestructura de cálculo*.

Normas de cálculo

Cada código de cálculo tiene un conjunto de *normas de cálculo*. Los costes de envío de un artículo de pedido pueden variar en función de la modalidad de envío, del centro de despacho de pedidos y de las jurisdicciones de envío.

ShippingJurisdictionGroupCalculationRules son los objetos de relación que asocian las normas de cálculo de envío con las jurisdicciones, los centros de despacho de pedidos y las modalidades de envío para determinar qué normas de cálculo deben utilizarse para cada artículo de pedido.

Si se suprime la norma de cálculo o cualquiera de los objetos a los que hace referencia ShippingJurisdictionGroupCalculationRules, también se suprimirá ShippingJurisdictionGroupCalculation.



Para obtener más información acerca de cómo utilizar los códigos de cálculo, consulte la publicación *WebSphere Commerce, Guía de infraestructura de cálculo*.

Jurisdicciones y grupos de jurisdicciones

Las *jurisdicciones* son regiones o zonas geográficas que representan un país o una región, una provincia o un territorio, o un rango de códigos postales, en los que se venden mercancías. Las jurisdicciones se agrupan para formar *grupos de jurisdicciones*.

WebSphere Commerce soporta dos tipos de jurisdicciones: jurisdicciones de envío y jurisdicciones fiscales. Cada una de estas jurisdicciones forma parte del grupo correspondiente, por ejemplo, las jurisdicciones de envío están en el grupo de jurisdicciones de envío y las jurisdicciones fiscales están en el grupo de jurisdicciones fiscales.

Los grupos de jurisdicciones se asocian a normas de cálculo. La norma de cálculo utiliza el grupo de jurisdicciones como parte del cálculo para determinar el importe de los costes de envío.

Las jurisdicciones y los grupos de jurisdicciones forman parte de una entidad de tienda. Si se suprime una entidad de tienda, también se suprimirán las jurisdicciones y los grupos de jurisdicciones asociados a dicha entidad de tienda.

Una dirección de envío se puede resolver en varias jurisdicciones de envío. Por ejemplo, una dirección de envío en Nueva York, Estados Unidos, se aplicará a las siguientes jurisdicciones de envío: "Nueva York, Estados Unidos", "Estados Unidos" y el "Mundo". Cuando una dirección de envío se aplica a varias jurisdicciones de envío, se podrán aplicar varias normas de cálculo de envío. En estos casos, se utiliza la prioridad de la ShippingJurisdictionGroupCalculationRules asociada para determinar qué norma o normas se utilizarán.



Para obtener información más detallada sobre la estructura de los elementos de envío en WebSphere Commerce Server, consulte los modelos de datos de envío en la ayuda en línea de WebSphere Commerce.

Creación de elementos de envío en WebSphere Commerce

Las herramientas de envío de WebSphere Commerce Accelerator le permiten crear y editar determinados elementos de envío (por ejemplo modalidades de envío y jurisdicciones), pero no todos los elementos de envío. La lista siguiente describe detalladamente las tablas de base de datos que se pueden editar con las herramientas de envío.

- CALCODE

- CALCODEDSC
- CALRULE
- SHPJCRULE
- CRULESCALE
- CALSCALE
- CALSCALEDSC
- CALRANGE
- CALRLOOKUP
- SHIPMODE
- SHPMODEDSC
- SHPARRANGE
- SHPARJURGP
- JURST
- JURSTGROUP
- JURSTGPREL
- CATENCALCD
- CATGPCALCD

También puede crear los elementos de envío en formato de archivos XML que se pueden cargar en la base de datos utilizando Loader package. De este modo, tiene las dos opciones siguientes para crear elementos de envío:

- Edite los elementos de envío existentes de una de las tiendas de ejemplo que se incluyen con WebSphere Commerce o de un archivador de tienda existente o cree elementos de envío nuevos.
- Cree nuevos elementos de envío con formato de archivo XML

Para obtener información sobre cómo crear y editar elementos de envío utilizando WebSphere Commerce Accelerator, consulte la Ayuda en línea a la producción de WebSphere Commerce. Para obtener información sobre cómo crear elementos de envío nuevos en forma de archivo XML, consulte “Creación de elementos de envío utilizando un archivo XML”.

Creación de elementos de envío utilizando un archivo XML

Cree los elementos de envío en formato de archivos XML que se pueden cargar en la base de datos utilizando Loader package. Para obtener más información sobre Loader package, consulte la Parte 10, “Publicación de la tienda”, en la página 345. Si crea una tienda globalizada, deberá crear archivos XML para cada entorno nacional al que la tienda dé soporte. El archivo específico de entorno nacional debe especificar toda la información de las descripciones, para que se pueda traducir fácilmente. Para obtener más información acerca de cómo crear tiendas globalizadas, consulte el Capítulo 34, “Globalización”, en la página 321.

Las tiendas de ejemplo, de las que se han tomado muchos de los ejemplos de estas tareas, utilizan un archivo `shipping.xml` para toda la información que no necesita traducirse y otro archivo `shipping.xml` para cada entorno nacional soportado por la tienda para la información que necesita traducirse. Los archivos específicos de entorno nacional contienen toda la información de descripción, para que ésta pueda traducirse fácilmente.

Para crear elementos de envío para la tienda utilizando un archivo XML, realice lo siguiente:

1. Consulte la publicación *WebSphere Commerce, Guía de infraestructura de cálculo*. La infraestructura de cálculo de WebSphere Commerce calcula los importes monetarios (por ejemplo, envío) asociados al producto o servicio que un cliente ha seleccionado para adquirir.
2. Revise la información del Capítulo 38, “Carga de grupos de elementos de base de datos de WebSphere Commerce”, en la página 405.
3. Revise los archivos XML utilizados para crear elementos de envío para las tiendas de ejemplo. Todos los archivos de las tiendas de ejemplo se encuentran en el archivador de tienda correspondiente. Cada tienda de ejemplo incluye dos o más archivos `shipping.xml`, que incluyen la información de envío. Los archivos de archivador de tienda están ubicados en el directorio siguiente:
 - `dir_instalación_WC/sampletores`

Nota: La publicación *WebSphere Commerce, Guía de tiendas de ejemplo* contiene información sobre cada uno de los elementos de datos contenidos en las tiendas de ejemplo.

Para ver los archivos `shipping.xml` del archivador de tienda, descomprímalos utilizando un programa ZIP. Los archivos `shipping.xml` se encuentran en el directorio de datos. El archivo `shipping.xml` específico de un idioma está situado en el subdirectorío específico del entorno nacional del directorio de datos.

4. Revise la información del Apéndice B, “Creación de los datos”, en la página 465.
5. Cree un archivo `shipping.xml`, copiando uno de los archivos `shipping.xml` de los archivadores de tienda de ejemplo o creando uno nuevo. Para obtener más información, consulte el archivo `wcs.dtd`. El archivo DTD se encuentra en el siguiente directorio:
 - `dir_instalación_WC/schema/xml`
6. Defina las jurisdicciones y el grupo de jurisdicciones a las que está enviando mercancías o servicios. Todas las jurisdicciones deben pertenecer a un grupo de jurisdicciones.
 - a. Utilizando el ejemplo siguiente como guía, defina un grupo de jurisdicciones en el archivo XML de la tabla JURSTGROUP:


```
<jurstgroup
jurstgroup_id="@jurstgroup_id_1"
description="Grupo de jurisdicciones1 para envío"
subclass="1"
storeent_id="@storeent_id_1"
code="Mundial"/>
```

donde

 - `jurstgroup_id` es una clave exclusiva generada
 - `description` es una breve descripción del grupo de jurisdicciones, adecuada para visualizarse en una interfaz de usuario para la gestión de grupos de jurisdicciones.
 - `subclass` es la subclase del grupo de jurisdicciones tal como se indica a continuación:
 - 1 = `ShippingJurisdictionGroup`
 - 2 = `TaxJurisdictionGroup`
 - `storeent_id` es la entidad de tienda asociada a este grupo de jurisdicciones.

- code que, junto con la entidad de tienda y la subclase, identifica de forma exclusiva este grupo de jurisdicciones.
- b. Utilizando el ejemplo siguiente como guía, defina una jurisdicción en el archivo XML de la tabla JURST.

```
< jurst
jurst_id="@jurst_id_1"
storeent_id="@storeent_id_1"
code="Mundial"
subclass="1"/>
```

donde

- jurst_id es una clave exclusiva generada
 - storeent_id es la entidad de tienda asociada a este grupo de jurisdicciones.
 - code que, junto con la entidad de tienda y la subclase, identifica de forma exclusiva este grupo de jurisdicciones.
 - subclass es la subclase de jurisdicción tal como se indica a continuación:
 - 1 = ShippingJurisdiction
 - 2 = TaxJurisdiction
- c. Utilizando el ejemplo siguiente como guía, asocie la jurisdicción que ha creado en el paso b al grupo de jurisdicciones que ha definido en el paso a, añadiendo información a la tabla JURSTGRPTEL.

```
<jurstgprel
jurst_id="@jurst_id_1"
jurstgroup_id="@jurstgroup_id_1"
subclass="1"/>
```

donde

- jurst_id es la jurisdicción
 - jurstgroup_id es el grupo de jurisdicciones
 - subclass es la subclase de la jurisdicción y del grupo de jurisdicciones. Estas deben coincidir:
 - 1 = ShippingJurisdiction[Group]
 - 2 = TaxJurisdiction[Group]
- d. Repita los pasos a a c para todas las jurisdicciones y todos los grupos de jurisdicciones que la tienda soporte.
7. Defina las modalidades de envío que utilizará la tienda.

- a. Utilizando el ejemplo siguiente como guía, defina una modalidad de envío en el archivo XML para la tabla SHIPMODE:

```
<shipmode
shipmode_id="@shipmode_id_1"
field1
storeent_id="@storeent_id_1"
code="1 semana por correo normal"
carrier="Empresa de transportes XYZ"/>
```

donde:

- shipmode_id es una clave exclusiva generada.
- field1 es un campo que se puede personalizar.

- storeent_id es la entidad de tienda asociada con esta modalidad de envío.
 - code es el código asignado por el comerciante, exclusivo para la entidad de tienda.
 - carrier es el nombre o identificador de la empresa de transportes.
- b. Utilizando el ejemplo siguiente como guía, añade información acerca de la modalidad de envío a la tabla SHPMODEDSC. Si está creando una tienda multicultural, deberá incluir esta información en un archivo XML específico de entorno nacional:
- ```
< shpmodedsc
description="Correo internacional"
field1="5 euros por pedido más 1 euro por cada artículo"
field2="5 días laborables"
shipmode_id="@shipmode_id_1"
language_id="&es_ES;" />
```
- donde:
- description es una breve descripción de la Modalidad de envío, adecuada para mostrarla a un cliente para que la seleccione.
  - field1 y field2 son campos disponibles que se pueden personalizar.
  - shipmode\_id es una clave exclusiva generada.
  - language\_id es el idioma utilizado.
- c. Repita los pasos a y b para todas las modalidades de envío de la tienda.
8. Defina los códigos de cálculo que deberá utilizar la tienda.
- a. Utilizando los ejemplos siguientes como guía, defina el código de cálculo en el archivo XML para la tabla CALCODE.
- ```
< calcode
calcode_id="@calcode_id_1"
code="Código de envío 1- por/pedido"
calusage_id="-2"
storeent_id="@storeent_id_1"
groupby="0"
published="1"
sequence="+0.00E+000"
calmethod_id="-23"
calmethod_id_app="-24"
calmethod_id_qfy="-22"
flags="0" />
```
- donde:
- calcode_id es una clave exclusiva generada.
 - code es una serie de caracteres que identifica de forma exclusiva este Código de cálculo, dado un uso de cálculo y una entidad de tienda determinados.
 - calusage_id indica la clase de cálculo para la que se utiliza este código de cálculo. Por ejemplo, el código de cálculo puede utilizarse para calcular uno de los siguientes importes monetarios:
 - descuentos (-1)
 - costes de envío (-2)

- impuestos sobre la venta (-3)
 - impuestos sobre el envío (-4)
 - cupones (-5)
 - `storeent_id` es la entidad de tienda asociada a este código de cálculo.
 - `groupby` son distintivos de bit que indican al método `CalculationCodeCombineMethod` cómo deben agruparse los artículos de pedido al realizar cálculos. 0 = Sin agrupación. Coloque todos los artículos de pedido aplicables en un solo grupo. Consulte *Tabla CALCODE: detalles* en la ayuda en línea de WebSphere Commerce para obtener más información.
 - `published` especifica si el código de cálculo se publica o no:
 - 0 = No publicado (inhabilitado temporalmente)
 - 1 = Publicado
 - 2 = Marcado para suprimirse (y no publicado)
 - `sequence`: Los códigos de cálculo se calculan y se aplican en secuencia, del más bajo al más alto. Si dos códigos de cálculo tienen el mismo número de secuencia, se calcularán primero los códigos de cálculo cuyo `calcode_id` sea menor.
 - `calmethod_id` es el método `CalculationCodeCalculateMethod` que define cómo calcular un importe monetario para este código de cálculo. `calmethod_id="-23"`, el método `CalculationCodeCalculateMethod` para envío, es el único método de cálculo de envío proporcionado con WebSphere Commerce.
 - `calmethod_id_app` es el método `CalculationCodeApplyMethod` que almacena el importe calculado para los artículos de pedido asociados. `calmethod_id_app="-24"`, el método `CalculationCodeApplyMethod` para envío, es el único método de aplicación de envío proporcionado con WebSphere Commerce.
 - `calmethod_id_qfy` es el método `CalculationCodeQualifyMethod` que define qué artículos de pedido se asocian a este código de cálculo. `calmethod_id_qfy="-22"`, el método `CalculationCodeQualifyMethod` para envío, es el único método de calificación de envío proporcionado con WebSphere Commerce.
 - `flags` especifica si se debe invocar el método `CalculationCodeQualifyMethod` de este código de cálculo.
 - 0 = no restringido. El método no se invocará
 - 1 = restringido. El método se invocará.
- b. Utilizando el ejemplo siguiente como guía, añada la información de descripción de código de cálculo al archivo XML para la tabla `CALCODEDSC`. Si está creando una tienda globalizada, deberá incluir esta información en un archivo XML específico del entorno nacional.

```
<calcodedsc
  calcode_id="@calcode_id_3"
  description="5 euros por pedido"
  language_id="&es_ES"
  longdescription="Este código de cálculo de envío cobra
  5 euros por pedido."
/>
```

donde

- `calcode_id` es el código de cálculo al que se aplica esta información.
- `description` es una descripción breve del código de cálculo.
- `language_id` es el idioma para el que se aplica esta información.

- longdescription es la descripción detallada del código de cálculo.
 - c. Repita los pasos a y b para cada código de cálculo utilizado en la tienda.
9. Defina las normas de cálculo para la tienda.
- a. Utilizando el ejemplo siguiente como guía, configure la norma de cálculo en el archivo XML para la tabla CALRULE:

```
<calrule
calrule_id="@calrule_id_1"
calcode_id="@calcode_id_1"
startdate="1900-01-01 00:00:00.000000"
enddate="2100-01-01 00:00:00.000000"
sequence="+1.000000000000000E+000"
combination="2"
calmethod_id="-27"
calmethod_id_qfy="-26"
flags="1"
identifier="1" />
```

- calrule_id es un identificador exclusivo generado.
- calcode_id es el código de cálculo del que forma parte esta norma de cálculo.
- startdate es el momento en el que entra en vigor esta norma de cálculo.
- enddate es la hora en que esta norma de cálculo deja de estar en vigor.
- sequence es el orden en el que se procesará esta norma de cálculo. Las normas de cálculo para el mismo código de cálculo se procesan en secuencia del valor más bajo al más alto.
- combination especifica el distintivo de bit para indicar el proceso especial que deberá realizar la implementación por omisión de CalculationRuleCombineMethod. Consulte la tabla CALRULE en la ayuda en línea de WebSphere Commerce para obtener más información.
- calmethod_id es el método CalculationRuleCalculateMethod que calcula un resultado monetario para un conjunto de artículos de pedido.
- calmethod_id_qfy es el método CalculationRuleQualifyMethod que determina qué artículo de pedido de un conjunto de artículos de pedido debe enviarse al método CalculationRuleCalculateMethod.
- flags: los utiliza el método CalculationRuleCombineMethod para determinar cómo se puede combinar esta norma de cálculo con otras normas de cálculo. Consulte la tabla CALRULE para obtener más información.
- identifier identifica la norma de cálculo, en combinación con el código de cálculo de la misma.

Para obtener más información, consulte la tabla CALRULE en la ayuda en línea de WebSphere Commerce.

- b. Repita el paso a para cada norma de cálculo utilizada en la tienda. Tenga en cuenta que cada código de cálculo puede tener varias normas de cálculo. Por ejemplo, calcode_id="@calcode_id_1" puede estar asociado a varios calrule_id.
10. Defina escalas de cálculo para la tienda.
- Una escala de cálculo es el conjunto de rangos que se aplicarán al cálculo. Por ejemplo, para los costes de envío puede tener un conjunto de rangos de peso

que correspondan cada uno a un coste determinado. Es decir, un producto que pese entre 0 y 5 kg puede tener un coste de envío de 10 euros. Y un producto que pese entre 5 y 10 kg puede tener un coste de envío de 15 euros. Estos rangos crean una escala.

- a. Utilizando el ejemplo siguiente como guía, configure la escala de cálculo en el archivo XML para la tabla CALSCALE:

```
<calscale  
  calscale_id="@calscale_id_1"  
  code="Código de escala 1 por pedido EUR"  
  storeent_id="@storeent_id_1"  
  calusage_id="-2"  
  setccurr="EUR"  
  calmethod_id="-28"/>
```

donde

- `calscale_id` es un identificador exclusivo generado.
- `code` es una serie de caracteres que identifica de forma exclusiva esta escala de cálculo, dado un uso de cálculo y una entidad de tienda determinados.
- `storeent_id` es la entidad de tienda de la que forma parte esta escala de cálculo.
- `calusage_id` indica la clase de cálculo para la que se utiliza esta escala de cálculo. Por ejemplo, la escala de cálculo puede utilizarse para calcular uno de los importes monetarios siguientes:
 - descuentos (-1)
 - costes de envío (-2)
 - impuestos sobre la venta (-3)
 - impuestos sobre el envío (-4)
 - cupones (-5)
- `setccurr`, si se especifica, indica la moneda para los valores de inicio de rango de los objetos de rango de cálculo para esta escala de cálculo. El método `CalculationScaleLookupMethod` debe devolver un "número de búsqueda" en esta moneda.
- `calmethod_id` es el método `CalculationScaleLookupMethod` que, dado un conjunto de artículos de pedido, determina un valor de búsqueda, un valor monetario base, un multiplicador de resultado y un conjunto de pesos matemáticos que la escala de cálculo puede utilizar para calcular el importe monetario. Para determinar qué método `CalculationScaleLookupMethod` se debe utilizar, realice lo siguiente:
 - Consulte la tabla CALMETHOD en la ayuda en línea de WebSphere Commerce. Consulte la descripción de la columna SUBCLASS. Pulse el enlace para la tabla CALMETHOD: detalles. Esta tabla lista los tipos de métodos de cálculo disponibles. El método `MonetaryCalculationScaleLookupMethod` es 9.
 - Abra el archivo de rutina de carga `wcs.bootstrap_xx_XX.xml`, donde `xx_XX` es el código del entorno nacional. Los archivos de rutina de carga se encuentran en el directorio siguiente:
 - `dir_instalación_WC/schema`
 - Localice la sección que lista los métodos de cálculo disponibles (CALMETHOD).

- Localice los métodos de cálculo con el valor `calusage_ID` para impuestos (-3 para impuestos sobre la venta y -4 para impuestos sobre el envío).
- Localice los métodos de cálculo, que tienen una subclase de 7; hay varios. Seleccione uno que satisfaga sus necesidades.

Para obtener más información, consulte la tabla CALSCALE en la ayuda en línea de WebSphere Commerce.

- b. Repita el paso a para cada escala de cálculo utilizada en la tienda. Por ejemplo, para el envío, FashionFlow crea una escala de coste por pedido y una escala de coste por artículo.

11. Defina rangos de cálculo para las escalas de cálculo.

- a. Utilizando el ejemplo siguiente como guía, configure el rango de cálculo en el archivo XML para la tabla CALRANGE.

```
<calrange
calrange_id="@calrange_id_1"
calscale_id="@calscale_id_1"
calmethod_id="-33"
rangestart="0.00000"
cumulative="0"/>
```

donde

- `calrange_id` es un identificador exclusivo generado.
- `calscale_id` es la escala de cálculo de la que forma parte este rango de cálculo.
- `calmethod_id` es el método `CalculationRangeMethod` que determina un importe monetario a partir de `CalculationRangeLookupResult`. Por ejemplo, `FixedAmountCalculationRangeCmd`, `PerUnitAmountCalculationRangeCmd` o `PercentageCalculationRangeCmd`. Para determinar el método `CalculationRangeMethod`, realice lo siguiente:
 - Consulte la tabla CALMETHOD en la ayuda en línea de WebSphere Commerce. Consulte la descripción de la columna SUBCLASS. Pulse el enlace para la tabla CALMETHOD: detalles. Esta tabla lista los tipos de métodos de cálculo disponibles. El método `CalculationRangeMethod` es 10.
 - Abra el archivo de rutina de carga `wcs.bootstrap_xx_XX.xml`, donde `xx_XX` es el código del entorno nacional. Los archivos de rutina de carga se encuentran en el directorio siguiente:
 - `dir_instalación_WC/schema`
 - Localice la sección que lista los métodos de cálculo disponibles (CALMETHOD).
 - Localice los métodos de cálculo con el valor `calusage_ID` para impuestos (-3 para impuestos sobre la venta y -4 para impuestos sobre el envío).
 - Localice los métodos de cálculo que tienen una subclase de 9; hay varios. Seleccione uno que satisfaga sus necesidades.
- `cumulative` son los valores válidos:
 - 0 = sólo se utiliza el rango de cálculo coincidente que tiene el valor RANGESTART más alto.
 - 1 = se utilizan todos los rangos de cálculo coincidentes. Los importes monetarios calculados se suman para llegar al resultado final.

Para obtener más información, consulte la tabla CALRANGE en la ayuda en línea de WebSphere Commerce.

- b. Repita el paso a para cada rango de cálculo asociado a la escala de cálculo utilizada en la tienda.
12. Defina los valores de búsqueda de cálculo para las escalas de cálculo. Los valores de búsqueda de cálculo son los valores asociados a la escala de cálculo. Por ejemplo, una escala de cálculo incluye los rangos de peso y los precios asociados siguientes para el envío:
 - De 0 a 5 kg cuesta 10 euros
 - De 5 a 10 kg cuesta 15 euros

Los valores de búsqueda son 10 euros y 15 euros.

- a. Utilizando como guía los ejemplos siguientes, configure valores de búsqueda de cálculo en el archivo XML para la tabla CALRLOOKUP. Si está creando una tienda multicultural, deberá incluir esta información en un archivo XML específico de entorno nacional, es decir, un archivo por cada entorno nacional que soporte la tienda. Por ejemplo, si la tienda realiza envíos a los clientes de España y EE.UU., deberá añadir los valores de búsqueda de euro en un archivo XML y los valores de búsqueda de dólar americano en otro archivo XML.

```
<calrlookup  
calrlookup_id="@calrlookup_id_1"  
setcurr="EUR"  
calrange_id="@calrange_id_1"  
value="500"/>
```

donde

- calrlookup_id es un identificador exclusivo generado.
- calrange_id es el rango de cálculo del que forma parte este resultado de búsqueda de rango de cálculo.
- value es el valor del resultado de búsqueda de rango de cálculo, utilizado por el método del rango de cálculo para determinar un resultado monetario.

Para obtener más información, consulte la tabla CALRLOOKUP en la ayuda en línea de WebSphere Commerce.

- b. Repita el paso a para cada valor de búsqueda asociado a la escala de cálculo utilizada en la tienda.
13. Asocie la norma de cálculo y la escala de cálculo

- a. Utilizando los ejemplos siguientes como guía, asocie la escala de cálculo a la norma de cálculo en el archivo XML para la tabla CRULESCALE.

```
< crulescale  
calrule_id="@calrule_id_1"  
calscale_id="@calscale_id_1" />
```

donde

- calrule_id es la norma de cálculo.
- calscale_id es la escala de cálculo.

- b. Repita el paso a para cada escala de cálculo y asociación de norma.



Para obtener más información sobre cómo utilizar @ y &, consulte el Apéndice B, “Creación de los datos”, en la página 465.

Creación de elementos de despacho de envío

Para que los elementos de envío funcionen correctamente en la tienda, deberá asociar los grupos de jurisdicciones de envío con las normas de cálculo y los centros de despacho de pedidos con las modalidades de envío utilizadas en la tienda.

Deberá crear los elementos de despacho para poder asociar los elementos de envío a un centro de despacho de pedidos. Para obtener más información sobre cómo crear elementos de despacho, consulte “Creación de elementos de despacho de pedidos en WebSphere Commerce” en la página 218.

Después de haber creado los elementos de despacho de pedidos, asocie a éstos los elementos de envío añadiendo información en las tablas SHPJCRULE y SHPARRANGE. Efectúe lo siguiente:

1. Consulte la publicación *WebSphere Commerce, Guía de infraestructura de cálculo*. La infraestructura de cálculo de WebSphere Commerce calcula los importes monetarios (por ejemplo, envío) asociados al producto o servicio que un cliente ha seleccionado para adquirir.
2. Revise la información del Capítulo 38, “Carga de grupos de elementos de base de datos de WebSphere Commerce”, en la página 405.
3. Revise los archivos XML utilizados para crear elementos de despacho de envío para las tiendas de ejemplo. Todos los archivos para las tiendas de ejemplo se encuentran en el archivo archivador de tienda correspondiente. Cada tienda de ejemplo incluye un archivo `shipfulfill.xml`, que incluye la información de despacho de envío. Para ver el archivo `shipfulfill.xml` del archivador de tienda, descomprímalo utilizando un programa ZIP. El archivo `shipfulfill.xml` está ubicado en el directorio de datos.

Los archivos de archivador de tienda están ubicados en el directorio siguiente:

- `dir_instalación_WC/samplestores`

Nota: La ayuda en línea de WebSphere Commerce contiene información sobre cada uno de los elementos de datos contenidos en las tiendas de ejemplo.

4. Revise la información del Apéndice B, “Creación de los datos”, en la página 465.
5. Cree un archivo `shipfulfill.xml`, copiando uno de los archivos `shipfulfill.xml` de los archivadores de tienda de ejemplo o creando uno nuevo. Para obtener más información, consulte el archivo `wcs.dtd`. Los archivos DTD se encuentran en el directorio siguiente:
 - `dir_instalación_WC/schema/xml`
6. Asocie normas de cálculo a un grupo de jurisdicciones de envío añadiendo información en la tabla SHPJCRULE. Utilice como guía el ejemplo siguiente. Si está creando una tienda multicultural, cree también un archivo XML para cada entorno nacional que la tienda soporte.

```
<shpjcrule  
calrule_id="@calrule_id_1"  
ffmcenter_id="@ffmcenter_id_1"
```

```
jurstgroup_id="@jurstgroup_id_1"
precedence="0"
shipmode_id="@shipmode_id_1"
shpjcrule_id="@shpjcrule_id_1"
```

donde

- calrule_id es la norma de cálculo utilizada.
 - ffmcenter_id es el centro de despacho de pedidos. Si es NULL, esta asociación se aplica a todos los centros de despacho de pedidos.
 - jurstgroup_id es el grupo de jurisdicciones de envío. Si es NULL, esta asociación se aplica a todos los grupos de jurisdicciones de envío.
 - precedence se aplica cuando una dirección de envío está en más de uno de los grupos de jurisdicciones de envío especificados para el mismo centro de despacho de pedidos y la misma modalidad de envío. Sólo es apta la norma de cálculo que tiene el valor de SHPJCRULE.PRECEDENCE más alto.
 - shipmode_id es la modalidad de envío.
 - shpjcrule_id es un identificador exclusivo generado.
7. Repita el paso 3 para cada grupo de jurisdicciones, centro de despacho de pedidos y asociación de normas de la tienda.
 8. Asocie la modalidad de envío y un centro de despacho de pedidos a la tienda, añadiendo información en la tabla SHPARRANGE. Utilice el ejemplo siguiente como guía:

```
<shparrange
shparrange_id="@shparrange_id_2"
store_id="@storeent_id_1"
ffmcenter_id="@ffmcenter_id_1"
shipmode_id= "@shipmode_id_2"
startdate="1970-06-22 23:00:00.000000"
enddate= "2008-06-22 23:00:00.000000"
precedence= "0"
flags="0"
/>
```

donde

- shparrange_id es un identificador exclusivo generado.
 - store_id es la tienda.
 - ffmcenter_id es el centro de despacho de pedidos.
 - shipmode_id es la modalidad de envío. NULL indica que este acuerdo de envío puede utilizarse independientemente de la modalidad de envío.
 - startdate es el momento en que este acuerdo de envío empieza a estar en vigor.
 - enddate es el momento en que este acuerdo de envío deja de estar en vigor.
 - precedence se aplica cuando en un momento determinado están en vigor más de un acuerdo de envío (para la misma tienda y modalidad de envío); se utiliza el que tiene la prioridad (PRECEDENCE) más alta.
 - flags contiene distintivos de bit:
 - 1 = restringido - Este acuerdo de envío sólo se aplica a artículos de pedido cuya dirección de envío coincide con uno de los grupos de jurisdicciones de envío asociados (mediante la tabla SHPARJURGP) con este acuerdo de envío.
9. Repita el paso 5 para todas las modalidades de envío utilizadas en la tienda.



Para obtener más información sobre cómo utilizar @ y &, consulte el Apéndice B, “Creación de los datos”, en la página 465.

Creación de elementos de tienda-catálogo-envío

Para asociar modalidades de envío con la tienda, deberá asociar un código de cálculo con las entradas de catálogo de la tienda para cada contrato que la tienda incluya.

Para poder crear elementos de tienda-catálogo-envío deberá crear elementos de tienda y de catálogo. Para obtener más información sobre cómo crear elementos de tienda, consulte “Creación de elementos de datos de tienda en un archivo XML” en la página 136. Para obtener más información sobre cómo crear elementos de catálogo, consulte “Visualización de los elementos de catálogo de la tienda” en la página 177.

Para crear elementos de tienda-catálogo-envío, realice lo siguiente:

1. Consulte la publicación *WebSphere Commerce, Guía de infraestructura de cálculo*. La infraestructura de cálculo de WebSphere Commerce calcula los importes monetarios (por ejemplo, envío) asociados al producto o servicio que un cliente ha seleccionado para adquirir.
2. Revise la información del Capítulo 38, “Carga de grupos de elementos de base de datos de WebSphere Commerce”, en la página 405.
3. Revise los archivos XML utilizados para crear elementos de despacho de envío para las tiendas de ejemplo. Todos los archivos para las tiendas de ejemplo se encuentran en el archivo archivador de tienda correspondiente.

Los archivos de archivador de tienda están ubicados en el directorio siguiente:

- *dir_instalación_WC/samplestores*

Nota: La ayuda en línea de WebSphere Commerce contiene información sobre cada uno de los elementos de datos contenidos en las tiendas de ejemplo.

Cada tienda de ejemplo incluye un archivo *store-catalog-shipping.xml*, que incluye la información de despacho de envío. Para ver el archivo *store-catalog-shipping.xml* del archivador de tienda, descomprímalo utilizando un programa ZIP. El archivo *store-catalog-shipping.xml* está ubicado en el directorio de datos.

4. Revise la información del Apéndice B, “Creación de los datos”, en la página 465.
5. Cree un archivo *store-catalog-shipping.xml*, copiando uno de los archivos *store-catalog-shipping.xml* de los archivadores de tienda de ejemplo o bien creando uno nuevo. Para obtener más información, consulte el archivo *wcs.dtd*. Los archivos DTD se encuentran en el directorio siguiente:
 - *dir_instalación_WC/schema/xml*
6. Cree la relación de tienda-catálogo-envío añadiendo información a la tabla CATENCALCD. Utilice el ejemplo siguiente como guía:

```
<catencalcd
  calcode_id="@calcode_id_1"
  catencalcd_id="@catencalcd_id_1"
  store_id="@storeent_id_1"
/>
```

donde

- calcode_id es el código de cálculo.
- catencalcd_id es un identificador exclusivo generado.
- store_id es la tienda.



Para obtener más información sobre cómo utilizar @ y &, consulte el Apéndice B, "Creación de los datos", en la página 465.

Creación de una modalidad de envío por omisión

Para establecer una modalidad de envío por omisión para la tienda, deberá añadir información en la tabla STOREDEF. Para añadir información a la tabla STOREDEF, realice lo siguiente:

1. Revise la información del Capítulo 38, "Carga de grupos de elementos de base de datos de WebSphere Commerce", en la página 405.
2. Revise los archivos XML utilizados para crear elementos por omisión de tienda para las tiendas de ejemplo. Todos los archivos para las tiendas de ejemplo se encuentran en el archivo archivador de tienda correspondiente.

Los archivos de archivador de tienda están ubicados en el directorio siguiente:

- *dir_instalación_WC/samplestores*

Nota: La ayuda en línea de WebSphere Commerce contiene información sobre cada uno de los elementos de datos contenidos en las tiendas de ejemplo.

Cada tienda de ejemplo incluye un archivo store-defaults.xml, que incluye la información de envío por omisión. Para ver el archivo store-defaults.xml del archivador de tienda, descomprímalo utilizando un programa ZIP. El archivo store-defaults.xml está ubicado en el directorio de datos.

3. Revise la información del Apéndice B, "Creación de los datos", en la página 465.
4. Cree un archivo store-defaults.xml, copiando uno de los archivos store-defaults.xml de los archivadores de tienda de ejemplo o bien creando uno nuevo. Para obtener más información, consulte el archivo wcs.dtd. Los archivos DTD se encuentran en el directorio siguiente:

- *dir_instalación_WC/schema/xml*

5. Utilizando como guía el ejemplo siguiente, en el archivo XML, especifique la modalidad de envío por omisión para la tienda añadiendo información en la tabla STOREDEF:

```
<storedef
  store_id="@storeent_id_1"
  shipmode_id="@shipmode_id_1"
/>
```

donde

- store_id es la tienda.
- shipmode_id es la modalidad de envío por omisión para la tienda.



Para obtener más información sobre cómo utilizar @ y &, consulte el Apéndice B, "Creación de los datos", en la página 465.

Capítulo 27. Elementos de impuestos

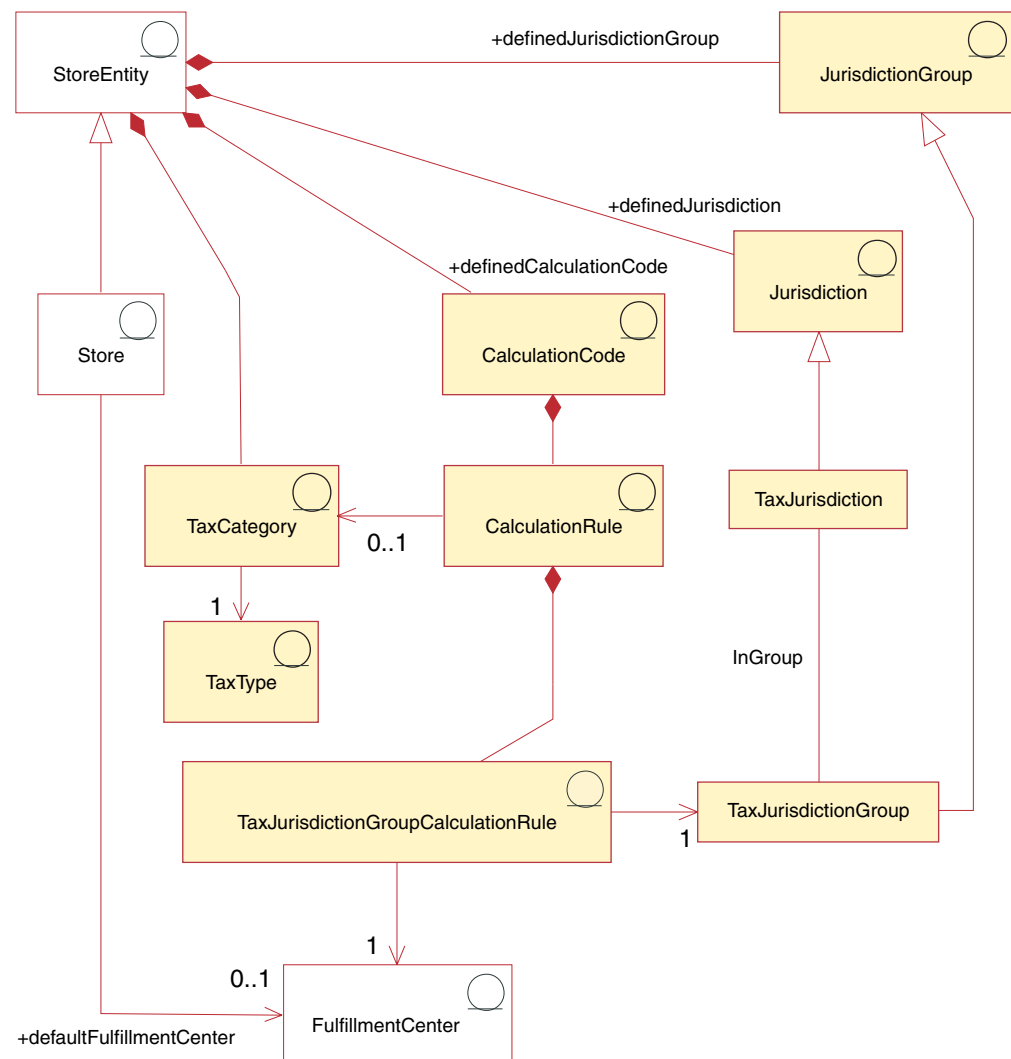
Para cargar o recaudar impuestos sobre las mercancías y los servicios que la tienda proporciona, una tienda creada con WebSphere Commerce debe incluir lo siguiente:

- Categorías de impuestos
- Códigos de cálculo
- Jurisdicciones y grupos de jurisdicciones

La combinación de las categorías de impuestos, los códigos de cálculo y las jurisdicciones y grupos de jurisdicciones crea las tarifas de impuestos para la tienda.

Los elementos de impuestos en WebSphere Commerce

El diagrama siguiente ilustra la estructura de fijación de impuestos en WebSphere Commerce Server.





Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce Server. Para obtener más información sobre el modelo de información, consulte “Modelo de información de los datos de la tienda” en la página 105. Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, “Descripción de UML”, en la página 463.

Categoría de impuestos

Las *categorías de impuestos* corresponden a las diferentes clases de impuestos que una tienda puede tener que recaudar, por ejemplo federales, estatales, provinciales y municipales.

Una categoría de impuestos forma parte de una entidad de tienda, aunque una entidad de tienda pueda tener varias categorías de impuestos. Si se suprime la entidad de tienda, también se suprimirán las categorías de impuestos asociadas a dicha entidad de tienda.

Tipo de impuestos

Una tienda recauda normalmente dos tipos de impuestos: impuestos sobre las ventas o el uso, e impuestos sobre el envío. Cada categoría de impuestos tiene un *tipo de impuesto*. Aunque cada categoría de impuestos puede ser sólo de un tipo de impuesto, (por ejemplo, la categoría de impuestos federal es un tipo de impuestos sobre la venta), varias categorías de impuestos diferentes pueden pertenecer al mismo tipo de impuesto (por ejemplo, el tipo de impuestos sobre la venta se aplica a las categorías federal, provincial y municipal).

Código de cálculo

Los *códigos de cálculo* se utilizan para calcular el importe de los impuestos, es decir, un código de cálculo de impuestos indica cómo se calculan los impuestos para los artículos de pedido. Para calcular el impuesto sobre un artículo del pedido, debe asignar códigos de cálculo de impuestos sobre la venta y de impuestos sobre el envío a una entrada de catálogo o a un grupo de entradas de catálogo. Sólo se puede aplicar un código de cálculo de impuestos de cada tipo de impuesto a una entrada de catálogo determinada o a un grupo de entradas de catálogo. Normalmente, los impuestos sobre las ventas o el uso se calculan sobre el precio neto y los impuestos sobre el envío se calculan sobre los costes de envío.

Un código de cálculo forma parte de una entidad de tienda. Un código de cálculo sólo se puede asociar a una entidad de tienda, pero una entidad de tienda puede tener varios códigos de cálculo. Si se suprime la entidad de tienda, también se suprimirán los códigos de cálculo asociados a dicha entidad de tienda.



Para obtener más información sobre el uso de códigos de cálculo, consulte la publicación *IBM WebSphere Commerce, Guía de infraestructura de cálculo*.

Normas de cálculo

Todo código de cálculo tiene como mínimo una *norma de cálculo*, que define los cálculos de cada categoría de impuesto y especifica las condiciones bajo las que se realizarán los cálculos. Cada norma de cálculo de impuestos está asociada a una categoría de impuesto, un grupo de jurisdicciones y un centro de despacho de pedidos, que juntos definen las condiciones bajo las cuales se utilizará la norma de cálculo. Por ejemplo, se puede seleccionar una norma diferente para calcular un

importe para una categoría de impuestos determinada dependiendo de la dirección de envío y del centro de despacho de pedidos especificados en el pedido.

Cada norma de cálculo pertenece exactamente a un código de cálculo.

Un código de cálculo de impuestos determinado puede tener varias normas de cálculo, una para cada combinación de categoría de impuestos, grupo de jurisdicciones fiscales y centro de despacho de pedidos asociado a la tienda. Todas las normas de cálculo de impuestos sobre la venta y de impuestos sobre el envío se pueden asociar con varias TaxJurisdictionGroupCalculationRules (TaxRules). Por ejemplo en el diagrama siguiente, la norma de cálculo 10001 se puede aplicar a los grupos de jurisdicciones 1234 y 1235.

TAXJCRULE_ID	CALRULE_ID	FFMCENTER_ID	JURSTGROUP_ID	PRECEDENCE
10001	10001	NULL	1234	0
10002	10001	NULL	1235	0

Cada TaxRule define las condiciones bajo las cuales se aplicará la norma de cálculo. Por ejemplo, puede definir una norma de cálculo para cada grupo de jurisdicciones a los que la tienda realiza envíos. En el ejemplo siguiente, la norma de cálculo 10001 se puede aplicar a los grupos de jurisdicciones 1234 y 1235.

En el ejemplo siguiente, el código de cálculo de impuestos utiliza la norma de cálculo A para la categoría provincial de impuestos sobre la venta, cuando la jurisdicción fiscal es Alberta y la norma C cuando la jurisdicción fiscal es British Columbia.

Jurisdicción fiscal	Impuestos sobre la venta federales	Impuestos sobre la venta provinciales
Alberta, Canadá	Norma de cálculo B, que da Y%	Norma de cálculo A, que da X%
British Columbia, Canadá	Norma de cálculo B, que da Y%	Norma de cálculo C, que da Z%

Cuando una dirección de envío coincide con más de un grupo de jurisdicciones fiscales, se utiliza la norma de cálculo con el valor de columna TAXJCRULE.PRECEDENCE asociada más alto.

La asociación de TaxJurisdictionGroupCalculationRules (TaxRule) con una norma de cálculo determina cuándo se puede aplicar la norma de cálculo. Una norma de cálculo de impuestos sobre la venta o de impuestos sobre el envío es aplicable cuando se satisfacen cualquiera de las condiciones determinadas por TaxRules. En el ejemplo siguiente, la norma de cálculo 10001 es aplicable cuando el envío se realiza al grupo de jurisdicciones 1001 o cuando el envío se realiza desde el centro de despacho de pedidos 1001 o si está realizando el envío al grupo de jurisdicciones 1001.

CALRULE_ID	FFMCENTER_ID	JURSTGROUP_ID
10001	NULL	1001
10001	1001	1001

Toda TaxJurisdictionGroupCalculationRule está asociada con un grupo de jurisdicciones como mínimo. Las normas de cálculo por sí mismas no están

asociadas directamente con los grupos de jurisdicciones.



Para obtener más información sobre el uso de normas de cálculo, consulte la publicación *IBM WebSphere Commerce, Guía de infraestructura de cálculo*.

Jurisdicciones y grupos de jurisdicciones

Las *jurisdicciones* son regiones o zonas geográficas que representan un país o una región, una provincia o un territorio, o un rango de códigos postales, en los que se venden mercancías. Las jurisdicciones se agrupan para formar *grupos de jurisdicciones*.

WebSphere Commerce soporta dos tipos de jurisdicciones: jurisdicciones de envío y jurisdicciones fiscales. Cada una de estas jurisdicciones forma parte del grupo correspondiente, por ejemplo, las jurisdicciones de envío están en el grupo de jurisdicciones de envío y las jurisdicciones fiscales están en el grupo de jurisdicciones fiscales.

Las jurisdicciones y los grupos de jurisdicciones determinan qué normas de cálculo se utilizan para calcular los impuestos.

Las jurisdicciones y los grupos de jurisdicciones forman parte de una entidad de tienda. Cada jurisdicción y grupo de jurisdicciones forma parte de una entidad de tienda, sin embargo una entidad de tienda puede tener varias jurisdicciones y grupos de jurisdicciones. Si se suprime la entidad de tienda, también se suprimirán las jurisdicciones y los grupos de jurisdicciones asociados a dicha entidad de tienda.



Para obtener información más detallada sobre la estructura de los elementos de impuestos en WebSphere Commerce Server, consulte los modelos de datos y de objeto de impuestos de la ayuda en línea de WebSphere Commerce.

Creación de elementos de impuestos en WebSphere Commerce

Las herramientas de impuestos de WebSphere Commerce Accelerator le permiten crear y editar determinados elementos de impuestos (por ejemplo, las categorías de impuestos y las jurisdicciones), pero no todos los elementos de impuestos.

La lista siguiente describe detalladamente las tablas de base de datos que se pueden editar con las herramientas de impuestos.

- CALCODE
- CALCODEDSC
- CALRULE
- TAXJCRULE
- CRULESCALE
- CALSCALE
- CALSCALEDS
- CALRANGE
- CALRLOOKUP
- TAXCGRY
- TAXCGRYDS

- JURST
- JURSTGROUP
- JURSTGPREL
- CATENCALCD
- CATGPCALCD

También puede crear los elementos de impuestos en formato de archivos XML que se pueden cargar en la base de datos utilizando Loader package. De este modo, tiene las dos opciones siguientes para crear elementos de envío:

- Edite los elementos de impuestos existentes de una de las tiendas de ejemplo que se incluyen con WebSphere Commerce o de un archivador de tienda existente o cree elementos de impuestos nuevos.
- Cree nuevos elementos de impuestos con formato de archivo XML

Para obtener información sobre cómo editar los elementos de impuestos de un archivador de tienda existente o la información general sobre impuestos, consulte la ayuda en línea de WebSphere Commerce. Para obtener información sobre cómo crear elementos de impuestos nuevos en forma de archivo XML, consulte “Creación de elementos de impuestos utilizando un archivo XML”.

Creación de elementos de impuestos utilizando un archivo XML

Cree los elementos de impuestos en formato de archivos XML que se puedan cargar en la base de datos utilizando Loader package. Para obtener más información sobre Loader package, consulte la Parte 10, “Publicación de la tienda”, en la página 345. Si crea una tienda globalizada, deberá crear archivos XML para cada entorno nacional al que la tienda dé soporte. El archivo específico de entorno nacional debe especificar toda la información de las descripciones, para que se pueda traducir fácilmente.

Las tiendas de ejemplo, de las que se toman muchos de los ejemplos de estas tareas, utilizan un archivo `tax.xml` para toda la información que no es necesario traducir y otro archivo `tax.xml` para cada entorno nacional que la tienda soporta, para la información que es necesario traducir. Los archivos específicos de entorno nacional contienen toda la información de las descripciones.

Para crear elementos de impuestos para la tienda utilizando un archivo XML, realice lo siguiente:

1. Revise la información del Capítulo 38, “Carga de grupos de elementos de base de datos de WebSphere Commerce”, en la página 405. Consulte la publicación *IBM WebSphere Commerce, Guía de infraestructura de cálculo*. La infraestructura de cálculo de WebSphere Commerce calcula los importes monetarios (por ejemplo, impuestos) asociados al producto o servicio que un cliente ha seleccionado para adquirir.
2. Revise los archivos XML utilizados para crear elementos de impuestos para las tiendas de ejemplo. Todos los archivos para las tiendas de ejemplo se encuentran en el archivo archivador de tienda correspondiente.
Los archivos de archivador de tienda están ubicados en el directorio siguiente:
 - `dir_instalación_WC/sampletores`
3. Revise la información del Apéndice B, “Creación de los datos”, en la página 465.

4. Cree un archivo `tax.xml`, copiando uno de los archivos `tax.xml` de los archivadores de tienda de ejemplo o creando uno nuevo. Para obtener más información, consulte el archivo DTD que corresponda a `tax.xml`. El archivo DTD se encuentra en el siguiente directorio:

- `dir_instalación_WC/schema`

5. Defina las jurisdicciones y el grupo de jurisdicciones a los que está enviando mercancías y servicios. Asigne las jurisdicciones fiscales a los grupos de jurisdicciones según sus normas de cálculo de categorías de impuestos.

- a. Utilizando el ejemplo siguiente como guía, defina un grupo de jurisdicciones en el archivo XML de la tabla JURSTGROUP:

```
<jurstgroup
jurstgroup_id="@jurstgroup_id_2"
description="Grupo de jurisdicciones fiscales 1"
subclass="2"
storeent_id="@storeent_id_1"
code="Mundial"/>
```

donde

- `jurstgroup_id` es una clave exclusiva generada
 - `description` es una breve descripción del grupo de jurisdicciones, adecuada para visualizarse en una interfaz de usuario que gestione grupos de jurisdicciones.
 - `subclass` es la subclase del grupo de jurisdicciones tal como se indica a continuación:
 - 1 = `ShippingJurisdictionGroup`
 - 2 = `TaxJurisdictionGroup`
 - `storeent_id` es la entidad de tienda asociada a este grupo de jurisdicciones.
 - `code` que, junto con la entidad de tienda y la subclase, identifica de forma exclusiva este grupo de jurisdicciones.
- b. Utilizando el ejemplo siguiente como guía, defina una jurisdicción en el archivo XML de la tabla JURST.

```
<jurst
jurst_id="@jurst_id_2"
storeent_id="@storeent_id_1"
code="Mundial"
subclass="2"/>
```

donde

- `jurst_id` es una clave exclusiva generada
 - `storeent_id` es la entidad de tienda asociada a este grupo de jurisdicciones.
 - `code` que, junto con la entidad de tienda y la subclase, identifica de forma exclusiva este grupo de jurisdicciones.
 - `subclass` es la subclase de jurisdicción tal como se indica a continuación:
 - 1 = `ShippingJurisdiction`
 - 2 = `TaxJurisdiction`
- c. Utilizando el ejemplo siguiente como guía, asocie la jurisdicción que ha creado en el paso b al grupo de jurisdicciones que ha definido en el paso a, añadiendo información a la tabla JURSTGRPTEL.

```
<jurstgpre1
jurst_id="@jurst_id_2"
jurstgroup_id="@jurstgroup_id_1"
subclass="2"/>
```

donde

- jurst_id es la jurisdicción
- jurstgroup_id es el grupo de jurisdicciones
- subclass es la subclase de la jurisdicción y del grupo de jurisdicciones. Estas deben coincidir:
 - 1 = ShippingJurisdiction[Group]
 - 2 = TaxJurisdiction[Group]

d. Repita los pasos a a c para todas las jurisdicciones y todos los grupos de jurisdicciones que la tienda soporte.

6. Defina las categorías de impuestos que utilizará la tienda.

a. Utilizando el ejemplo siguiente como guía, defina una categoría de impuestos en el archivo XML para la tabla TAXCGRY:

```
<taxcgry
taxcgry_id="@taxcgry_id_1"
taxtype_id="-3"
storeent_id="@storeent_id_1"
name="Impuestos sobre la venta"
displayseq="0"
displayusage="0"/>
```

donde:

- taxcgry_id es una clave exclusiva generada.
- taxtype_id="-3" es el tipo de impuestos para esta categoría de impuestos. WebSphere Commerce soporta dos tipos de impuestos:
 - impuestos sobre la venta o el uso (-3)
 - impuestos sobre el envío (-4)
- storeent_id es la entidad de tienda asociada a esta categoría de impuestos.
- name es el nombre de la categoría de impuestos. Junto con la entidad de tienda, el nombre identifica esta categoría de impuestos de forma exclusiva.
- displayseq especifica la secuencia, del más bajo al más alto, de los importes de los impuestos cuando se visualizan, por ejemplo, en un pedido.
- displayusage especifica que esta categoría de impuestos en relación con el bean de datos de precio es la siguiente:
 - 0 = no se calcula
 - 1 = se calcula

El bean de datos de precio puede utilizarse para obtener importes de impuestos que deben mostrarse junto con el precio del producto.

b. Repita el paso a para cada categoría de impuestos utilizada en la tienda.

c. Utilice el ejemplo siguiente como guía y añada la información de descripción de categoría de impuestos a su archivo XML para la tabla TAXCGRYDS. Si está creando una tienda multicultural, deberá incluir esta información en un archivo XML específico de entorno nacional.

```
<taxcgryds
  taxcgry_id="@taxcgry_id_1"
  description="Impuestos sobre la venta"
  language_id="&es_ES"/>
```

donde

- taxcgry_id es la categoría de impuestos.
- description es una descripción breve de la categoría de impuestos, adecuada para mostrarla a los clientes.
- language_id es el idioma en que se visualiza esta información.

d. Repita el paso c para cada categoría de impuestos utilizada en la tienda.

7. Defina los códigos de cálculo que deberá utilizar la tienda.

a. Utilizando los ejemplos siguientes como guía, defina el código de cálculo en el archivo XML para la tabla CALCODE.

```
<calcode
  calcode_id="@calcode_id_3"
  code="Código de impuestos 1"
  calusage_id="-3"
  storeent_id="@storeent_id_1"
  groupby="0"
  published="1"
  sequence="0"
  calmethod_id="-43"
  calmethod_id_app="-44"
  calmethod_id_qfy="-42"
  displaylevel="0"
  flags="0"
  precedence="0"
/>
```

donde:

- calcode_id es una clave exclusiva generada.
- code es una serie de caracteres que identifica de forma exclusiva este código de cálculo, dado un uso de cálculo y una entidad de tienda determinados.
- calusage_id indica la clase de cálculo para la que se utiliza este código de cálculo. Por ejemplo, el código de cálculo puede utilizarse para calcular uno de los importes monetarios siguientes:
 - descuentos (-1)
 - costes de envío (-2)
 - impuestos sobre la venta (-3)
 - impuestos sobre el envío (-4)
 - cupones (-5)
- storeent_id es la entidad de tienda asociada a este código de cálculo.
- groupby son distintivos de bit que indican al método de combinación de código de cálculo cómo deben agruparse los artículos de pedido al realizar los cálculos. Cero especifica que no se agrupan (todos los artículos de pedido están en un solo grupo). Consulte *Tabla CALCODE: detalles* en la ayuda en línea de WebSphere Commerce para obtener más información.
- published especifica si el código de cálculo se publica o no:
 - 0 = no publicado (inhabilitado temporalmente)
 - 1 = publicado
 - 2 = marcado para supresión (y no publicado)

- `sequence` es el orden en el que se calcula el código de cálculo. Los códigos de cálculo se calculan y se aplican en secuencia del más bajo al más alto. Si dos códigos de cálculo tienen el mismo número de secuencia, se calcularán primero los códigos de cálculo cuyo `calcode_id` sea menor.
 - `calmethod_id`: método de cálculo del código de cálculo que define cómo se calculan los importes de los impuestos para este código de cálculo. Para determinar qué método de cálculo de código de cálculo se debe utilizar, realice lo siguiente:
 - Consulte la tabla CALMETHOD en la ayuda en línea de WebSphere Commerce. Consulte la descripción de la columna SUBCLASS. Pulse el enlace para la tabla CALMETHOD: detalles. Esta tabla lista los tipos de CALMETHOD disponibles. El método de cálculo del código de cálculo es 3.
 - Abra el archivo de rutina de carga `wcs.bootstrap_xx_XX.xml`, donde `xx_XX` es el código del entorno nacional. Los archivos de rutina de carga se encuentran en el directorio siguiente:
 - `dir_instalación_WC/schema/xml`
 - Localice la sección que lista los métodos de cálculo disponibles (CALMETHOD).
 - Localice los métodos de cálculo con el valor `calusage_ID` para impuestos (-3 para impuestos sobre la venta y -4 para impuestos sobre el envío).
 - Localice el método de cálculo que tiene la subclase de 3. Este método de cálculo es -43.
 - `calmethod_id_app` es el método `CalculationCodeApplyMethod` que almacena el importe calculado para los artículos de pedido asociados. Utilice el método descrito en `calmethod_id` para determinar qué método de aplicación de código de cálculo se debe utilizar.
 - `calmethod_id_app="-44"` es el método `CalculationCodeApplyMethod` para Impuesto sobre la venta
 - `calmethod_id_qfy` es el método `CalculationCodeQualifyMethod` que define qué artículos de pedido están asociados a este código de cálculo. Utilice el método descrito en `calmethod_id` para determinar qué método de calificación de código de cálculo se debe utilizar.
 - `calmethod_id_qfy="-42"` es el método `CalculationCodeQualifyMethod` para Impuesto sobre la venta.
 - `display level` determina si los importes calculados por este código de cálculo deben visualizarse con cada:
 - 0 = Artículo de pedido
 - 1 = Pedido
 - 2 = Producto
 - 3 = Artículo
 - 4 = Contrato
 - `flags` especifica si se debe invocar el método `CalculationCodeQualifyMethod` de este código de cálculo.
 - 0 = no restringido. El método no se invocará
 - 1 = restringido. El método se invocará.
- b. Utilizando el ejemplo siguiente como guía, añada la información de descripción de código de cálculo al archivo XML para la tabla CALCODEDSC. Si está creando una tienda multicultural, deberá incluir esta información en un archivo XML específico de entorno nacional.

```
<calcodedsc
  calcode_id="@calcode_id_3"
  description="Vitaminas
  language_id="@es_ES"
  longdescription="En Ontario, a las vitaminas se les aplican impuestos
                  federales, pero no provinciales."
/>
```

donde

- calcode_id es el código de cálculo al que se aplica esta información.
- description es una descripción breve del código de cálculo.
- language_id es el idioma para el que se aplica esta información.
- longdescription es la descripción detallada del código de cálculo.

c. Repita los pasos a y b para cada código de cálculo utilizado en la tienda.

8. Defina las normas de cálculo para la tienda.

a. Utilizando el ejemplo siguiente como guía, configure la norma de cálculo en el archivo XML para la tabla CALRULE:

```
<calrule
  calrule_id="@calrule_id_10"
  calcode_id="@calcode_id_3"
  startdate="1900-01-01 00:00:00.000000"
  taxcgry_id="@taxcgry_id_1"
  enddate="2100-01-01 00:00:00.000000"
  flags="1"
  identifier="1"
  combination="2"
  calmethod_id="-47"
  calmethod_id_qfy="-46"
/>
```

donde

- calrule_id es un identificador exclusivo generado.
- calcode_id es el código de cálculo del que forma parte esta norma de cálculo.
- startdate es el momento en el que entra en vigor esta norma de cálculo.
- taxcgry_id es la categoría de impuestos para la que está en vigor esta norma de cálculo.
- enddate es la hora en que esta norma de cálculo deja de estar en vigor.
- combination lo utiliza el método CalculationRuleCombineMethod para determinar cómo se puede combinar esta norma de cálculo con otras normas de cálculo. Consulte la tabla CALRULE para obtener más información.
- identifier identifica la norma de cálculo, en combinación con el código de cálculo de la misma.
- flags especifica el distintivo de bit para indicar el proceso especial que deberá realizar la implementación por omisión de CalculationRuleCombineMethod. Consulte la tabla CALRULE en la ayuda en línea de WebSphere Commerce para obtener más información.
- calmethod_id es el método CalculationRuleCalculateMethod que calcula un resultado monetario para un conjunto de artículos de pedido. Para determinar qué método de cálculo de norma de cálculo se debe utilizar, realice lo siguiente:
 - Consulte la tabla CALMETHOD en la ayuda en línea de WebSphere Commerce. Consulte la descripción de la columna SUBCLASS. Pulse el

enlace para la tabla CALMETHOD: detalles. Esta tabla lista los tipos de CALMETHOD disponibles. El método de cálculo de la norma de cálculo es 7.

- Abra el archivo de rutina de carga `wcs.bootstrap_xx_XX.xml`, donde `xx_XX` es el código del entorno nacional. Los archivos de rutina de carga se encuentran en el directorio siguiente:
 - `dir_instalación_WC/schema/xml`
- Localice la sección que lista los métodos de cálculo disponibles (CALMETHOD).
- Localice los métodos de cálculo con el valor `calusage_ID` para impuestos (-3 para impuestos sobre la venta y -4 para impuestos sobre el envío).
- Localice el método de cálculo que tiene la subclase de 7. Este método de cálculo es -47.
- `calmethod_id_qfy` es el método `CalculationRuleQualifyMethod` que determina qué artículo de pedido de un conjunto de artículos de pedido debe enviarse al método `CalculationRuleCalculateMethod`. Utilice el método descrito en `calmethod_id` para determinar qué método de calificación de norma de cálculo se debe utilizar.

- b. Repita el paso a para cada norma de cálculo utilizada en la tienda. Tenga en cuenta que cada código de cálculo puede tener varias normas de cálculo, una para cada categoría de impuesto aplicable. Por ejemplo, `calcode_id="@calcode_id_1"` puede estar asociado a varios `calrule_id`.

9. Defina escalas de cálculo para la tienda.

Una escala de cálculo es el conjunto de rangos que se aplicarán al cálculo. Estos rangos crean una escala.

- a. Utilizando el ejemplo siguiente como guía, configure la escala de cálculo en el archivo XML para la tabla CALSCALE:

```
<calscale
  calscale_id="@calscale_id_19"
  code="Impuestos sobre la venta 1"
  storeent_id="@storeent_id_1"
  calusage_id="-3"
  setccurr="EUR"
  calmethod_id="-53"
/>
```

donde

- `calscale_id` es un identificador exclusivo generado.
- `code` es una serie de caracteres que identifica de forma exclusiva esta escala de cálculo, dado un uso de cálculo y una entidad de tienda determinados.
- `storeent_id` es la entidad de tienda de la que forma parte esta escala de cálculo.
- `calusage_id` indica la clase de cálculo para la que se utiliza esta escala de cálculo. Por ejemplo, la escala de cálculo puede utilizarse para calcular uno de los importes monetarios siguientes:
 - descuentos (-1)
 - costes de envío (-2)
 - impuestos sobre la venta (-3)
 - impuestos sobre el envío (-4)
 - cupones (-5)

- `setcurr`, si se especifica, indica la moneda para los valores de inicio de rango de los objetos de rango de cálculo para esta escala de cálculo. El método `CalculationScaleLookupMethod` debe devolver un "número de búsqueda" en esta moneda. En este caso, no se especifica; el método `CalculationScaleLookupMethod` devolverá un número de búsqueda en la moneda del pedido. No es necesario especificar la moneda a menos que los valores de inicio del rango de escala no sean cero.
- `calmethod_id` es el método `CalculationScaleLookupMethod` que, dado un conjunto de artículos de pedido, determina un número de búsqueda, un importe base, un multiplicador de resultado y un conjunto de directrices matemáticas que la escala de cálculo puede utilizar para calcular un importe monetario. Para determinar qué método `CalculationScaleLookupMethod` se debe utilizar, realice lo siguiente:
 - Consulte la tabla CALMETHOD en la ayuda en línea de WebSphere Commerce. Consulte la descripción de la columna SUBCLASS. Pulse el enlace para la tabla CALMETHOD: detalles. Esta tabla lista los tipos de CALMETHOD disponibles. El método `MonetaryCalculationScaleLookupMethod` es 9.
 - Abra el archivo de rutina de carga `wcs.bootstrap_xx_XX.xml`, donde `xx_XX` es el código del entorno nacional. Los archivos de rutina de carga se encuentran en el directorio siguiente:
 - `dir_instalación_WC/schema/xml`
 - Localice la sección que lista los métodos de cálculo disponibles (CALMETHOD).
 - Localice los métodos de cálculo con el valor `calusage_ID` para impuestos (-3 para impuestos sobre la venta y -4 para impuestos sobre el envío).
 - Localice el método de cálculo que tiene la subclase de 9. Existen varios métodos de cálculo con la subclase de 9. Seleccione el que satisfaga sus necesidades.

Para obtener más información, consulte la tabla CALSCALE en la ayuda en línea de WebSphere Commerce.

- b. Repita el paso a para cada escala de cálculo utilizada en la tienda.
- c. Utilizando el ejemplo siguiente como guía, añada la información de descripción de escala de cálculo al archivo XML para la tabla CALSCALDS. Si está creando una tienda multicultural, deberá incluir esta información en un archivo XML específico de entorno nacional.

```
<calscaleds
  calscale_id="@calscale_id_19"
  description="5% de impuestos sobre la venta"
  language_id="&es_ES"
/>
```

donde

- `calscale_id` es la escala de cálculo a la que se aplica la descripción.
 - `description` es una descripción breve de la escala de cálculo, adecuada para describir a los clientes cómo se realiza un cálculo. Por ejemplo, "0,10 dólares por kilo, importe mínimo de 5,00 dólares" o "10% de descuento en la compra de 5 o más".
 - `language_id` es el idioma en que se visualiza esta información.
- d. Repita el paso c para cada escala de cálculo utilizada en la tienda.

10. Defina rangos de cálculo para las escalas de cálculo.

- a. Utilizando el ejemplo siguiente como guía, configure el rango de cálculo en el archivo XML para la tabla CALRANGE.

```
<calrange  
  calrange_id="@calrange_id_37"  
  calscale_id="@calscale_id_19"  
  calmethod_id="-59"  
  rangestart="0.00000"  
  cumulative="0"  
>
```

donde

- `calrange_id` es un identificador exclusivo generado.
- `calscale_id` es la escala de cálculo de la que forma parte este rango de cálculo.
- `calmethod_id` es el método `CalculationRangeMethod` que determina un importe monetario a partir de `CalculationRangeLookupResult`. Por ejemplo, `FixedAmountCalculationRangeCmd`, `PerUnitAmountCalculationRangeCmd` o `PercentageCalculationRangeCmd`. Para determinar el método `CalculationRangeMethod`, realice lo siguiente:
 - Consulte la tabla `CALMETHOD` en la ayuda en línea de WebSphere Commerce. Consulte la descripción de la columna `SUBCLASS`. Pulse el enlace para la tabla `CALMETHOD`: detalles. Esta tabla lista los tipos de `CALMETHOD` disponibles. El método `CalculationRangeMethod` es 10.
 - Abra el archivo de rutina de carga `wcs.bootstrap_xx_XX.xml`, donde `xx_XX` es el código del entorno nacional. Los archivos de rutina de carga se encuentran en el directorio siguiente:
 - `dir_instalación_WC/schema/xml`
 - Localice la sección que lista los métodos de cálculo disponibles (`CALMETHOD`).
 - Localice los métodos de cálculo con el valor `calusage_ID` para impuestos (-3 para impuestos sobre la venta y -4 para impuestos sobre el envío).
 - Localice el método de cálculo cuya subclase es 10. Hay varios métodos de cálculo cuya subclase es 10. Seleccione uno que satisfaga sus necesidades.
- `rangestart` es un número de búsqueda mayor o igual que `RANGESTART` o si `RANGESTART` es `NULL`, esta fila coincide con el número de búsqueda.
- `cumulative` es lo siguiente:
 - 0 = sólo se utiliza el rango de cálculo coincidente que tiene el valor `RANGESTART` más alto.
 - 1 = se utilizan todos los rangos de cálculo coincidentes. Los importes monetarios calculados se suman para llegar al resultado final.

Para obtener más información, consulte la tabla `CALRANGE` en la ayuda en línea de WebSphere Commerce.

- b. Repita el paso a para cada rango de cálculo asociado a la escala de cálculo utilizada en la tienda. En el ejemplo anterior solamente hay un rango, ya que a todos los importes de los impuestos se les aplica el mismo tipo impositivo.
11. Defina los valores de búsqueda de cálculo para las escalas de cálculo. Los valores de búsqueda de cálculo son los valores asociados a la escala de

cálculo. Por ejemplo, una escala de cálculo incluye los rangos siguientes y los tipos impositivos para los impuestos sobre ventas provinciales de Ontario en las comidas de un restaurante:

- el tipo impositivo de cantidades comprendidas entre 0,00 dólares y 3,99 dólares es del 0,00%
 - el tipo impositivo de cantidades de 4,00 y más es del 8,00%
- Los valores de búsqueda son de 0,00 y 8,00.

a. Utilizando los ejemplos siguientes como guía, configure la búsqueda de cálculo en el archivo XML para la tabla CALRLOOKUP.

```
<calrlookup  
  calrlookup_id="@calrlookup_id_37"  
  calrange_id="@calrange_id_37"  
  value="5.00"  
>
```

donde

- `calrlookup_id` es un identificador exclusivo generado.
- `calrange_id` es el rango de cálculo del que forma parte este resultado de búsqueda de rango de cálculo.
- `value` es el valor del resultado de búsqueda de rango de cálculo, utilizado por el método de rango de cálculo para determinar un resultado monetario. En este ejemplo, el tipo impositivo es de 5,00%.

Para obtener más información, consulte la tabla CALRLOOKUP en la ayuda en línea de WebSphere Commerce.

b. Repita los pasos a y b para cada valor de búsqueda asociado a la escala de cálculo utilizada en la tienda. En este ejemplo, solamente hay un valor CALRLOOKUP, ya que CALRLOOKUP.SETCCURR es NULL, y solamente hay un CALRANGE ya que el tipo impositivo es el mismo para todos los importes.

12. Asocie la norma de cálculo y la escala de cálculo.

a. Utilizando los ejemplos siguientes como guía, asocie la escala de cálculo a la norma de cálculo en el archivo XML para la tabla CRULESCALE.

```
<crulescale  
  calrule_id="@calrule_id_10"  
  calscale_id="@calscale_id_19"  
>
```

donde

- `calrule_id` es la norma de cálculo.
 - `calscale_id` es la escala de cálculo.
- b. Repita el paso a para cada escala de cálculo y asociación de norma. En el ejemplo que se ha utilizado anteriormente, solamente hay una escala de cálculo para cada norma de cálculo.

Nota: Si el tipo impositivo varía según la cantidad comprada, necesitará crear escalas cuyos valores de `rangestart` no sean cero. A continuación, tendrá que crear una escala de cálculo para cada moneda soportada (estableciendo `CALSCALE.SETCCURR` en la moneda correcta) para la que no ha establecido una tasa de conversión (consulte la tabla `CURCONVERT`) y asícielas todas con la norma de cálculo para esa categoría de impuestos determinada. Por ejemplo, en Ontario no hay ningún impuesto provincial sobre las ventas de comida cuyo importe esté por debajo de los 4,00 dólares. Si la tienda soporta la venta de comida en dólares americanos, tendrá que establecer una conversión de dólares

americanos a dólares de Canadá, o crear una escala de cálculo de impuestos diferente con un valor rangestart adecuado, probablemente 6,00 dólares americanos, y asociarla con la misma norma de cálculo de impuestos. Solamente se utilizará la escala de cálculo adecuada, según la moneda del pedido.



Para obtener más información sobre cómo utilizar @ y &, consulte el Apéndice B, "Creación de los datos", en la página 465.

Creación de elementos de despacho de impuestos

Para que los elementos de impuestos funcionen correctamente en la tienda, deberá asociar los grupos de jurisdicciones fiscales de la tienda al centro de despacho de pedidos utilizado por la tienda y, a continuación, asociar una norma de cálculo a ambos.

Deberá crear los elementos de despacho para poder asociar los elementos de impuestos a un centro de despacho de pedidos. Para obtener más información sobre cómo crear elementos de despacho, consulte "Creación de elementos de despacho de pedidos en WebSphere Commerce" en la página 218.

Después de haber creado los elementos de despacho, asocie a los mismos los elementos de impuestos, añadiendo información a la tabla TAXJCRULE. Efectúe lo siguiente:

1. Consulte la publicación *IBM WebSphere Commerce, Guía de infraestructura de cálculo*. La infraestructura de cálculo de WebSphere Commerce calcula los importes monetarios (por ejemplo, impuestos) asociados al producto o servicio que un cliente ha seleccionado para adquirir.
2. Revise los archivos XML utilizados para crear elementos de despacho de impuestos para las tiendas de ejemplo. Todos los archivos para las tiendas de ejemplo se encuentran en el archivo archivador de tienda correspondiente.

Los archivos de archivador de tienda están ubicados en el directorio siguiente:

- *dir_instalación_WC/samplestores*

Cada tienda de ejemplo incluye un archivo *taxfulfill.xml*, que contiene la información de impuestos. Para ver el archivo *taxfulfill.xml* del archivador de tienda, descomprímalo utilizando un programa ZIP. El archivo *taxfulfill.xml* está ubicado en el directorio de datos.

3. Revise la información del Apéndice B, "Creación de los datos", en la página 465.
4. Cree un archivo *taxfulfill.xml*, copiando uno de los archivos *taxfulfill.xml* de los archivadores de tienda de ejemplo o creando uno nuevo. Para obtener más información, consulte el archivo DTD que corresponda a *taxfulfill.xml*. Los archivos DTD se encuentran en el directorio siguiente:
 - *dir_instalación_WC/xml/sar*
5. Utilizando el ejemplo siguiente como guía, en el archivo XML añada información para la tabla TAXJCRULE:

```
<taxjcrule
taxjcrule_id="@taxjcrule_id_1"
calrule_id="@calrule_id_10"
ffmcenter_id="@ffmcenter_id_1"
jurstgroup_id="@jurstgroup_id_2"
precedence="0"
/>
```

donde

- `taxjcrule_id` es el identificador exclusivo generado.
- `calrule_id` es la norma de cálculo utilizada.
- `ffmcenter_id` es el centro de despacho de pedidos. Si es NULL, esta asociación se aplica a todos los centros de despacho de pedidos.
- `jurstgroup_id` es el grupo de jurisdicciones fiscales. Si es NULL, esta asociación se aplica a todos los grupos de jurisdicciones fiscales.
- `precedence`: cuando una dirección de envío está en más de uno de los grupos de jurisdicciones fiscales especificados, para el mismo centro de despacho de pedidos, sólo se habilita la norma de cálculo con el valor `TAXJCRULE.PRECEDENCE` más alto.

6. Repita el paso 3 para cada grupo de jurisdicciones, centro de despacho de pedidos y asociación de normas de la tienda.



Para obtener más información sobre cómo utilizar @ y &, consulte el Apéndice B, "Creación de los datos", en la página 465.

Creación de elementos tienda-catálogo-impuestos

Para asociar los impuestos a las mercancías y los servicios de la tienda, deberá asociar un código de cálculo a las entradas de catálogo de la tienda para cada contrato que la tienda incluya.

Para poder crear elementos de tienda-catálogo-impuestos, deberá crear los elementos de tienda y de catálogo. Para obtener más información sobre cómo crear elementos de tienda, consulte "Creación de elementos de datos de tienda en un archivo XML" en la página 136. Para obtener más información sobre cómo crear elementos de catálogo, consulte "Visualización de los elementos de catálogo de la tienda" en la página 177.

Para crear elementos de tienda-catálogo-impuestos, realice lo siguiente:

1. Consulte la publicación *IBM WebSphere Commerce, Guía de infraestructura de cálculo*. La infraestructura de cálculo de WebSphere Commerce calcula los importes monetarios (por ejemplo, envío) asociados al producto o servicio que un cliente ha seleccionado para adquirir.
2. Revise los archivos XML utilizados para crear elementos de tienda-catálogo-impuestos para las tiendas de ejemplo. Todos los archivos para las tiendas de ejemplo se encuentran en el archivo archivador de tienda correspondiente.
Los archivos de archivador de tienda están ubicados en el directorio siguiente:
 - `dir_instalación_WC/samplestores`
3. Revise la información del Apéndice B, "Creación de los datos", en la página 465.
4. Cree un archivo `store-catalog-tax.xml`, copiando uno de los archivos `store-catalog-tax.xml` de los archivadores de tienda de ejemplo o creando uno nuevo. Para obtener más información, consulte el archivo DTD que corresponda a `store-catalog-tax.xml`. Los archivos DTD se encuentran en el directorio siguiente:
 - `dir_instalación_WC/xml/sar`
5. Cree la relación de tienda-catálogo-impuestos añadiendo información a la tabla `CATENCALCD`. Utilice el ejemplo siguiente como guía:

```
<catenalcld
  calcode_id="@calcode_id_3"
  catenalcld_id="@catenalcld_id_3"
  store_id="@storeent_id_1"
/>
```

donde

- calcode_id es el código de cálculo.
- catenalcld_id es un identificador exclusivo generado.
- store_id es la tienda.



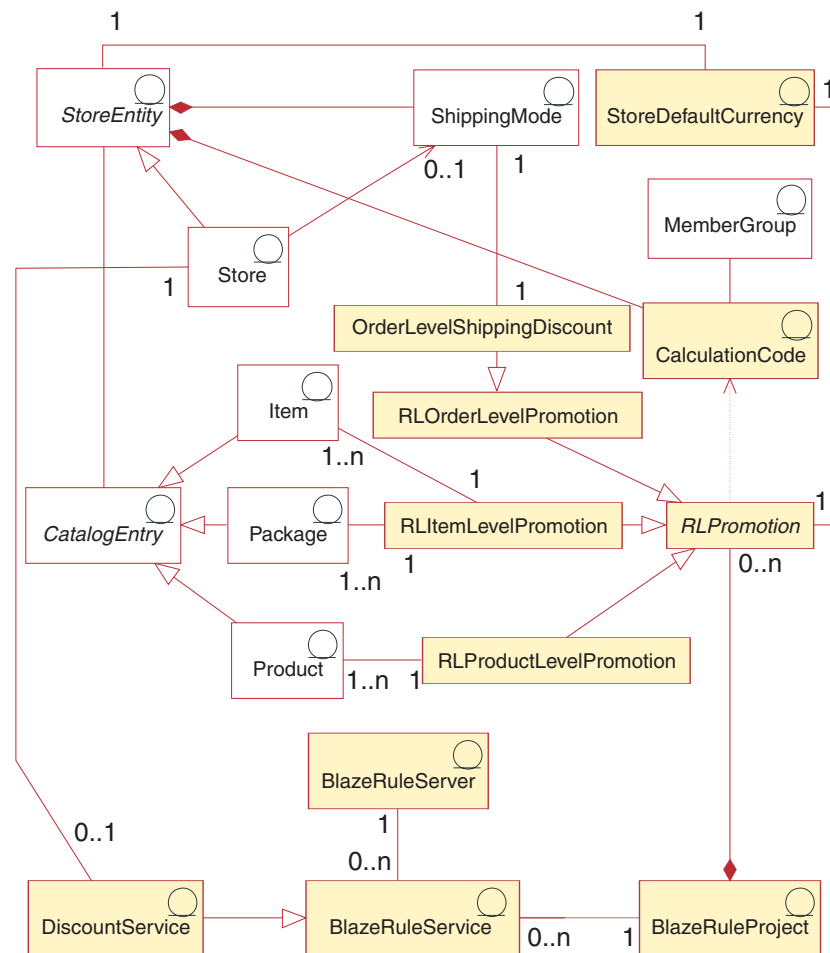
Para obtener más información sobre cómo utilizar @ y &, consulte el Apéndice B, "Creación de los datos", en la página 465.

Capítulo 28. Elementos de descuento

Los descuentos le permiten ofrecer a los clientes incentivos de precio para promocionar un producto. WebSphere Commerce dispone de dos implementaciones de descuentos: los descuentos basados en normas y los descuentos basados en esquema. Cada implementación permite ofrecer descuentos de porcentaje (por ejemplo, un descuento del 10%) o descuentos de cantidad fija (por ejemplo, un descuento de 15 euros). Los descuentos pueden aplicarse a productos específicos o a la compra total. Por ejemplo, puede ofrecer una reducción del 20% a las personas mayores; o si tiene muchas gorras rojas en existencias, puede ofrecer un 25% de descuento en las gorras durante un tiempo limitado. Los descuentos basados en normas van más allá de estos tipos de descuentos y ofrecen descuentos de gastos de envío y regalos para determinadas compras.

Descripción de los descuentos basados en normas en WebSphere Commerce

El diagrama siguiente ilustra la estructura de descuentos basados en normas de WebSphere Commerce Server.





Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce Server. Para obtener más información sobre el modelo de información, consulte “Modelo de información de los datos de la tienda” en la página 105. Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, “Descripción de UML”, en la página 463.

Moneda por omisión de la tienda

Esta es la moneda por omisión de la tienda según se ha definido en la tabla STOREENT. Los descuentos basados en normas se definen en esta moneda pero también se pueden evaluar a petición, utilizando cualquier moneda que soporte la tienda. Para obtener más información acerca del uso de la moneda, consulte el Capítulo 23, “Elementos de moneda”, en la página 235.

Código de cálculo

Un *descuento* se representa en la infraestructura de cálculo mediante un *código de cálculo*. Un código de cálculo de descuento indica cómo se calcula el descuento para los artículos de pedido mediante la norma de cálculo correspondiente.

Un código de cálculo forma parte de una entidad de tienda. En una entidad de tienda se pueden definir varios códigos de cálculo. Si se suprime la entidad de tienda, también se suprimirán los códigos de cálculo definidos en dicha entidad de tienda.

Cada código de cálculo de descuento tiene una fecha de inicio y una fecha final, que definen el período de tiempo en que el descuento es efectivo. El código de cálculo de descuento también se puede asociar con uno o varios grupos de miembros, que definen los grupos de miembros elegibles.

RLPromotion

Este es el objeto padre de los descuentos basados en normas. Aunque RLPromotion es el nombre del tipo de objeto, debe comprenderse que corresponde a un descuento basado en normas. Cada descuento basado en normas tiene un nombre, diferentes descripciones que se visualizan en diferentes circunstancias, una prioridad, un segmento de destino (perfil de cliente predefinido) y una planificación de ejecución que rige fechas y horas.

El atributo de prioridad necesita una clarificación adicional. El atributo de prioridad ayuda a resolver aquellos conflictos en los que se pueden aplicar simultáneamente varios descuentos. Los descuentos se aplican en el orden definido por sus respectivos valores de prioridad, en orden descendente. Es decir, el descuento con el valor de prioridad más alto se aplica primero.

Todos los objetos hijo listados a continuación clasifican adicionalmente los tipos de descuentos basados en normas y añaden valores específicos al tipo de descuento, cuando es necesario. Cada uno de estos objetos contiene también la lógica adecuada para manipular el archivo XML del dominio que define el descuento.

Promoción a nivel de RLProduct

Estos objetos representan descuentos basados en normas a nivel de producto. Esta clase se deriva de la clase RLPromotion. Esta clase requiere un atributo adicional, SKU, que identifica el producto de destino.

Promoción a nivel de RLItem

Estos objetos representan descuentos basados en normas a nivel de artículo. Esta clase se deriva de la clase RLPromotion. Esta clase requiere un atributo adicional, `catEntryID`, que identifica el producto de destino. Estas promociones a nivel de artículos también se utilizan para los kits premontados de destino con descuentos basados en normas ya que se pueden solicitar por separado con su propio atributo `catEntryID` y el precio. Los paquetes compuestos y los kits dinámicos no pueden ser destino de los descuentos basados en normas.

Promoción a nivel de RLOrder

Estos objetos representan descuentos basados en normas a nivel de pedido. Se deriva de RLPromotion. Esta clase requiere un atributo adicional, `inCombineWithProductLevelDiscount`, que determina si el descuento a nivel de pedido se puede aplicar al mismo tiempo que una promoción a nivel de producto.

Descuento de envío a nivel de pedido: Esta clase se deriva de la clase de promoción a nivel de RLOrder. Esta clase requiere atributos adicionales que definan el método de envío que se va a utilizar y una tarifa con el descuento aplicado.

Proyecto de normas de Blaze

El proyecto de normas de Blaze contiene todos los descuentos definidos actualmente para una tienda, generados a partir del archivo XML del dominio. El proyecto de normas reside en el sistema de archivos y se utiliza para rellenar el servicio de descuento.

Servicio de normas de Blaze

Se trata de una interfaz que WebSphere Commerce utiliza para comunicarse con el servidor de normas de Blaze.

Servicio de descuentos

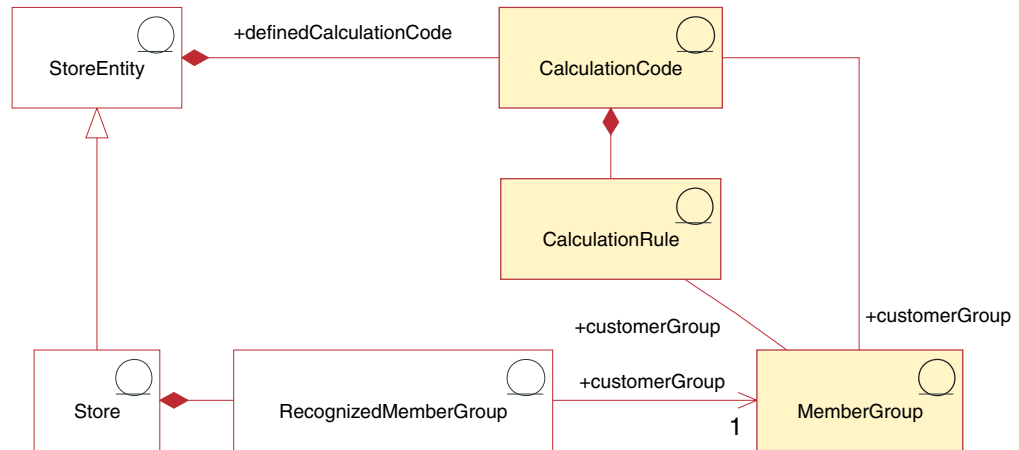
Es una ampliación del servicio de normas de Blaze y contiene una instancia del proyecto en el que se llevan a cabo la evaluación y el cálculo. Este servicio pasa el contexto de descuento, un objeto que contiene información acerca de los descuentos aplicables, a la infraestructura de cálculo de WebSphere Commerce.

Servidor de normas de Blaze

El software de Blaze que evalúa el proyecto de normas cuando llega una petición procedente de los mandatos de proceso de pedidos.

Descripción de los descuentos basados en esquema en WebSphere Commerce

El diagrama siguiente ilustra la estructura de descuentos basados en esquema de WebSphere Commerce Server.



Código de cálculo

Un *descuento* se representa y se calcula utilizando un *código de cálculo* de descuento. Un código de cálculo de descuento indica cómo se calcula el descuento para los artículos de pedido.

Un código de cálculo forma parte de una entidad de tienda. En una entidad de tienda se pueden definir varios códigos de cálculo. Si se suprime la entidad de tienda, también se suprimirán los códigos de cálculo definidos en dicha entidad de tienda.

Cada código de cálculo de descuento tiene una fecha de inicio y una fecha final, que definen el período de tiempo en que el descuento es efectivo. El código de cálculo de descuento también se puede asociar con uno o varios grupos de miembros, que definen los grupos de miembros elegibles.

Un código de cálculo de descuento puede adjuntarse a una o más entradas de catálogo y grupos de catálogo. La acción de adjuntar un código de cálculo a un grupo de catálogo tiene el mismo efecto que si se adjunta a todas las entradas de catálogo directamente en el grupo de catálogo. Sin embargo, los códigos de cálculo de descuento asociados al grupo de catálogo A no estarán asociados a los productos y artículos del grupo de catálogo B si el grupo de catálogo A contiene el grupo de catálogo B.

Las entradas de catálogo o los grupos de catálogo pueden tener más de un descuento asociado a ellos. Cuando sea aplicable más de un código de cálculo de descuento a un pedido, los cálculos de descuento se realizarán en secuencia ascendente de sus atributos de secuencia de código de cálculo.

Nota: Defina el orden de la secuencia de descuentos para implementar descuentos en los descuentos.

Los artículos de pedido se agrupan para el cálculo de uno de los modos siguientes:

- por acuerdo comercial

- por producto
- por oferta
- por dirección de envío

Para obtener más información, consulte la ayuda en línea de WebSphere Commerce.



Para obtener más información sobre el uso de códigos de cálculo, consulte la publicación *IBM WebSphere Commerce, Guía de infraestructura de cálculo*.

Normas de cálculo

Cada código de cálculo tiene un conjunto de normas de cálculo, que definen las condiciones bajo las cuales se realizará el cálculo. Cada norma de cálculo de descuento se asocia a uno o varios grupos de miembros, para el que está en vigor el descuento. Los grupos de miembros pueden tener derecho a obtener más de un descuento a la vez.

Nota: Si hay definido un grupo de miembros elegible en el nivel de código de cálculo, no será necesario volverlo a definir en el nivel de norma de cálculo.



Para obtener más información sobre el uso de normas de cálculo, consulte la publicación *IBM WebSphere Commerce, Guía de infraestructura de cálculo*.

Creación de elementos de descuento en WebSphere Commerce

El método principal para crear descuentos en una tienda creada con WebSphere Commerce consiste en utilizar el asistente de descuentos de WebSphere Commerce Accelerator. Para obtener más información sobre cómo crear descuentos utilizando WebSphere Commerce Accelerator, consulte la ayuda en línea de WebSphere Commerce.

Los descuentos también se pueden crear utilizando un archivo XML y, a continuación, se pueden cargar mediante Loader package. Sin embargo, los descuentos creados de este modo, así como los descuentos importados durante la migración de versiones anteriores, funcionarán correctamente pero puede que no se visualicen del modo adecuado en WebSphere Commerce Accelerator.

Capítulo 29. Elementos de inventario

El inventario incluye todo aquello que se puede contabilizar físicamente que se encuentra en un centro de despacho de pedidos. Se pueden proporcionar definiciones específicas de tipos de inventario, por ejemplo artículos, productos, códigos de artículo, paquetes compuestos y paquetes; pero todos ellos se consideran inventario. Los productos se configuran para el despacho de pedidos en el asistente Producto y el cuaderno Producto. Esto incluye opciones para hacer un seguimiento del inventario, permitir pedidos en espera, forzar pedidos en espera, despachar por separado y especificar que el producto no se debe devolver. WebSphere Commerce Accelerator hace una distinción entre los dos tipos principales de inventario que se puede recibir:

- El inventario esperado, que tiene asociado un registro de inventario esperado.
- El inventario ad hoc o inventario no registrado como esperado.

El inventario esperado se recibe de un proveedor y normalmente se paga con una orden de compra. WebSphere Commerce Accelerator hace un seguimiento del inventario esperado con los registros de inventario esperado y le permite registrar un identificador externo, normalmente un número de orden de compra de un sistema externo. De este modo, puede hacer fácilmente un seguimiento del inventario que ha solicitado, así como de lo que ha llegado y de lo que no. Los detalles del inventario esperado son los datos específicos acerca de los productos de un registro de inventario esperado, por ejemplo el centro de despacho de pedidos que espera el producto, la fecha de recepción esperada, la cantidad esperada y comentarios.

Un registro de inventario esperado no puede suprimirse una vez se ha recibido un inventario que se basa en él, y los detalles de un inventario esperado no pueden modificarse ni suprimirse en cuanto se ha recibido cualquier parte de ese inventario.

Cuando se formalizan pedidos del inventario que está disponible en un centro de despacho de pedidos, el sistema de pedidos asigna inventario a dichos pedidos. La asignación de inventario a un pedido deja el inventario no disponible para el sistema de pedidos. Si el pedido se cancela, el inventario vuelve a estar disponible. Si se formaliza un pedido de inventario que no está disponible, se puede crear un pedido en espera. Si hay inventario esperado que puede utilizarse para formalizar el pedido en espera, se asigna el inventario esperado al pedido en espera y se puede proporcionar al cliente la fecha de envío esperada.

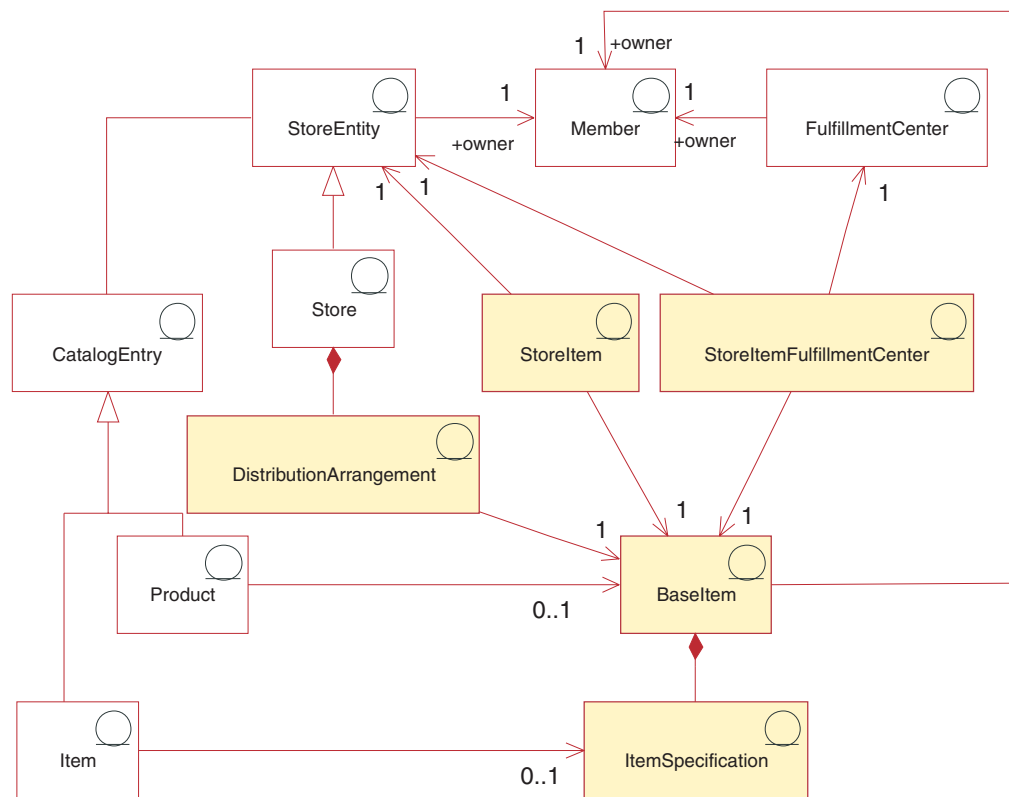
Las recepciones de inventario ad hoc se crean cuando llega inventario a un centro de despacho de pedidos sin un registro de inventario esperado correspondiente. Esto puede deberse a una llegada inesperada de inventario o puede que el comerciante o el vendedor haya elegido no utilizar registros de inventario esperado para registrar las recepciones de inventario.

Nota: Para que los productos puedan recibirse, deben existir en el sistema WebSphere Commerce, tanto si la recepción de inventario es esperada como si es ad hoc.

Los elementos de inventario en WebSphere Commerce

Para comprender los elementos de inventario, es necesario comprender las relaciones entre el inventario y la tienda. Esto puede explicarse mediante el uso de un modelo de información. Las secciones siguientes describen las relaciones y las asociaciones que tiene el inventario con una tienda y otros elementos. Los diagramas proporcionados a continuación muestran las relaciones y las asociaciones para el inventario ATP (Disponibilidad asegurada) y el inventario no ATP. Una tienda puede utilizar los métodos de inventario ATP y no ATP. Una tienda se considera habilitada para ATP si la columna ALLOCATIONGOODFOR de la tabla de base de datos STORE contiene un valor mayor que cero. A continuación se describe cada diagrama y sus asociaciones. Para obtener más información sobre ATP, consulte la ayuda en línea.

Inventario ATP



Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce Server. Para obtener más información sobre el modelo de información, consulte “Modelo de información de los datos de la tienda” en la página 105. Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, “Descripción de UML”, en la página 463.

Artículo base

El *artículo base* es el centro del diagrama de inventario y representa una familia general de mercancías con un nombre y una descripción comunes. Los artículos base se utilizan exclusivamente para el despacho y no son particulares de ninguna

tienda. Cada entrada de catálogo que representa un producto del catálogo tiene un artículo base correspondiente para el despacho. Los artículos base se definen en la tabla BASEITEM.

Especificación de artículo

Una *especificación de artículo* es un artículo base con valores definidos para todos sus atributos. Cada entrada de catálogo que representa un artículo en el catálogo tiene una especificación de artículo correspondiente para el despacho.

Entradas de catálogo

Los productos y los artículos son *entradas de catálogo*. Las entradas de catálogo están asociadas a entidades de tienda, lo que significa que las entradas de catálogo, por ejemplo los productos y los artículos, se encuentran en las tiendas.

Plan de distribución

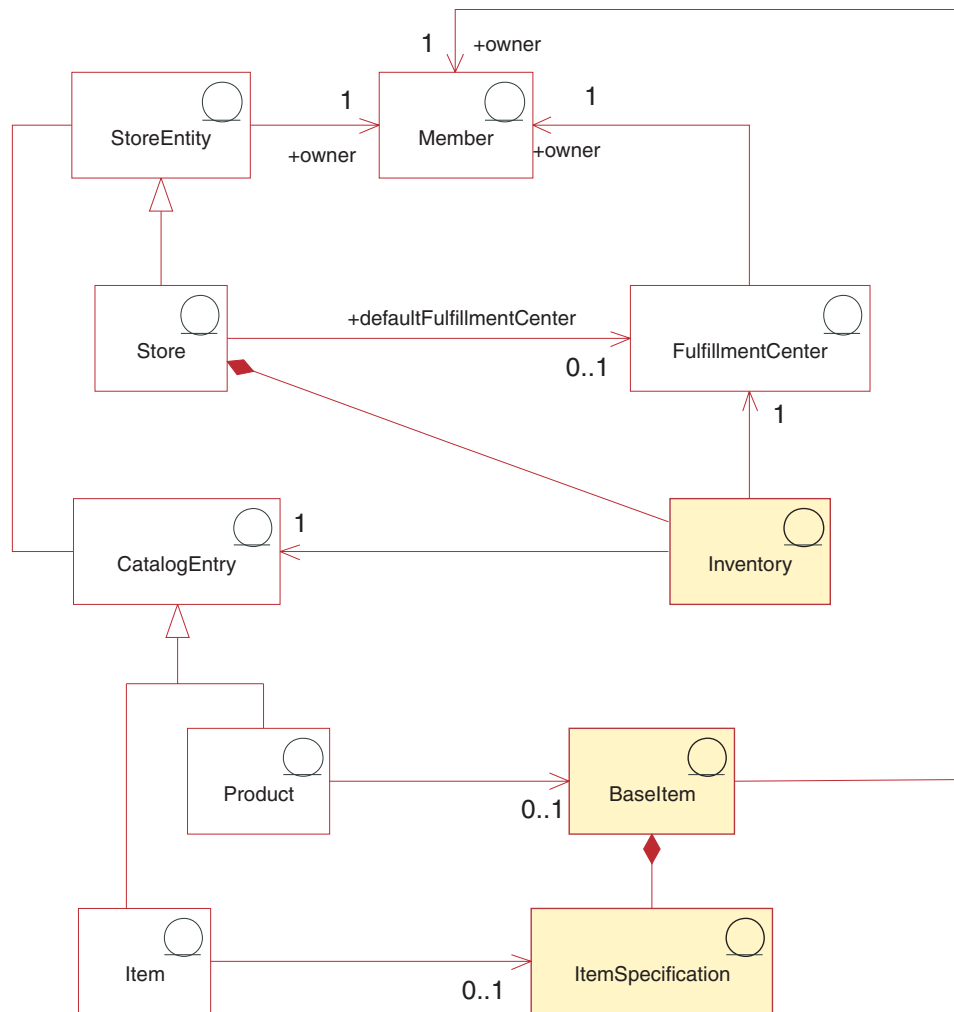
Un *plan de distribución* está asociado a un artículo base, lo que permite que una tienda venda su propio inventario. Los planes de distribución se almacenan en la tabla DISTARRANG.

Artículo de tienda

Un *artículo de tienda* representa atributos que afectan el modo en que una tienda o un grupo de tiendas determinado asigna inventario para los artículos especificados de un artículo base determinado, incluido si se debe permitir poner pedidos en espera y hacer un seguimiento del inventario. La tabla STORITMFFC define una estimación del número de segundos que se tardan desde el momento en que un artículo de pedido se entrega para el despacho hasta que se envía al cliente. Esta tabla sólo se llena de datos si una tienda desea definir un cambio en el margen de tiempo para el envío por omisión de FFMCENTER, para un artículo de tienda.

El elemento de artículo de tienda lo pueden utilizar otras tiendas, como se describe en el Capítulo 14, “Relaciones entre tiendas”, en la página 143.

Inventario no ATP



Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce Server. Para obtener más información sobre el modelo de información, consulte “Modelo de información de los datos de la tienda” en la página 105. Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, “Descripción de UML”, en la página 463.

El artículo base también es el centro del diagrama de inventario no ATP. La relación del artículo base con los productos, artículos y entradas de catálogo, es la misma que en el diagrama de inventario general. Un artículo base sigue siendo propiedad de un miembro y, una vez definido por dicho miembro, puede venderse en la tienda. Sin embargo, en este caso, no hay plan de distribución, ni asociación de artículo de tienda ni centro de despacho de pedidos de artículos de tienda.

Centro de despacho de pedidos

El inventario está asociado a un *centro de despacho de pedidos* y una tienda. Una tienda puede designar un centro de despacho de pedidos por omisión. Al igual que los artículos base, los centros de despacho de pedidos son propiedad de los miembros y éstos comparten dicha propiedad con la tienda. Para obtener más información sobre los elementos de despacho, consulte el Capítulo 19, “Elementos

de despacho de pedidos”, en la página 215.



Para obtener información más detallada sobre la estructura de los elementos de inventario en WebSphere Commerce Server, consulte los modelos de datos y de objeto de inventario en la ayuda en línea de WebSphere Commerce.

Creación de elementos de inventario en WebSphere Commerce

Puesto que el inventario son datos operativos, cambia a diario a medida que los clientes compran productos en la tienda o devuelven artículos a la misma. Como resultado, los niveles de inventario suben y bajan a medida que se venden productos y a medida que los centros de despacho de pedidos reciben inventario nuevo de los proveedores. WebSphere Commerce Accelerator le permite realizar las siguientes tareas relacionadas con el inventario:

- registrar inventario esperado
- recibir inventario esperado y ad hoc de los proveedores
- ajustar inventario
- mantener registros de devoluciones
- mantener razones de devolución
- recibir inventario devuelto por los clientes
- gestionar la disposición del inventario devuelto

Para obtener más información sobre cómo utilizar WebSphere Commerce Accelerator para gestionar inventario, consulte la ayuda en línea de WebSphere Commerce.

Gestión de códigos de ajuste de inventario

WebSphere Commerce proporciona códigos de ajuste de inventario para que los usuarios puedan especificar el motivo por el que se ha realizado un ajuste concreto del inventario. En la tabla siguiente se listan los códigos de ajuste de inventario que se proporcionan inicialmente con WebSphere Commerce. También puede ver estos códigos si realiza una consulta de base de datos `SELECT *` en las tablas de base de datos `INVADJCODE` e `INVADJDESC`. Como desarrollador de tiendas, puede añadir, cambiar o suprimir estos códigos según sea necesario para que se ajusten a las necesidades del entorno de inventario. Los códigos que se describen en la tabla siguiente pueden considerarse ejemplos de los tipos de códigos que puede utilizar. Puede obtener un entorno más personalizado añadiendo sus propios códigos de ajuste o modificando los códigos cargados previamente.

Tabla 10. Códigos de ajuste de inventario cargados previamente

Código de ajuste	Descripción	Explicación
RTND	Devuelto	Utilice este código para especificar que se ha devuelto un artículo* al inventario. Por ejemplo, es posible que un cliente devuelva un artículo de un pedido porque el color o el tamaño no eran los correctos. Este código se carga previamente para que lo utilicen todas las tiendas de ejemplo.

Tabla 10. Códigos de ajuste de inventario cargados previamente (continuación)

Código de ajuste	Descripción	Explicación
EXPD	Caducado	Utilice este código para especificar que se ha caducado un artículo* en el inventario. Por ejemplo, las medicinas bajo receta o los productos con una fecha de caducidad corta como, por ejemplo, los productos lácteos, pueden caducar. Este código se carga previamente para que lo utilicen todas las tiendas de ejemplo.
DMGD	Dañado	Utilice este código para especificar que se ha dañado un artículo (por ejemplo, si tiene un golpe, se ha rayado o tiene algún defecto). Este código se carga previamente para que lo utilicen todas las tiendas de ejemplo.
LOST	Perdido	Utilice este código para especificar que un artículo se ha perdido o ha sido robado del centro de despacho de pedidos o del lugar donde se almacenaba. Este código se carga previamente para que lo utilicen todas las tiendas de ejemplo.
MSCT	Mal calculado	Utilice este código para especificar que se ha calculado erróneamente el número para este artículo. Este código se carga previamente para que lo utilicen todas las tiendas de ejemplo.
PCNT	Número real	Utilice este código para especificar el número físico real de este artículo. Este código se carga previamente para las tiendas de ejemplo Directo al consumidor y Directo B2B pero no para las tiendas alojadas.
SPLG	Roto	Utilice este código para especificar que se ha estropeado un artículo. Este código se carga previamente para las tiendas de ejemplo Directo al consumidor y Directo B2B pero no para las tiendas alojadas.
DISC	Descartar	Utilice este código para indicar que se ha descartado o se descartará un artículo de este inventario. Este código se carga previamente para las tiendas de ejemplo Directo al consumidor y Directo B2B pero no para las tiendas alojadas.
*Aquí se utiliza el término "artículo" en su sentido general. Los artículos de inventario, pueden ser artículos, productos, códigos de artículo, paquetes y paquetes compuestos.		

En esta tabla, el valor que aparece en la columna de código de artículo representa el valor que aparece en la columna ADJUSTCODE de la tabla de base de datos INVADJCODE. El valor que aparece en la columna de descripción representa el valor de la columna DESCRIPTION de la tabla de base de datos INVADJDESC.

Para añadir, modificar o suprimir los códigos de ajuste de inventario, utilice el mandato Load (massloader). Puede encontrar información detallada acerca del mandato Load en el apartado "Mandato Load" en la página 374.

Para obtener más información acerca de cómo se organizan los datos en las tiendas de ejemplo, consulte el Capítulo 38, "Carga de grupos de elementos de base de datos de WebSphere Commerce", en la página 405.

Antes de utilizar el mandato Load, asegúrese de que el archivo `wcs.dtd` esté en la vía de acceso (`dir_instalación_WC\schema\xml`). El archivo `massloader.cmd` también debe estar en la vía de acceso (`dir_instalación_WC\bin`).

Adición de códigos de ajuste de inventario

Para añadir códigos de ajuste de inventario para la tienda, haga lo siguiente:

1. Cree un archivo XML (con el nombre que desee) similar al del ejemplo siguiente y colóquelo en una ubicación donde pueda encontrarlo Loader. En el archivo XML, especifique los valores de los elementos `invadjcode` e `invadjdesc` para añadir el código de ajuste de inventario nuevo a dos tablas de base de datos: `INVADJ` e `INVADJDESC`. El valor de `invadjcode_id` debe ser el mismo en ambas tablas para cualquier código dado.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE import SYSTEM "wcs.dtd">
<import>
<invadjcode invadjcode_id="404" adjustcode="BRKN" storeent_id="-1"
  markfordelete="0"
<invadjdesc invadjcode_id="404" description="BROKEN" language_id="-1" />
</import>
```

donde:

- `invadjcode_id` es un identificador de código de ajuste que ha asignado. El identificador se utiliza de forma interna y no se muestra a los usuarios de la interfaz de usuario de WebSphere Commerce.
- `adjustcode` es un código de 4 caracteres que identifica de forma exclusiva el código y puede visualizarse en una interfaz de usuario.
- `storeent_id` es el identificador de la entidad de tienda o grupo de tiendas. Para añadir este código de ajuste de inventario a todas las tiendas, escriba un valor de `"-1"`. De lo contrario, proporcione una entidad de tienda o grupo de tienda específico.
- `description` es una descripción de texto del código de ajuste de inventario que puede visualizarse en una interfaz de usuario.
- `language_id` es el idioma por omisión para la información que se muestra a los clientes que compran en la tienda. Para obtener más información, consulte el Capítulo 34, "Globalización", en la página 321.

Si tiene que añadir más de un código de ajuste de inventario, puede especificar varios códigos en el archivo XML.

2. Ejecute el mandato Load para el archivo XML para cargar los datos en las dos bases de datos de destino.

Windows

```
massload.cmd -dbname nombre_bd -dbuser usuario_bd -dbpwd contraseña_bd -infile
nombre_archivo_xml -method sqlimport
```

AIX

Solaris

Linux

En primer lugar debe ser el usuario `wasuser` (el ID de usuario de WebSphere Application Server):

```
su - wasuser
```

A continuación, emita este mandato:

```
./massload.sh -dbname nombre_bd -dbuser usuario_bd -dbpwd contraseña_bd -infile
nombre_archivo_xml -method sqlimport
```

400

Inicie una sesión QShell (STRQSH). A continuación, ejecute el mandato siguiente desde el directorio `dir_instalación_WC/bin`:

```
massload.sh -dbname nombre_bd
-dbuser usuario_bd -dbpwd contraseña_bd -infile
nombre_archivo_xml -method sqlimport
```

3. Para asegurarse de que se han añadido los códigos de ajuste de inventario, ejecute una consulta en la base de datos y compruebe que puede ver los nuevos valores de código añadidos a las dos tablas.

Cambio de códigos de ajuste de inventario

Para cambiar la descripción de un código de ajuste de inventario, haga lo siguiente:

1. Siga los pasos similares a los realizados para añadir un código nuevo, pero cambie los valores de los elementos `adjustcode` y `description` en el archivo XML para el `invadjcode_id` concreto. Por ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE import SYSTEM "wcs.dtd">
<import>
<invadjcode invadjcode_id="404" adjustcode="DEFE" storeent_id="-1"
  markfordelete="0"
<invadjdesc invadjcode_id="404" description="DEFECTIVE" language_id="-1" />
</import>
```

En este ejemplo, el código de ajuste anterior y la descripción (BRKN, BROKEN) se han modificado por los valores nuevos que se muestran. Si es necesario puede cambiar también los valores de `invadjcode_id`.

Para buscar el valor de `invadjcode_id` puede emitir el mandato `Extract` (`massextract.cmd`). Puede obtener más información acerca del mandato `Extract` en la ayuda en línea al desarrollo de WebSphere Commerce. El mandato `massextract.cmd` extrae los datos de la base de datos y los coloca en un archivo XML. A continuación, puede explorar el archivo para buscar el valor de `invadjcode_id`. Como alternativa, también puede ejecutar una consulta `SELECT*` en las tablas de base de datos `INVADJCODE` e `INVADJDESC`.

2. Ejecute el mismo `massload.cmd` que para añadir un nuevo código de ajuste:

```
Windows
massload.cmd -dbname nombre_bd -dbuser usuario_bd -dbpwd contraseña_bd -infile
nombre_archivo_xml -method sqlimport
AIX Solaris Linux Como usuario wasuser, ejecute este mandato:
./massload.sh -dbname nombre_bd -dbuser usuario_bd -dbpwd contraseña_bd -infile
nombre_archivo_xml -method sqlimport
400 Inicie una sesión QShell (STRQSH). A continuación, ejecute el
mandato siguiente desde el directorio dir_instalación_WC/bin:
massload.sh -dbname nombre_bd
-dbuser usuario_bd -dbpwd contraseña_bd -infile
nombre_archivo_xml -method sqlimport
```

3. Confirme que los códigos de ajuste de inventario se han modificado en las tablas de base de datos.

Supresión de códigos de ajuste de inventario

Para suprimir un código de ajuste de inventario de las tablas de base de datos de WebSphere Commerce, efectúe lo siguiente:

1. Cree un archivo XML que contenga los códigos que desee suprimir. Para ver un ejemplo, consulte el archivo XML de ejemplo que se muestra en el apartado "Adición de códigos de ajuste de inventario" en la página 295.
2. Ejecute `massload.cmd` para asegurarse de que ha especificado el método de supresión como se indica

```
Windows
massload.cmd -dbname nombre_bd -dbuser usuario_bd -dbpwd contraseña_bd -infile
nombre_archivo_xml -method delete
```

► AIX ► Solaris ► Linux Como usuario *wasuser*, ejecute este mandato:
`./massload.sh -dbname nombre_bd -dbuser usuario_bd -dbpwd contraseña_bd -infile nombre_archivo_xml -method delete`

► 400 Inicie una sesión QShell (STRQSH). A continuación, ejecute el mandato siguiente desde el directorio *dir_instalación_WC/bin*:

```
massload.sh -dbname nombre_bd  
-dbuser usuario_bd -dbpwd contraseña_bd -infile  
nombre_archivo_xml -method delete
```

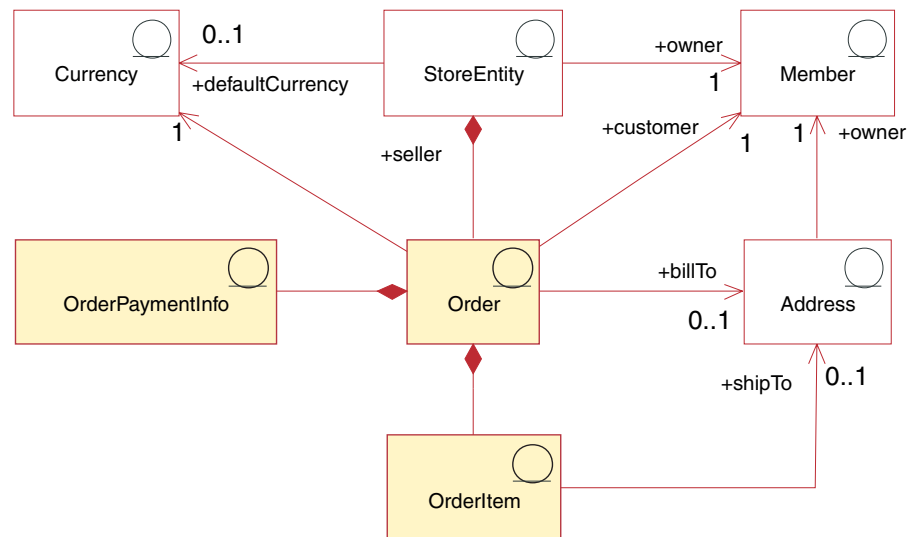
3. Confirme que los códigos de ajuste de inventario se han suprimido en las tablas de base de datos.

Capítulo 30. Elementos de pedido

Los elementos de pedido en el sistema WebSphere Commerce proporcionan funciones de carro de la compra, gestión de pedidos y proceso de pedidos. Las posibilidades de proceso de pedidos incluyen la compra o el pedido rápido, los pedidos planificados, los pedidos múltiples pendientes, la repetición de pedidos, pedidos divididos y pedidos en espera. Los servicios relacionados, por ejemplo la fijación de precios, los impuestos, el pago, el inventario y el despacho, también forman parte de los elementos de pedido.

Los elementos de pedido en WebSphere Commerce

El diagrama siguiente ilustra los elementos de pedido en WebSphere Commerce Server. A continuación del diagrama, se proporcionan descripciones de cada elemento.

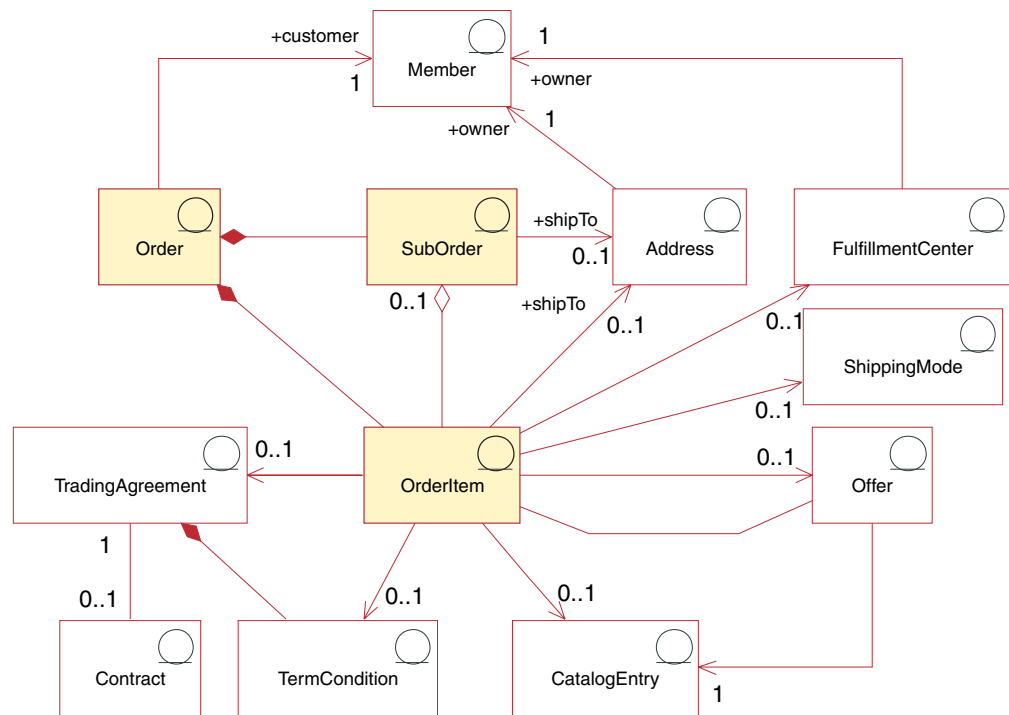


Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce Server. Para obtener más información sobre el modelo de información, consulte “Modelo de información de los datos de la tienda” en la página 105.

Pedidos y artículos de pedido

En el sistema WebSphere Commerce, para un cliente o un comprador, un *pedido* es una lista de productos seleccionados (por ejemplo, un pedido puede contener dos libros y un CD) y cada producto de esa lista es un *artículo de pedido* (por ejemplo, cada libro y CD es un artículo del mismo pedido). Cuando un cliente formaliza un pedido con la tienda, el cliente debe proporcionar una dirección de facturación a la que la tienda envía la factura. Con cada pedido se asocia un solo identificador de moneda. Desde la perspectiva de una tienda, un pedido es una lista de artículos de pedido. Forma parte de los datos de la tienda.

El diagrama siguiente ilustra los elementos de artículo de pedido de WebSphere Commerce.



Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce Server. Para obtener más información sobre el modelo de información, consulte “Modelo de información de los datos de la tienda” en la página 105.

Moneda

Una tienda puede mostrar los precios en una *moneda* o utilizar varias monedas. Cada tienda debe definir también una *moneda por omisión*. También puede permitir que los clientes seleccionen una *moneda de compra*. Si la moneda de compra es la misma que la moneda por omisión de la tienda, dicha moneda ya está soportada en la tabla STOREENT. Si la moneda de compra no es la moneda por omisión de la tienda, deberá añadir dicha moneda a la tabla CURLIST. Los clientes utilizan la moneda de compra para formalizar pedidos en la tienda.

Información de pago

Una vez que un cliente ha seleccionado una moneda de compra preferida, todos los pagos se procesarán en esa moneda. En función de las políticas y del soporte de pago de la tienda, los clientes pueden pagar las compras utilizando el pago en línea (en el que un cliente proporciona información de pago a través de Internet en el sitio de la tienda) o el pago fuera de línea (en el que el cliente proporciona información de pago sin utilizar el canal de Internet, por ejemplo por teléfono o fax). Independientemente de los métodos de pago en línea o fuera de línea, al formalizar los pedidos, los clientes deben proporcionar *información de pago*, que incluye cualquiera de los elementos siguientes:

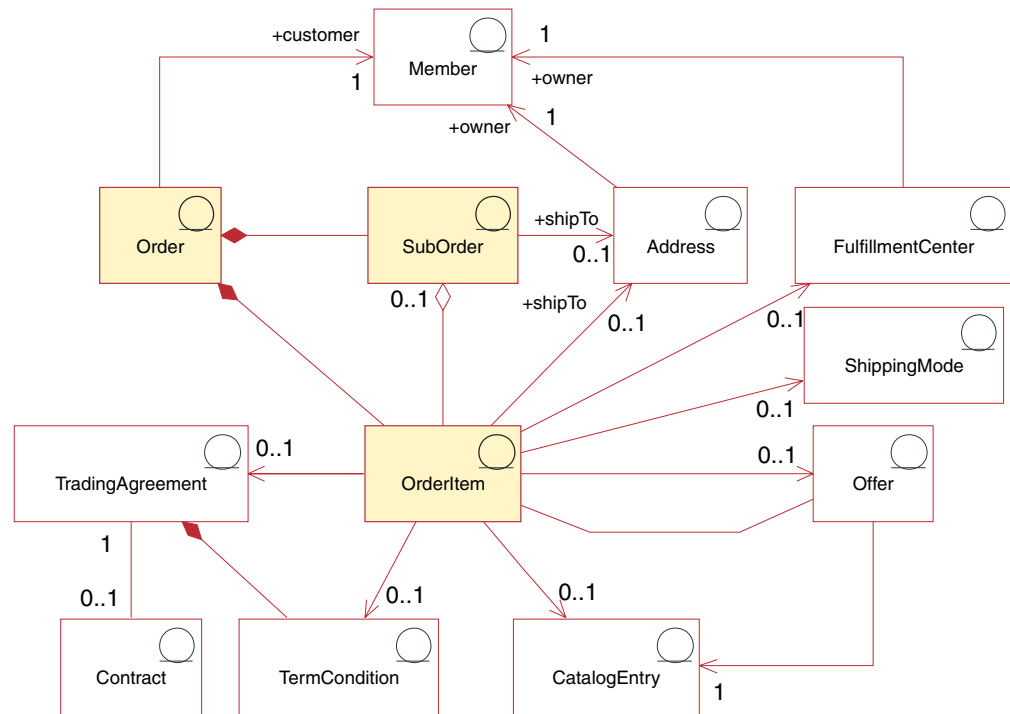
- Método de pago: El método de pago del cliente para el pedido. En función de los casetes de pago configurados en WebSphere Commerce Payments para la tienda, puede configurar la tienda de modo que acepte el pago fuera de línea, o utilizar otros protocolos de pago para los pagos en línea que no requieran que los clientes utilicen un monedero en línea, o un método de pago en línea.

- Para los pagos con tarjeta de crédito, información acerca de la tarjeta: El tipo de tarjeta, el número y la fecha de caducidad de la tarjeta de crédito del cliente utilizada para pagar el pedido. Normalmente, la información de la tarjeta de crédito es necesaria si la tienda soporta el pago en línea.
- Número de orden de compra: El número de orden de compra, que el cliente puede haber proporcionado al realizar el pedido en la tienda. El número de orden de compra autentica al cliente como cliente autorizado para realizar pedidos en la tienda, según lo estipulado en los términos del contrato entre la tienda y el cliente.

Artículos de pedido

Los *artículos de pedido* son los productos o artículos individuales que forman un pedido. Un pedido debe tener como mínimo un artículo de pedido. Cada artículo de pedido representa algo que un cliente ha seleccionado para comprar. Además, cada artículo de pedido tiene una referencia a un acuerdo comercial (generalmente un contrato), una modalidad de envío, un centro de despacho de pedidos y cualquier oferta de precio especial. Los descuentos, los costes de envío y el total de impuestos se almacenan con cada artículo de pedido.

El diagrama siguiente ilustra los elementos de artículo de pedido de WebSphere Commerce. A continuación del diagrama, se proporcionan descripciones de cada elemento.



Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce Server. Para obtener más información sobre el modelo de información, consulte “Modelo de información de los datos de la tienda” en la página 105. Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, “Descripción de UML”, en la página 463.

Subpedidos

Los artículos de pedido se agrupan para formar *subpedidos*. Un subpedido es la parte de un pedido que se está enviando a una dirección específica. Por ejemplo, un cliente puede indicar direcciones de envío diferentes para diferentes productos del carro de la compra. Cada dirección de envío y los productos asociados a la misma constituyen un subpedido. Los artículos de pedido de un subpedido tienen la misma dirección de envío y pueden utilizarse para visualizar los subtotales de los importes de los artículos de pedido.

El atributo de cantidad del objeto Artículo de pedido es un número sin unidad que se puede multiplicar por el atributo de cantidad nominal del objeto Información de envío de entrada de catálogo asociado al objeto Entrada de catálogo para llegar a la cantidad real representada por el Artículo de pedido. El objeto de información de envío de entrada de catálogo especifica la unidad de medida en la que se indican las cantidades.

Aunque los pedidos se asocian normalmente a una sola tienda, existe un tipo especial de pedido que se puede asociar a una tienda o un grupo de tiendas que es el perfil de pedido. El perfil de pedido se representa en el modelo de objeto como un Pedido con el estado de 'Q'. El perfil de pedido contiene información por omisión acerca de un cliente, por ejemplo información de pago, dirección de envío, modalidad de envío y dirección de facturación.

Otros elementos de artículo de pedido

Un artículo de pedido puede asociarse, si lo desea, a una opción de cada uno de los objetos siguientes.

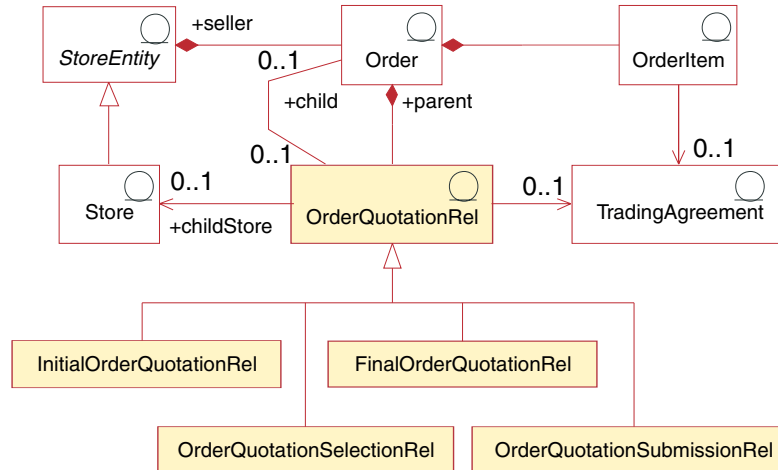
- Una dirección de envío para el cliente que ha formalizado el pedido que contiene el artículo de pedido. Un cliente debe especificar una dirección de envío durante el proceso de pedido, para que el centro de despacho de pedidos de la tienda pueda utilizar esta dirección para enviar el artículo de pedido en consecuencia.
- Un centro de despacho de pedidos para enviar y recibir los artículos de pedido necesarios para los pedidos de cliente y para almacenar inventario para el artículo de pedido.
- Una modalidad de envío para el artículo de pedido, que es una combinación de una empresa de transportes (una empresa que proporciona servicios de envío desde un centro de despacho de pedidos a un cliente) y el servicio de envío ofrecido por dicha empresa de transportes. Por ejemplo, Compañía de transportes ABC, Servicio 12 horas y Compañía de transportes ABC, Entrega urgente son modalidades de envío.
- Una oferta de precio asociada al artículo de pedido. Al incluir diferentes ofertas en listas de precios diferentes (o "contenedores de propuestas de comercio"), las tiendas pueden presentar precios diferentes para el mismo producto o código de artículo a clientes diferentes. Por ejemplo, una agencia de viajes puede ofrecer billetes de avión en cuatro listas de precios diferentes: precios para adultos, precios para personas mayores, precios para niños y precios para estudiantes.
- Una entrada de catálogo para el artículo de pedido; es decir, cada artículo de pedido solicita un artículo de un catálogo.
- Un acuerdo comercial que define los términos y condiciones bajo los cuales se realiza el pedido del artículo. Normalmente es un contrato, pero puede ser una solicitud de presupuesto (RFQ), que representa una negociación, hasta que el pedido se ha sometido a proceso.



Para obtener información más detallada sobre la estructura de los elementos de pedido en WebSphere Commerce Server, consulte los modelos de datos y de objeto de pedido en la ayuda en línea de WebSphere Commerce.

Relaciones de solicitud de presupuesto

El diagrama siguiente describe las relaciones de solicitud de presupuesto.



Para obtener información más detallada sobre la estructura de los elementos de pedido en WebSphere Commerce Server, consulte los modelos de datos y de objeto de pedido en la ayuda en línea de WebSphere Commerce.

Una relación de solicitud de presupuesto (`OrderQuotationRel`) es una relación entre un pedido de carro de la compra padre y un pedido relacionado con presupuesto hijo. En el modelo de negocio de cadena de valor (demanda), en el que la venta se realiza mediante canales, los compradores del área de canal pueden utilizar un carro de la compra para obtener presupuestos de precios de varias tiendas para productos o servicios, seleccionar cantidades de los presupuestos resultantes y enviar pedidos a las tiendas que han proporcionado los presupuestos. Las relaciones de solicitud de presupuesto indican la tienda y el contrato (acuerdo comercial) desde el que se ha solicitado el presupuesto.

Un carro de la compra padre puede tener los siguientes tipos de pedidos relacionados con presupuesto para cada pareja de tienda y contrato relacionadas con el presupuesto: un presupuesto inicial, una selección de presupuesto actual, un presupuesto final y uno o más pedidos sometidos.

El pedido hijo de una relación `InitialOrderQuotationRel` representa el presupuesto inicial recibido para los artículos de un pedido de carro de la compra padre. Un presupuesto inicial puede incluir productos alternativos y relacionados que no se mencionan explícitamente en el pedido padre.

El pedido padre de una `OrderQuotationSelectionRel` representa los elementos y las cantidades seleccionadas actualmente en el pedido inicial o final. Se puede enviar un pedido hijo `OrderQuotationSelectionRel` a su tienda para que se procese.

El pedido hijo de FinalOrderQuotationRel representa el presupuesto final recibido para los artículos de un pedido de selección. Un presupuesto final proporciona precios y cantidades sólo para los artículos del pedido de selección cuando se envía la petición para un presupuesto final.

El pedido hijo de una OrderQuotationSubmissionRel representa un pedido sometido. Cuando se somete un pedido de selección, su objeto OrderQuotationSelectionRel se cambia para que pase a ser un objeto OrderQuotationSubmissionRel.

Creación de elementos de pedido en WebSphere Commerce

Un cliente puede formalizar pedidos en una tienda o solicitar que un Representante de servicio al cliente de la tienda le ayude a realizar esta tarea (utilizando WebSphere Commerce Accelerator). Para crear un pedido en nombre de un cliente Directo al consumidor, consulte el apartado "Creación de un pedido para un cliente registrado" y "Creación de un pedido para un cliente no registrado" en la ayuda en línea de WebSphere Commerce. Para crear un pedido en nombre de un cliente Directo a B2B, consulte el tema de ayuda "Crear un pedido para un usuario de empresa".

Capítulo 31. Elementos de proveedor

Un proveedor representa una fuente de artículos recibidos en un centro de despacho de pedidos o que espera recibir en un centro de despacho de pedidos. Dependiendo del modelo de tienda, los proveedores los define el comprador, jefe de producto, vendedor (comerciante) o cualquier otro rol autorizado. Mediante WebSphere Commerce Accelerator, puede ver una lista de todos los proveedores, crear un proveedor nuevo, cambiar un proveedor existente y suprimir un proveedor.

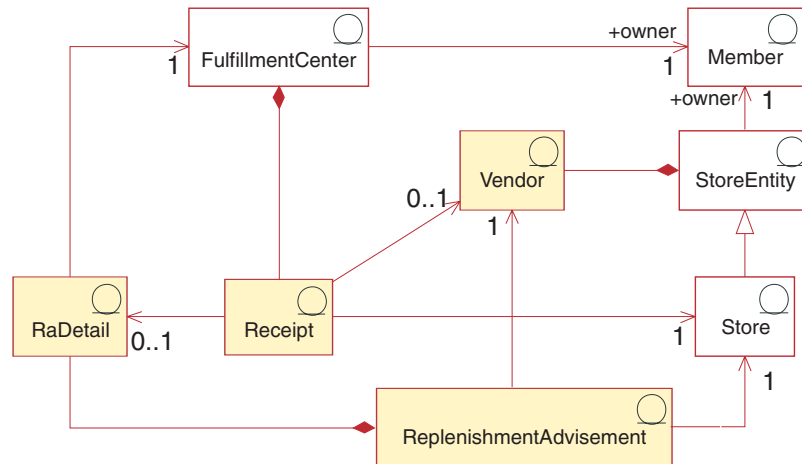
El registro de proveedores incluye información acerca del proveedor como, por ejemplo, la dirección y la información de contacto. Los proveedores deben crearse antes de crear los registros de inventario esperado.

Los registros de inventario esperado se muestran por proveedor, ID externo (generalmente el número de pedido de compra) y la fecha de pedido de la página Inventario esperado.

Para obtener más información acerca de cómo gestionar registros de proveedores, consulte los temas *Información de proveedores* de la Ayuda en línea a la producción de WebSphere Commerce.

Descripción de los elementos de proveedores de WebSphere Commerce

La sección siguiente describe las relaciones que tienen los proveedores con una tienda y otros elementos.



Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce Server. Para obtener más información sobre el modelo de información, consulte “Modelo de información de los datos de la tienda” en la página 105. Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, “Descripción de UML”, en la página 463.

El proveedor representa la fuente de la que una tienda recibe mercancías y debe considerarse como el proveedor de la tienda.

Las consideraciones de reabastecimiento son equiparables a los registros de inventario esperado.

Las recepciones hacen referencia a las recepciones de inventario. Generalmente, una recepción es el resultado de un registro de inventario esperado.

RaDetail representa información detallada acerca de los artículos de un registro de inventario esperado, como por ejemplo la fecha en que se espera el inventario, el ID del centro de despacho de pedidos y la cantidad solicitada.

Para obtener más información acerca de las recepciones de inventario, consulte el Capítulo 29, "Elementos de inventario", en la página 289.

Creación de elementos de proveedor

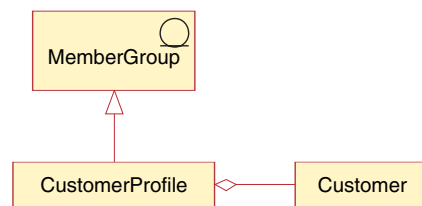
Los elementos de proveedor se pueden crear utilizando WebSphere Commerce Accelerator. Para obtener las instrucciones, consulte el tema "*Creación de un proveedor*" en la Ayuda en línea a la producción de WebSphere Commerce.

Capítulo 32. Perfiles de cliente

Los perfiles de cliente ayudan a organizar los esfuerzos de marketing ya que agrupan los objetivos de los mensajes de marketing. Normalmente, los perfiles de cliente los crea un comerciante utilizando WebSphere Commerce Accelerator.

Descripción de los perfiles de cliente en WebSphere Commerce

El diagrama siguiente ilustra los elementos de perfil de cliente en WebSphere Commerce Server:



Este diagrama, así como todos los demás que se encuentran en la sección de datos de la tienda, forma parte del modelo de información de WebSphere Commerce Server. Para obtener más información sobre el modelo de información, consulte “Modelo de información de los datos de la tienda” en la página 105. Para obtener información sobre los convenios utilizados en este diagrama, consulte el Apéndice A, “Descripción de UML”, en la página 463.

Un perfil de usuario básico para un miembro incorpora información de registro, datos estadísticos, información de dirección, historial de compras y otros atributos diversos que definen un grupo dinámico de clientes o cuentas. Los perfiles de cliente se utilizan como objetivos de anuncios, promociones, ventas por sugerencia, descuentos y actividades de correo electrónico. Debe crear perfiles de cliente antes de crear las campañas. Los perfiles se consideran dinámicos porque los clientes pertenecen a los mismos según sus datos personales y su historial de compras, los cuales pueden variar. Por ejemplo, puede crear perfiles basándose en el estado de registro de un cliente. Si crea un perfil que requiera que los clientes se registren para poder ser considerados como tales, se excluirá un cliente que no se haya registrado. Si este mismo cliente se registra en una fecha posterior, a continuación, se convertiría en miembro de dicho perfil de destino y continuaría siendo miembro hasta que se suprima el perfil.

Los perfiles de cliente también dan soporte a los criterios estáticos. Puede incluir o excluir explícitamente clientes o cuentas determinadas, lo que altera temporalmente cualquier criterio de definición. De este modo, por ejemplo, puede incluir un cliente en un perfil que, de lo contrario, no coincidiría, o excluir una cuenta de un perfil que, de lo contrario, no coincidiría. En otras palabras, si se ha definido tanto el criterio estático como el dinámico en el mismo perfil de cliente, en primer lugar se evaluará el criterio dinámico y luego el criterio estático.

Asimismo, los clientes o cuentas que puede incluir o excluir explícitamente pueden basarse en los resultados de la minería de datos de WebSphere Commerce Analyzer. WebSphere Commerce integra herramientas de analítica avanzada de

WebSphere Commerce Analyzer que permiten a los comerciantes crear fácilmente un perfil de cliente explícito basado en los segmentos que genera WebSphere Commerce Analyzer.

Parte 7. Adición de control de acceso a la tienda

Capítulo 33. Control de acceso en la tienda

WebSphere Commerce le permite determinar, mediante el control de acceso, qué tareas puede realizar un usuario determinado, ya sea un cliente o un administrador. Este capítulo describe cómo puede añadir control de acceso a la tienda, con lo que limitará las páginas que pueden ver y las tareas que pueden realizar los clientes.

Para obtener más información acerca del control de acceso en WebSphere Commerce, consulte el Capítulo 4, “Control de acceso en WebSphere Commerce”, en la página 37. Para obtener información detallada acerca del modelo de control de acceso, consulte la publicación *WebSphere Commerce, Guía de seguridad*.

Descripción del control de acceso en WebSphere Commerce

Control de acceso en las tiendas

Todas las tiendas creadas en WebSphere Commerce están sometidas a las políticas de control de acceso por omisión a las que se suscribe la organización propietaria de la tienda o las heredadas de los antecesores de dicha organización.

No obstante, la organización raíz se suscribe al grupo de políticas Gestión y administración. Si crea la tienda basándose en una de las tiendas de ejemplo que se incluyen con WebSphere Commerce, creará políticas de control de acceso y grupos de políticas que serán propiedad de la organización propietaria de la tienda. Además del grupo de políticas específico de la tienda, la organización propietaria también se suscribe a los grupos de políticas Gestión y administración, Compras comunes y B2B o B2C, dependiendo de la naturaleza de la tienda. Para obtener información acerca de la suscripción a grupos de políticas, consulte la publicación *WebSphere Commerce, Guía de seguridad*.

Cuando cree su propia tienda, independientemente de si está basada o no en un ejemplo, es posible que desee crear una política de control de acceso nueva o modificar las políticas existentes, que solamente se aplicarán a las tiendas que sean propiedad de dicha organización. Por ejemplo, si crea vistas nuevas para mostrar las páginas de la tienda, deberá asignar políticas de control de acceso a estas vistas.

Los datos de control de acceso se definen en archivos de políticas de control de acceso de alto nivel. Estos archivos definen las acciones posibles, los grupos de acciones, los recursos, los grupos de recursos y las relaciones que cualquier política puede utilizar. También definen las políticas y grupos de políticas específicos de una organización determinada. Las tiendas de ejemplo que se incluyen con WebSphere Commerce contienen estos archivos de políticas de control de acceso de alto nivel. El apartado siguiente describe cómo las tiendas de ejemplo utilizan los archivos de políticas de control de acceso para definir los grupos de políticas de control de acceso a los que se suscribe la organización.

Control de acceso en las tiendas de ejemplo

Todas las tiendas de ejemplo contienen archivos de políticas de control de acceso de alto nivel, que definen las políticas de control de acceso y los grupos de políticas creados específicamente para las tiendas. Estos datos de control de acceso son propiedad de la organización propietaria de la tienda.

Los archivos de políticas de control de acceso de alto nivel para las tiendas de ejemplo son los siguientes:

- Directo al consumidor
 - FashionFlowAccessControl.xml
- Directo a B2B
 - ToolTechAccessControl.xml
- Cadena de demanda
 - CommercePlazaAccessControl.xml (para el Centro de canal)
 - ResellerStoreFrontAssetStoreAccessControl.xml
- Cadena de oferta
 - SupplierHubAccessControl.xml
 - SupplierAssetStoreAccessControl.xml
- Alojamiento
 - CommerceHostingHubAccessControl.xml
 - HostedStoreFrontAssetStoreAccessControl.xml
 - StoreDirectoryAccessControl.xml


Estos archivos se encuentran en el directorio siguiente:

- *dir_instalación_WC /samples/stores/modelo_negocio*

Descripción de los archivos de políticas de control de acceso de las tiendas de ejemplo: Para comprender cómo se añade el control de acceso a nivel de tienda, debe familiarizarse con los archivos de políticas de control de acceso de alto nivel de las tiendas de ejemplo. Los ejemplos siguientes se han tomado del archivo

 ToolTechAccessControl.xml.

Definición de acciones: La primera sección del archivo

 ToolTechAccessControl.xml define las acciones nuevas de la tienda, que no quedan cubiertas por las políticas de control de acceso de rutina de carga. En este caso, todas las acciones son vistas que se utilizan en la tienda. Para poder visualizar una página de la tienda mediante una vista que pueda llamarse directamente desde un URL o que pueda iniciarse mediante un redireccionamiento desde otro mandato (en lugar de iniciarla mediante un reenvío a la vista), debe definirlo como una acción. Observe el ejemplo siguiente:

```
<!-- [Definiciones de inicio de acción] -->
<!-- [este es el diccionario de acciones posibles -->
<Action Name="GenericApplicationError"
  CommandName="GenericApplicationError">
</Action>

<Action Name="GenericSystemError"
  CommandName="GenericSystemError">
</Action>

<Action Name="OrderOptionsView"
  CommandName="OrderOptionsView">
</Action>

<!--[Fin de definiciones de acción] -->
```

donde

- Action Name es la etiqueta que se utiliza para hacer referencia a esta acción en el archivo XML. En estos ejemplos, la etiqueta es la misma que el nombre de vista.

- CommandName es el nombre de la vista almacenada en la columna VIEWNAME de la tabla VIEWREG. CommandName se almacenará en la columna Action de la tabla ACACTION.

Definición de grupos de acciones: La segunda sección define el grupo de acciones. El grupo de acciones es una agrupación de las acciones definidas en la primera sección del archivo. En el ejemplo ToolTech, todas las vistas de usuario nuevas se agrupan en el grupo ToolTechAllUserViews, que se utilizará en una política que permitirá a los usuarios acceder a dichas vistas, o en ToolTechRegisteredCustomerViews, que se utilizará en una política que permitirá que solamente los usuarios registrados accedan a dichas vistas.

Nota: También puede añadir acciones, definidas en cualquier otro lugar de WebSphere Commerce, a los grupos de acciones. Si se han definido en cualquier otro lugar de WebSphere Commerce, es necesario volver a definir estas acciones en la lista de acciones que se describe en el apartado “Definición de acciones” en la página 312.

```
<!-- [Inicio de definiciones de grupo de acciones] -->
<!-- Diccionario de acciones agrupadas utilizables en políticas -->
<!-- vista cruzada de componentes y grupos de acciones relacionados -->
<ActionGroup Name="ToolTechAllUsersViews"
  OwnerID="RootOrganization">

  <ActionGroupAction Name="UserRegistrationForm"/>
  <ActionGroupAction Name="UserRegistrationErrorView"/>
  <ActionGroupAction Name="GenericApplicationError"/>
  <ActionGroupAction Name="GenericSystemError"/>
  <ActionGroupAction Name="LogonForm"/>
  </ActionGroup>

<!-- [Fin de definiciones de grupo de acciones] -->
```

donde

- ActionGroup Name es el nombre del grupo de acciones. El nombre de grupo de acciones se define en la tabla ACACTGRP.
- OwnerID es el propietario del grupo de acciones. La organización raíz debe ser la propietaria del grupo de acciones. Si se utiliza cualquier otra organización, se utiliza el valor de orgentity id de dicha organización.
- ActionGroupAction Name es el nombre de una acción perteneciente a este grupo. ActionGroupAction Name debe coincidir con el nombre definido en el elemento Action Name del apartado “Definición de acciones” en la página 312. Esta relación entre acción y grupo de acciones se almacena en la tabla ACACTACTGP.

Definición de políticas: La sección siguiente define las nuevas políticas utilizadas en la tienda.

```
<!-- [Inicio de definiciones de políticas] -->

<!-- AllUsers for ToolTech can execute ToolTechAllUsersViews -->

<Policy Name="AllUsersForToolTechExecuteToolTechAllUsersViews"
  OwnerID="&seller_b2b_mbr_id;"
  UserGroup="AllUsers"
  UserGroupOwner="RootOrganization"
  ActionGroupName="ToolTechAllUsersViews"
  ResourceGroupName="ViewCommandResourceGroup"
  PolicyType="groupableStandard">
  </Policy>
```

```

<!-- RegisteredApprovedUsers for ToolTech can execute
ToolTechRegisteredApprovedUsersViews -->

<Policy Name="RegisteredCustomersForOrgForTool
TechExecuteToolTechRegisteredCustomerViews"
  OwnerID="&seller_b2b_mbr_id;"
  UserGroup="RegisteredCustomersForOrg"
  UserGroupOwner="RootOrganization"
  ActionGroupName="ToolTechRegisteredCustomerViews"
  ResourceGroupName="ViewCommandResourceGroup"
  PolicyType="groupableTemplate">
  </Policy>

<!-- [Fin de definiciones de políticas] -->

```

donde

- Policy Name es el nombre de la política que se está definiendo. La política se define en la tabla ACPOLICY.
- OwnerId es el propietario de la política. En este caso, el propietario de la política es la organización que posee la tienda.
- UserGroup es el grupo de usuarios (el grupo de acceso) al que se aplica la política.
- UserGroupOwner es el propietario del grupo de acceso. En este ejemplo, el propietario del grupo de acceso es distinto del propietario de la política. Si el propietario de la política y UserGroupOwner son el mismo, este elemento se puede omitir.
- ActionGroupName es el grupo de acciones al que se aplica la política.
- ResourceGroupName es el grupo de recursos al que se aplica la política. En este ejemplo, ya está definido el grupo de recursos, ViewCommandResourceGroup, en los datos de rutina de carga. De aquí que no sea necesario volverlo a definir en este archivo xml de alto nivel.
- PolicyType es el tipo de la política. Una política puede ser de tipo groupableStandard o groupableTemplate. Para obtener más información acerca de los tipos de política, consulte la publicación *WebSphere Commerce, Guía de seguridad*.

Definición de grupos de políticas: La sección final define el grupo de políticas específico de la tienda.

```

<PolicyGroup Name="ToolTechPolicyGroup" OwnerID="&seller_b2b_mbr_id;">
  <PolicyGroupPolicy Name="AllUsersForToolTechExecuteToolTechAllUsersViews"/>
  <PolicyGroupPolicy Name="RegisteredCustomersForOrgForToolTechExecuteTool
TechRegisteredCustomerViews"/>
  <PolicyGroupSubscription OrganizationID="&seller_b2b_orentity_id;">
</PolicyGroup

```

donde

- PolicyGroup Name es el nombre del grupo de políticas. Se define en la tabla ACPOLGRP.
- OwnerID es el propietario del grupo de políticas. En este caso, el propietario del grupo de políticas es la organización propietaria de la tienda.
- PolicyGroupPolicy Name es el nombre de la política que pertenece a este grupo. Esta relación entre política y grupo de políticas se almacena en la tabla ACPOLGPPOL.
- PolicyGroupSubscription OrganizationID es la organización que se suscribe a este grupo de políticas. Esta suscripción se almacena en la tabla ACPLGPSUBS.

Para obtener más información acerca de cómo utilizar los archivos XML para el control de acceso, consulte el apartado "Personalización de las políticas de control de acceso utilizan XML" de la publicación *WebSphere Commerce, Guía de seguridad*.

Adición de control de acceso a la tienda

Desde el punto de vista del desarrollo de tienda, los tipos de control de acceso más comunes necesarios son para las vistas nuevas y los mandatos que cree para la tienda. Sin embargo, es posible que desee añadir otros tipos de control de acceso a la tienda. Para obtener más información sobre el control de acceso para vistas, mandatos y otras características, consulte la publicación *WebSphere Commerce, Guía de seguridad*. Antes de continuar con los pasos que se describen a continuación en este manual, no olvide leer la publicación *WebSphere Commerce, Guía de seguridad*.

Si va a añadir nuevas características de control de acceso a una tienda basándose en la tienda de ejemplo, edite el archivo XML existente de política de control de acceso de alto nivel. Si va a añadir control de acceso a una tienda que no está basada en una tienda de ejemplo, deberá crear un nuevo archivo XML de políticas de control de acceso de alto nivel. Para obtener las instrucciones detalladas para ambos escenarios, consulte el apartado "Creación o edición de control de acceso en la tienda".

Creación o edición de control de acceso en la tienda

Los elementos de control de acceso son diferentes del resto de los demás elementos de la tienda en que para el control de acceso crea archivos XML de control de acceso de alto nivel y luego los transforma y los carga.

Para crear o editar elementos de control de acceso, haga lo siguiente:

1. Revise los archivos XML de alto nivel que se utilizan para crear elementos de tienda para las tiendas de ejemplo: *nombre_tienda_ejemploAccessPolicies.xml*. Estos archivos se encuentran en el directorio siguiente:

- *dir_instalación_WC/samples/stores/modelo_negocio*

Para obtener la sintaxis de los archivos xml de control de acceso, consulte el archivo DTD al que se hace referencia mediante *nombre_tienda_ejemploAccessPolicies.xml*. Los archivos DTD se encuentran en el directorio siguiente:

- *dir_instalación_WC/xml/policies/dtd*

Si va a cambiar una tienda basada en un ejemplo, edite estos archivos. Si va a añadir control de acceso a una tienda que no está basada en el ejemplo, cree un archivo de control de acceso de alto nivel nuevo para la tienda copiando uno de los archivos de tienda de ejemplo. Después de copiar el archivo, cambie el nombre y edítelo según corresponda. Para obtener más información acerca de cómo añadir control de acceso a la tienda, consulte la publicación *WebSphere Commerce, Guía de seguridad*.

2. Copie el archivo *nombre_tienda_ejemploAccessPolicies.xml* en el directorio siguiente:
 - *dir_instalación_WC/xml/policies/xml*
3. Sustituya los contenedores identificados en el diagrama siguiente por los valores reales. Para buscar los valores, efectúe los pasos siguientes:
 - a. Busque el valor del nombre distinguido (DN) utilizando la consulta siguiente: Utilice en la consulta los valores de DN que se proporcionan en el diagrama siguiente:

- select orgentity_id from orgentity where dn= <el DN de la organización>

Ejemplo: En el archivo CommercePlazaAccessControl.xml , sustituya &channel_mbr_id; y &channel_orgentity_id; por el valor que encuentre utilizando la consulta anterior, cuando el valor de DN sea ou=Organización de centro de revendedores, o=Organización de gestión de cadena de demanda, o=Organización raíz.

Tabla 11.







Nombre de archivo	Sustituya los contenedores siguientes por el valor actual	Valor de DN
CommercePlazaAccessControl.xml	&channel_mbr_id; &channel_orgentity_id;	ou=Organización de centro de revendedores, o=Organización de gestión de cadena de demanda, o=Organización raíz
	&proxy_orgentity_id;	ou=Organización de proxy de distribuidor, o=Organización de gestión de cadena de demanda, o=Organización raíz
ResellerStoreFrontAssetStoreAccessControl.xml	&profile_mbr_id;	ou=Organización de tienda con elementos, o=Organización de gestión de cadena de demanda, o=Organización raíz
SupplierAssetStoreAccessControl.xml	&profile_orgentity_id &profile_mbr_id;	ou=Organización de tienda con elementos, o=Organización de gestión de cadena de oferta, o=Organización raíz
SupplierHubAccessControl.xml	&channel_orgentity_id; &channel_orgentity_id;	ou=Organización de centro de proveedores, o=Organización de gestión de cadena de oferta, o=Organización raíz
FashionFlowAccessControl.xml	&seller_b2c_mbr_id; &seller_b2c_orgentity_id;	"ou=B2C, o=Organización vendedora, o=Organización raíz"
ToolTechAccessControl.xml	&seller_b2b_mbr_id; seller_b2b_orgentity_id;	ou=B2B, o=Organización vendedora, o=Organización raíz
CommerceHostingHublAccessControl.xml	&channel_mbr_id; &channel_orgentity_id;	ou=Organización de centro de alojamiento, o=Organización de alojamiento, o=Organización raíz
HostedStoreFrontAssetStoreAccessControl.xml	&profile_mbr_id;	ou=Organización de tienda con elementos, o=Organización de alojamiento, o=Organización raíz

Tabla 11. (continuación)

StoreDirectoryAccess Control.xml	&public_mbr_id; &public_orgentity_id;	ou=Organización de directorío de tiendas, o=Organización de alojamiento, o=Organización raíz
-------------------------------------	--	--

4. Añada la información de control de acceso correcta al archivo. Para obtener más información, consulte “Control de acceso en las tiendas de ejemplo” en la página 311 y la publicación *WebSphere Commerce, Guía de seguridad*.
5. Ejecute el mandato `acpload` para transformar el archivo, `samplestorenameAccessPolicies.xml` y cargarlo en la base de datos.

Nota: El ID de usuario de base de datos debe tener los permisos siguientes para poder ejecutar el mandato `acpload`:

- Autorización de lectura/grabación/ejecución en directorios, subdirectorios y archivos de `dir_instalación_WC/xml/policies`, `dir_instalación_WC/logs`.
 - Autorización de lectura/ejecución sobre el directorio `dir_instalación_WC/bin` y sus archivos.
- a. En un indicador de mandatos, vaya al directorio siguiente:
 - `dir_instalación_WC/bin`
 - b. Ejecute el archivo de mandatos `acpload`.
 -  **Sintaxis:** `acpload.cmd nombre_base_datos usuario_base_datos contraseña_usuario_base_datos archivo_xml_políticas [nombre esquema]`
Ejemplo: `acpload mall dbuser dbusrpwd ChannelHubAccessControl.xml`
 -     **Sintaxis:** `acpload.sh nombre_base_datos usuario_base_datos contraseña_base_datos Archivo_XML_políticas [nombre esquema]` **Ejemplo:** `acpload.sh mall dbuser dbusrpwd ChannelHubAccessControl.xml`
 - c. Consulte el archivo de anotaciones cronológicas siguiente para asegurarse de que los datos de control de acceso se han cargado correctamente:
 - `dir_instalación_WC/logs/acpload.log`
 -  `dir_usuario_WC/instances/acpload.log`



Para obtener más información sobre cómo utilizar los signos @ y &, consulte el Apéndice B, “Creación de los datos”, en la página 465.

Parte 8. Globalización de la tienda

Capítulo 34. Globalización

Soporte de la globalización

WebSphere Commerce se ha traducido a los diez idiomas siguientes:

- Inglés de EE.UU.
- Francés
- Alemán
- Italiano
- Español
- Portugués de Brasil
- Chino simplificado
- Chino tradicional
- Coreano
- Japonés

Esto incluye el software, la documentación, las interfaces de usuario y los ejemplos. Puede añadir soporte para otros idiomas. Puede traducir muchas de las características de su sitio como, por ejemplo, las descripciones de productos, mensajes y texto de las páginas. Esto se aplica a las páginas de la tienda y al navegador basado en las herramientas de WebSphere Commerce como, por ejemplo, WebSphere Commerce Accelerator y la Consola de administración.

WebSphere Commerce incluye varias características que le permiten crear un sitio que se puede adaptar a las necesidades de una base de clientes internacionales o de diferentes culturas. Utilizando la tecnología Java y un esquema de base de datos flexible, se pueden variar algunas de las características culturales del sitio, dependiendo de la ubicación o preferencia de los clientes, por ejemplo:

Idioma

Una tienda se puede visualizar en más de un idioma. Por ejemplo, es posible que desee que los usuarios seleccionen el idioma en el que desean ver su sitio o es posible que desee seleccionar automáticamente un idioma por omisión.

Moneda

Una tienda puede mostrar los precios en más de una moneda. En WebSphere Commerce, la selección de una moneda específica no está relacionada con la selección de un idioma determinado.

Formato cultural

Los datos se pueden visualizar en diferentes formatos personalizados. Los clientes de diferentes culturas pueden tener expectativas diferentes con respecto a cómo debe visualizarse determinada información. Por ejemplo, se puede indicar un número decimal mediante una coma o un punto, dependiendo del idioma, país o región del cliente.

Dirección

Las direcciones se pueden especificar, almacenar y visualizar en diferentes formatos para que se ajusten a los diferentes estándares internacionales.

Impuestos

Se puede definir la legislación sobre impuestos para diferentes

jurisdicciones. Esto incluye la legislación relacionada con impuestos sobre ventas y otros impuestos para empresas.

Envío Se pueden definir normas de envío y de empresas de transporte para diferentes jurisdicciones.

Método de pago


Se pueden definir métodos de pago para diferentes jurisdicciones.

Precios

Se puede establecer el precio en una moneda y un factor de conversión puede convertirlo desde esta moneda a otras, o se puede establecer el precio por moneda.

Contenido del catálogo en línea

Las descripciones de productos y categorías, los atributos y las imágenes pueden ser diferentes en las diferentes ubicaciones. WebSphere Commerce permite que un comerciante gestione el contenido del catálogo en línea con un formato de visualización seleccionable, definiendo cada formato dentro de la tabla de idiomas. También se puede mantener un catálogo de productos maestro que pueden compartir varias tiendas.

El diseño de la base de datos de WebSphere Commerce le ofrece la flexibilidad necesaria para crear y mantener datos reconocibles internacionalmente ya que utiliza la codificación Unicode UTF-8 o  UCS-2, que se puede convertir a la mayor parte de formatos de codificación internacionales cuando se envían al navegador Web. Puede consultar cualquiera de las tiendas de ejemplo para ver en funcionamiento un ejemplo de un sitio globalizado para muchos países.

La Consola de administración y WebSphere Commerce Accelerator dan soporte al uso de la globalización. Estas herramientas se muestran en cualquiera de los diez idiomas internacionales a los que da soporte WebSphere Commerce. Utilizan un modelo de programación que permite añadir idiomas adicionales sin afectar el funcionamiento general ni el diseño de las páginas.

Puede traducir las herramientas a otros idiomas, del mismo modo en que traduciría una tienda u otro sitio Web. Para traducir las herramientas, modifique los archivos de propiedades correspondientes situados en el directorio siguiente:
dir_usuario_WAS/installedApps/nombre_célula/nombre_instancia_WC.ear/properties/com/ibm/commerce/tools/

Tiendas de ejemplo

Las tiendas y sitios de ejemplo proporcionan una infraestructura para crear la tienda. Todas las tiendas de ejemplo le muestran cómo puede crear y mantener un sitio globalizado.

Las tiendas de ejemplo permiten a los clientes seleccionar el idioma en el que desean ver el sitio. En FashionFlow y Commerce Plaza, se selecciona en una lista el formato de visualización deseado. La lista se visualiza en una lista desplegable en el marco de la izquierda de todo el sitio. Los clientes pueden navegar por el sitio y visualizarlo en el idioma que seleccionen.

En ToolTech, se puede cambiar de idioma seleccionando el idioma preferido en la página de información personal.

En el Centro de alojamiento de Commerce y en el Centro de proveedores de la página de conexión hay una lista desplegable donde puede seleccionar el idioma.

Las tiendas de ejemplo utilizan una sola plantilla para el modelo de programación de todas las tiendas e idiomas. Cada uno de los idiomas soportados tiene su propio archivo de propiedades, que contiene el texto traducido y los mensajes para dicho idioma. Todos los archivadores de tienda contienen: `publishNLS.properties`, `publishNLS_en_US.properties`. Se utilizan simplemente en el asistente de publicación.

Para FashionFlow y las tiendas alojadas basadas en FashionFlow:

- `infashiontext_entorno_nacional.properties`
- `infashiontext_dynamic_entorno_nacional.properties`
- `infashiontext_dynamic_labels_entorno_nacional.properties` (sólo se utiliza para la GUI Cambiar páginas).
- `AuctionSample_entorno_nacional.properties`

Para ToolTech:

- `tooltechtext_entorno_nacional.properties`

Para Commerce Plaza:

- `pcdmarket_entorno_nacional.properties`

Para el Centro de proveedores de Commerce:

- `tooltechtext_entorno_nacional.properties`

Para el Centro de alojamiento de Commerce:

- `b2cHostingChannel_entorno_nacional.properties`

Para el Directorio de tiendas del ejemplo de alojamiento:

- `b2cHostingPublic_entorno_nacional.properties`

En el archivo de propiedades, se han traducido varios elementos de la página:

Texto Contenido textual de la página.

Etiquetas

Etiquetas de campos del formulario.

Mensajes

Mensajes de error, estado y confirmación.

Texto alternativo

Para imágenes, Java applets y otros objetos intercalados.

Determinación del juego de caracteres

El juego de caracteres en el que se visualiza el texto en un navegador se define en el archivo de propiedades utilizando la propiedad `ENCODING`. Por ejemplo, el archivo `infashiontext_es_ES.properties` contiene la sentencia: `ENCODING = text/html; charset=ISO_8859-1`. Dado que la codificación se especifica en el archivo de propiedades y no en la plantilla JSP, el juego de caracteres puede ser diferente para cada idioma. La codificación de caracteres de la página JSP se establece utilizando la sentencia siguiente de la plantilla JSP: `<%response.setContentType(infashiontext.getString("ENCODING")); %>` Durante la ejecución cada archivo JSP (JavaServer Page) incluye el archivo `EnvironmentSetup.jsp`. En este archivo, se recupera el contexto de mandato y partir del mismo, se utiliza el entorno nacional para recuperar el objeto java de propiedades `infashiontext`

que obtiene sus valores del archivo `infashiontext_locale.properties` del directorio específico del entorno nacional adecuado. A continuación, la plantilla tiene acceso a cada una de las propiedades necesarias, utilizando el método `getString()` del objeto `ResourceBundle`.

Nota: *Esta nota es sólo para las tiendas alojadas de Cadena de valor creadas con el Asistente de creación de tiendas.* Cuando un vendedor crea una tienda alojada, no se trata de una publicación de tienda *normal*, y los elementos de idioma de la tienda, salvo el idioma por omisión de la tienda, no se traspasan a la tienda alojada. Por lo tanto, si un vendedor añade un idioma soportado a una tienda, los elementos de tienda para ese idioma no están disponibles. Si se va a añadir un idioma soportado a una tienda alojada, asegúrese de que los elementos traducidos (archivos de propiedades) estén disponibles para la tienda o las páginas de tienda no funcionarán correctamente.

Visualización de las imágenes traducidas

El entorno nacional y el idioma se recuperan durante la ejecución para determinar la carpeta correcta en la que se ha de buscar el archivo de imágenes. La plantilla puede ser para el archivo `FashionFlow/idioma_nacional/images/go_button.gif`, donde `idioma_nacional` se sustituye por el formato de visualización del contexto de mandato. Por ejemplo, la página resultante mostrará la imagen: `FashionFlow/en_US/images/go_button.gif` o `FashionFlow/jp_JA/images/go_button.gif`.

Visualización del contenido del catálogo

El catálogo contiene varias traducciones, una para cada entorno nacional soportado. Durante la ejecución, se envía el contexto de mandatos mediante un bean de datos para determinar la traducción se ha de recuperar de la base de datos y visualizar la página.

Paquetes de recursos y archivos de propiedades

Los paquetes de recursos y los archivos de propiedades le permiten mantener grupos de objetos java que son específicos del entorno nacional para sus JSP (JavaServer Pages). Cuando la página necesita un recurso específico del entorno nacional, por ejemplo, una etiqueta de campo de formulario, un mensaje de la interfaz gráfica de usuario o un valor de un menú desplegable, la página puede cargarlo desde el archivo de propiedades o paquete de recursos apropiado para el entorno nacional seleccionado, lo que permite al cliente ver la página en su propio idioma. De este modo, puede crear plantillas JSP que dependan principalmente del entorno nacional del cliente y aislar toda la información específica del entorno nacional en los archivos de propiedades o los paquetes de recursos.

Aunque los archivos de propiedades y los paquetes de recursos efectúan funciones similares, hay algunas diferencias en cuanto al modo en que se procesan. La tabla siguiente muestra algunas de las diferencias más importantes entre los archivos de propiedades y los paquetes de recursos.

Tabla 12. Archivos de propiedades y paquetes `ListResource`

Archivos de propiedades	Paquetes <code>ListResource</code>
Archivos de texto	Formato compilado
Ligera disminución del rendimiento	Ligero aumento del rendimiento
Si se modifica un archivo de propiedades, WebSphere Application Server debe reiniciarse para poder ver los cambios.	Si se modifica un paquete de recursos, se debe volver a compilar y se debe reiniciar WebSphere Application Server para ver los cambios.

Tabla 12. Archivos de propiedades y paquetes ListResource (continuación)

Dependiente del idioma y del entorno nacional. Debe existir un archivo para cada entorno nacional. Necesita ejecutar native2ascii en los caracteres que no pertenecen al estándar ISO-8859-1.	Dependiente del idioma y del entorno nacional. Debe existir un archivo para cada entorno nacional. Necesita ejecutar native2ascii en los caracteres que no pertenecen al estándar ISO-8859-1.
---	---

Para obtener un ejemplo de cómo se pueden utilizar los archivos de propiedades en un sitio globalizado para muchos países, consulte las tiendas de ejemplo. Para obtener más información sobre estos temas, visite el sitio Web de Sun Microsystems Java.

Almacenamiento y transferencia de datos

Una sola tienda puede mostrar páginas en varios idiomas, incluso cuando utilizan diferentes juegos de caracteres. Para realizarlo, los datos se almacenan en la base de datos de WebSphere Commerce con un formato universal que se puede aplicar a un amplio número de idiomas. Dado que no todos los navegadores Web dan soporte a los mismos juegos de caracteres, cuando un JSP (JavaServer Page) solicita los datos, éstos se convierten al juego de caracteres adecuado.

A continuación, se describe cómo se trasladan los datos desde la base de datos al navegador:

1. Los datos de texto se almacenan en la base de datos de WebSphere Commerce utilizando la codificación Unicode UTF-8 o [400](#) UCS-2.
2. Los controladores JDBC cargan los datos desde la base de datos y los convierte de la codificación UTF-8 a la codificación Unicode de 16 bits nativa de Java.
3. Los archivos JSP proporcionan la salida de los datos utilizando la codificación Java de 16 bits.
4. WebSphere Application Server convierte la salida del archivo JSP de la codificación Unicode de 16 bits a la codificación de destino. La codificación se puede especificar en el archivo JSP o en un archivo de propiedades. Por ejemplo, para especificar la codificación Shift-JIS para una página en japonés, debe hacer lo siguiente:
 - Archivo JSP `<%@ page contentType="text/html; charset=Shift-JIS"%>` .
 - Archivo de propiedades `ENCODESTATEMENT = text/html; charset=Shift-JIS`
La codificación de caracteres del archivo JSP generado se establece utilizando la sentencia siguiente en la plantilla JSP:
`<%response.setContentType(fashionflowtext.getString("ENCODESTATEMENT")); %>`

Dado que no todos los navegadores pueden comprender cada esquema de codificación, le recomendamos que especifique los esquemas de codificación más conocidos como, por ejemplo, UTF-8 y Shift-JIS.

5. Los datos convertidos se devuelven al navegador.
6. El navegador interpreta la respuesta HTTP basándose en la codificación especificada en la cabecera.

A continuación, se describe cómo se trasladan los datos desde el navegador a la base de datos:

1. Los datos se introducen en el navegador. Se pueden entrar datos en varias lenguas utilizando un método de entrada.
2. WebSphere Commerce convierte los datos procedentes del navegador a la codificación Java de 16 bits utilizando el método `setCharacterEncoding()`. Cada

LANGUAGE_ID de la tabla LANGUAGE se correlaciona con un valor de codificación utilizando la columna ENCODING. Este valor se utiliza para interpretar los datos procedentes del navegador.

3. Los datos se envían a la base de datos donde se convierten de la codificación Java de 16 bits a la codificación UTF-8 que es cómo se almacenan en la base de datos.

Formatos de visualización

Los formatos de visualización permiten que una sola tienda venda a una base de clientes globalizada, de varios idiomas. Cada formato de visualización se puede identificar mediante tres factores: un idioma, una región y una variante que puede definir. Puede diseñar su sitio para que visualice diferente contenido en grupos que difieren según cualquiera de estos factores. Por ejemplo, puede utilizar el idioma y el entorno nacional para que tengan un formato diferente para inglés de EE.UU. y para inglés de Canadá. Estos formatos de visualización pueden tener el mismo texto pero diferentes monedas y unidades de medida. Puede añadir un formato de visualización para francés de Canadá. Este formato de visualización puede mostrar la misma moneda y unidades de medida que el inglés de Canadá pero el texto estará en francés de Canadá. Incluso puede utilizar el tercer factor para tener un formato diferente para los tipos diferentes de clientes dentro de una cultura como, por ejemplo, adolescentes, la comunidad científica o los profesionales técnicos, adaptando el sitio a sus expectativas.

Los clientes pueden seleccionar el formato en el que desea ver el sitio o puede establecer un valor por omisión para ellos. La información relacionada con el formato se pasa mediante un parámetro de URL cuando desea cambiar el idioma. Por ejemplo, si pasa langId=-2, la sesión establecerá el idioma actual en francés. El valor de langID se almacena en la sesión. Cuando el cliente solicita una página, el formato de visualización determina los elementos Web y la información de catálogo que se ha de recuperar.

Creación de un formato de visualización nuevo

Para crear un formato de visualización nuevo, haga lo siguiente:

1. Ejecute el mandato siguiente: `select * from language`. Este mandato devuelve los ID de idioma que utilizan actualmente los formatos de visualización disponibles. Seleccione el ID_VALUE siguiente que esté disponible.
2. Ejecute el mandato siguiente: `insert into language (LANGUAGE_ID, ENCODING, LOCALENAME, LANGUAGE, COUNTRY) values (ID_Value, ENCODING_VALUE, 'x', 'y', 'z')` donde:

ID_VALUE

El valor que ha seleccionado en el paso 2.

LANGUAGE_ID (obligatorio)

Un identificador que identifica de forma exclusiva el formato de visualización.

ENCODING_VALUE (obligatorio)

El valor de codificación de caracteres que debe utilizar el navegador para mostrar la página para este idioma. Debe ser igual que el valor de codificación utilizado en los archivos de propiedades.

`ENCODING_STATEMENT = text/html; charset=[ENCODING_VALUE]`.

Puede encontrar una lista de los valores de codificación soportados por Sun JDK en el sitio de Sun Java: www.java.sun.com.

LOCALENAME (obligatorio)

Se utiliza un entorno nacional java para representar una región política, geográfica o cultural que tenga un idioma y costumbre diferentes para su formato. El nombre del entorno nacional es el código de idioma ISO 639 de dos letras, seguido por el código de país ISO 3166 de dos letras, separados por el carácter de subrayado.

LANGUAGE (opcional)

El nombre del idioma.

COUNTRY (opcional)

El país o región para el formato de visualización.

VARIANT (opcional)

La columna VARIANT es una columna adicional que permite describir un subgrupo dentro de una cultura determinada como, por ejemplo, adolescentes, técnicos u cualquier otra clasificación.

Por ejemplo, para añadir un formato de visualización para el italiano que se habla en Estados Unidos, puede ejecutar la sentencia: insert into language (LANGUAGE_ID, ENCODING, LOCALENAME, LANGUAGE, COUNTRY) values ('333', 'ISO8859-1', 'it_US', 'Italian', 'United States'). Es posible que desee también especificar un idioma alternativo para lo que puede consultar *Ejemplo de creación de un formato de visualización nuevo*.

3. Añada una entrada a la tabla LANGUAGEDS. Para obtener un ejemplo, consulte el *Ejemplo de creación de un formato de visualización nuevo*.
4. Añada una entrada a la tabla LANGPAIR. Para obtener un ejemplo, consulte el *Ejemplo de creación de un formato de visualización nuevo*.
5. Añada un idioma a una tienda. Para obtener información detallada acerca de cómo añadir un idioma a la tienda, consulte el apartado "Adición de un idioma a una tienda" en la página 334

Ejemplo de creación de formato de visualización nuevo

El ejemplo siguiente le muestra cómo puede crear un formato de visualización para ver las páginas de la tienda de ejemplo FashionFlow en tailandés.

1. Traduzca el archivo infashiontext_locale.properties a tailandés.
2. Asegúrese de que la sentencia de codificación del archivo de propiedades haga referencia al juego de caracteres al que dan soporte los navegadores de destino. Para tailandés, la sentencia de codificación es la siguiente:
ENCODESTATEMENT = text/html; charset=MS874
3. Guarde el archivo como infashiontext_th_TH.properties
4. Abra una ventana de mandatos DB2.
5. Ejecute el mandato siguiente: select * from language Este mandato devuelve los ID de idioma que utilizan actualmente los formatos de visualización disponibles. Seleccione el ID_VALUE siguiente que esté disponible. En este ejemplo, el valor de ID_VALUE para tailandés es 3.
6. Ejecute el mandato siguiente: insert into language (LANGUAGE_ID, ENCODING, LOCALENAME, LANGUAGE, COUNTRY, MIMECHARSET) values (ID_Value, ENCODING_VALUE, 'w', 'x', 'y', 'z'), utilizando los valores siguientes: insert into language (LANGUAGE_ID, ENCODING, LOCALENAME, LANGUAGE, COUNTRY) values ('3', 'MS874', 'th_TH', 'Thai', 'Thailand', 'MS874') donde:

ID_VALUE

El valor que ha seleccionado en el paso 2.

LANGUAGE_ID (obligatorio)

Un identificador que identifica de forma exclusiva el formato de visualización.

ENCODING_VALUE (obligatorio)

El valor de codificación de caracteres que debe utilizar el navegador para mostrar la página para este idioma. Puede encontrar una lista de los valores de codificación soportados por Sun JDK en el sitio de Sun Java: www.java.sun.com. El valor de codificación debe estar soportado por Sun JDK.

LOCALENAME (obligatorio)

Se utiliza un entorno nacional java para representar una región política, geográfica o cultural que tenga un idioma y costumbre diferentes para su formato. El nombre del entorno nacional es el código de idioma ISO 639 de dos letras, seguido por el código de país ISO 3166 de dos letras, separados por el carácter de subrayado. Para los códigos de idioma ISO, consulte el sitio Web de International Standards Organization: www.iso.ch.

LANGUAGE (opcional)

El nombre del idioma.

COUNTRY (opcional)

El país o región para el formato de visualización.

MIMECHARSET (opcional)

El juego de caracteres que se utiliza en la mensajería MIME.

VARIANT (opcional)

La columna VARIANT es una columna adicional que permite describir un subgrupo dentro de una cultura determinada como, por ejemplo, adolescentes, técnicos u cualquier otra clasificación.

7. Añada una entrada a la tabla LANGUAGES:
languages language_id=-1, description=su_idioma_en_ingles,
language_id_desc=-11.
Por ejemplo: "languages language_id=-1, description=French,
language_id_desc=-11, languages language_id=-11,
description=su_idioma_en_su_propio_idioma, language_id_desc=-11, desc=-11.
Por ejemplo: "languages language_id=-11, description=Francés,
language_id_desc=-11.
8. Cree un idioma alternativo para el tailandés que se utilizará si los datos solicitados no están en tailandés. Esto resulta útil si no todos los datos de la base de datos se traducen al nuevo idioma. Para crear un idioma alternativo ejecute el mandato siguiente: insert into langpair(LANGUAGE_ID, LANGUAGE_ID_ALT, SEQUENCE, STOREENT_ID) values (ID_Value, ID_Value_ALT, 'x', 'y'), utilizando los valores siguientes: insert into langpair(LANGUAGE_ID, LANGUAGE_ID_ALT, SEQUENCE, STOREENT_ID) values ('3', '-1', '1' '12345'), donde LANGUAGE_ID es el idioma solicitado. LANGUAGE_ID_ALT es el idioma alternativo. SEQUENCE: cuando el idioma solicitado está soportado según lo especifica la tabla STORELANG pero la información no está disponible en dicho idioma, se intenta utilizar cada uno de los idiomas alternativos por orden ascendente del valor de SEQUENCE. Una tienda puede alterar temporalmente el valor de SEQUENCE especificado para su StoreGroup. STOREENT_ID: El valor de StoreEntity al que pertenece esta relación. Las relaciones de idiomas alternativos para una tienda incluyen las relaciones de idiomas alternativos para su StoreGroup. La sentencia de inserción anterior asignará el idioma inglés (language id = -1) como el primer

idioma alternativo que se intentará utilizar para la tienda con id '12345' en caso de que no se encuentren datos en tailandés.

9. Convierta cualquier archivo de propiedades de nativo a ascii: Copie `infashiontext_th_TH.properties` en un directorio temporal (por ejemplo, `/tmp`). Ejecute el mandato siguiente: `dir_JDK/bin/native2ascii -encoding TIS620 /tmp/infashiontext_th_TH.properties /tmp/infashiontext_th_TH_new.properties`. Copie esto: `/tmp/infashiontext_th_TH_new.properties` en este directorio: `/dir_usuario_WAS/installedApps/nombre_célula/instancia_WC.ear/Stores.war /WEB-INF/classes/dirTienda/infashiontext_th_TH.properties`. Donde `dir_JDK` es la vía de acceso a su JDK.

Entrada de datos en varios idiomas

Puede visualizar varios idiomas simultáneamente en un navegador, si el navegador da soporte a Unicode. Tanto Netscape Navigator como Internet Explorer Versión 4 o superior dan soporte a la visualización Unicode. No obstante, es posible que el sistema operativo no tenga los caracteres que necesita para escribir el texto en algunos idiomas. Para hacerlo, es posible que necesite un método de entrada. Un método de entrada es un componente de software que convierte la entrada mediante pulsaciones de tecla en entrada de texto que no se puede escribir directamente. Los métodos de entrada se utilizan normalmente para el texto de entrada para idiomas que tienen más caracteres de los que caben en un teclado estándar como, por ejemplo, japonés, chino y coreano, tailandés e hindi. Una herramienta de método de entrada común es Microsoft Global Input Method Editor que está disponible en el sitio Web de Microsoft: Global IME es adecuado tanto para que los clientes entren datos en las páginas de compra como para que los administradores entren datos en WebSphere Commerce Accelerator y en la Consola de administración.

Si opta por no utilizar un método de entrada, también puede entrar datos en los diferentes idiomas proporcionados siempre que tenga una configuración de varias máquinas, cada una de las cuales utilice un idioma diferente en el sistema operativo con un navegador configurado correctamente. Un navegador dará soporte automáticamente al idioma nativo de la máquina en la que está instalado. Por ejemplo, para escribir datos en japonés y datos en alemán, puede configurar dos máquinas, una de las cuales utilice un sistema operativo alemán y el otro un sistema operativo japonés, cada uno con un navegador que pueda visualizar datos desde dicho sistema operativo. Para obtener más información sobre este tema, consulte la documentación del sistema operativo o navegador Web.

Unicode: Los datos de texto de WebSphere Commerce se codifican utilizando el juego de caracteres Unicode. Unicode puede visualizar los caracteres utilizados en los idiomas importantes, incluidos los idiomas europeos, de Oriente Medio y Asia. En WebSphere Commerce, se utiliza el estándar Unicode UTF-8 para almacenar los datos en varios idiomas en la misma instancia de base de datos. Aunque los clientes no necesitan tener un navegador habilitado para Unicode para ver sitios dirigidos por WebSphere Commerce, es posible que los administradores necesiten uno si desean ver el sitio en más de un idioma en la misma máquina. Si desea ver el sitio en un idioma distinto al idioma inglés, necesita un navegador habilitado para Unicode. Para obtener más información sobre Unicode, visite el sitio Web de Unicode.

Creación de una tienda globalizada

Para crear una tienda globalizada, efectúe lo siguiente:

1. Cree una tienda.
2. Gestione la plantilla.
3. Añada un idioma a la tienda.
4. Cree un catálogo globalizado.
5. Gestione los elementos globalizados.
6. Traduzca sus archivos de propiedades.

Creación de una tienda

Puede crear una tienda publicando uno de los archivadores de tienda de ejemplo y editando la tienda resultante, o puede crear el escaparate, la lógica de negocio o los elementos de datos por separado.

- Escaparate: La parte externa de la tienda, o la parte que se muestra a los clientes, se conoce como escaparate. El escaparate consta de elementos Web tales como páginas HTML, archivos JSP, hojas de estilo, imágenes, gráficos y otros tipos de archivos multimedia. Para obtener más información, consulte la Parte 4, “Desarrollo del escaparate”, en la página 79.
- Lógica de negocio: La parte de la tienda que procesa peticiones de cliente, incluidos los mandatos y el código personalizado se conoce como lógica de negocio. Para obtener información más detallada sobre la creación de lógica de negocio o código personalizado, consulte la publicación *WebSphere Commerce, Guías de programación y aprendizaje*.
- Datos de tienda: Los elementos de datos que forman la tienda. Para que una tienda funcione correctamente debe tener los datos en una ubicación que dé soporte a todas las actividades de los clientes. Por ejemplo, para que un cliente efectúe una compra, la tienda debe contener un catálogo de los productos en venta y debe incluir un proceso para el manejo de pedidos, el inventario para satisfacer el pedido y un proceso de envío. También debe tener métodos para procesar y recoger los pagos. Los conceptos y las tareas involucradas en la creación de datos de tienda se tratan en la Parte 6, “Desarrollo de los datos de la tienda”, en la página 117.

Para obtener más información sobre cómo publicar un archivador de tienda de ejemplo, consulte el tema ‘Publicar un archivador de tienda’ de la Ayuda en línea a la producción de WebSphere Commerce.

Gestión de la plantilla para un sitio globalizado

Para gestionar las páginas estáticas y las plantillas dinámicas de un sitio globalizado, es necesario almacenar archivos en una estructura de directorio que le permite identificar fácil y rápidamente los archivos y el entorno nacional al que pertenecen.

La vía de acceso del directorio de archivos se crea basándose en la instancia de WebSphere Commerce, la vía de acceso de tienda contenida en el perfil de la tienda y también la vía de acceso a archivos registrada. Cuando crea un sitio globalizado, crea varias tiendas, cada una de las cuales representa una jurisdicción de envío soportada por el sitio y contiene una lista de los idiomas soportados. Dado que los archivos de plantilla afectan el diseño de un sitio, se deben almacenar bajo los directorios específicos del entorno de nacional, de modo que se puedan seleccionar del mismo modo en que se seleccionan los paquetes de recursos utilizando un valor de entorno nacional. Cuando el sistema selecciona la

plantilla que se utilizará para un formato de idioma determinado, se utiliza el entorno nacional para determinar el formato de idioma que se utilizará para determinar el directorio desde el que se recuperará el archivo.

Hay tres modelos de almacenamiento de plantillas en un entorno globalizado:

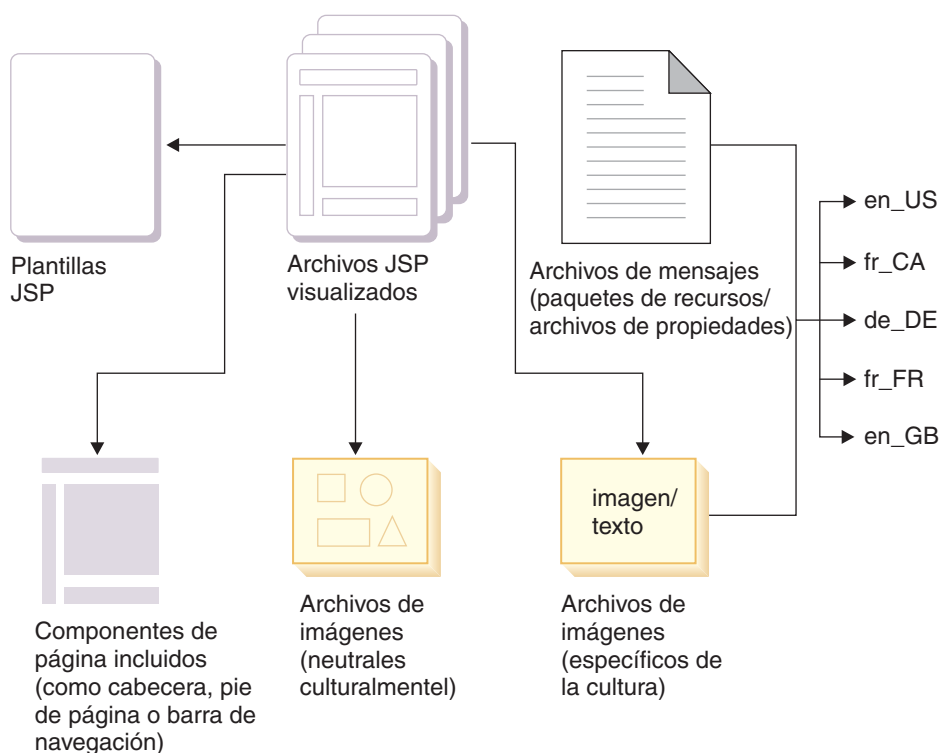
Tabla 13. Almacenamiento de plantillas en un entorno globalizado

	Una plantilla para todas las tiendas e idiomas	Una plantilla por idioma	Una plantilla por tienda
Personalización	En la mayor parte de las tiendas, proporciona los niveles suficientes de personalización entre cada tienda y cada formato de idioma de la tienda.	Permite el nivel máximo de personalización entre cada tienda y cada formato de idioma.	Determinado nivel de personalización en cada tienda.
Diseño de las páginas	Las páginas parecerán similares.	Las páginas pueden ser muy diferentes.	Las páginas tienen el mismo diseño general.
Mantenimiento	Permite realizar fácilmente cambios en el diseño de las páginas de todo el sitio ya que sólo es necesario cambiar una plantilla. En la mayor parte de sitios globalizados, este modelo proporciona los niveles óptimos de mantenimiento y escalado.	Debe gestionar varias copias de cada plantilla. Los cambios que afectan a todas las tiendas o todos los formatos de tienda tienen que realizarse en cada plantilla.	Los cambios realizados en el diseño de un archivo JSP que afecten a todo el sitio tendrán que realizarse en varias plantillas.
Cuándo se ha de utilizar	Se ha de utilizar cuando el diseño de cada tienda y cada idioma sea muy similar.	Utilícelo cuando el diseño y el contenido de la página sea muy diferente entre idiomas. En este caso, no hay mucho que pueda compartirse entre los idiomas y resulta más fácil desarrollar páginas diferentes para cada idioma.	Utilícelo cuando el diseño de las tiendas sea realmente muy diferente aunque, independientemente del idioma, el diseño de la tienda se conserve relativamente igual.
Cuándo no se ha de utilizar	No lo utilice si el sitio ha de ser muy diferente en las diferentes tiendas e idiomas.	No lo utilice si las páginas son muy similares entre tiendas y entre formatos de idioma.	No lo utilice si el diseño de las tiendas es muy similar.

Tabla 13. Almacenamiento de plantillas en un entorno globalizado (continuación)

Archivos de propiedades	Obligatorio. Cada idioma soportado tiene también su propio archivo de propiedades que se incluye cuando se genera la página.	No es obligatorio. Cada combinación de tienda y entorno nacional tiene su propia plantilla JSP (JavaServer Page).	Obligatorio. Para que se puedan compartir las plantillas en los diferentes formatos de idioma.
Flujo de compra	El flujo de compra entre idiomas y tiendas permanece igual.	El flujo de compra puede variar de modo significativo entre idiomas.	El flujo de compra entre idiomas y tiendas permanece igual.

Modelo de programación de una plantilla para todas las tiendas e idiomas



En el modelo de programación de una plantilla para todas las tiendas e idiomas, cada página consta de una sola plantilla JSP (JavaServer Page) que contiene un diseño de página básico y datos e imágenes neutras culturalmente. Esta plantilla se combina durante la ejecución con componentes sensibles a la cultura, basándose en el formato de visualización que ha seleccionado el cliente. Los cambios realizados en el diseño de una página sólo es necesario realizarlos una vez, independientemente del número de culturas soportado. Añadir o suprimir idiomas o culturas es sencillo ya que la parte sensible a la cultura se separa de las demás características de la página.

La tabla siguiente muestra cómo se pueden organizar los archivos. Tenga en cuenta que *webapp* hace referencia al directorio raíz del sitio o aplicación. En dicho directorio puede tener un directorio común y un directorio para cada formato de visualización o combinación de idioma y entorno nacional soportado. La estructura exacta depende de usted. Durante la ejecución, la plantilla utiliza la información de

idioma y entorno nacional del contexto de mandatos y la utiliza para determinar la carpeta adecuada desde la que recuperará el archivo de propiedades, los archivos de imágenes y cualquier otro contenido específico de la cultura. Por ejemplo, si el contexto de mandatos indica un formato de visualización es_ES, utilice el archivo raíz_sitio/es_ES/sensitivetext.properties y recuperará las imágenes del directorio raíz_sitio/es_ES/images/.

Tabla 14. Modelo de programación de una plantilla para todas las tiendas e idiomas

Plantillas	/webapp/common/web/template/template.jsp Se utiliza la misma plantilla para todos los formatos de visualización.
Componentes de página incluidos	/webapp/common/web/template/header.jsp /webapp/common/web/template/footer.jsp Los componentes de página comunes se encuentran en este directorio.
Archivos de imágenes neutrales culturalmente	/webapp/common/web/images/image.gif Las imágenes están en un directorio común y se utilizan para todos los formatos de visualización.
Archivos de propiedades (seleccione cualquiera de los siguientes métodos para almacenar los archivos de propiedades)	/webapp/language_LocaleA/web/sensitivetext.properties /webapp/language_LocaleB/web/sensitivetext.properties Cada formato de visualización tiene un archivo de propiedades diferente. Los archivos de propiedades de los diferentes formatos de visualización tienen el mismo nombre base, pero se añade el sufijo de entorno nacional xx_XX al nombre antes de la extensión de archivo. Se encuentran en el mismo directorio. El nombre del directorio está basado en la combinación <i>Idioma_Entorno_nacional</i> como aparece en la columna LOCALENAME de la tabla LANGUAGE. Para obtener un ejemplo de este método, consulte una tienda de ejemplo. O puede utilizar este método: /webapp/properties/sensitivetext_Language_LocaleA.properties /webapp/properties/sensitivetext_Language_LocaleB.properties Los archivos de propiedades se almacenan en un sólo directorio pero tienen nombres de archivo específicos del entorno nacional.
Archivos de imágenes específicos de la cultura	/webapp/language_LocaleA/web/images/image.gif /webapp/language_LocaleB/web/images/image.gif Se almacena una imagen traducida diferente para cada formato de visualización. Los archivos tienen el mismo nombre pero se encuentran en directorios diferentes, que corresponden al nombre del formato de visualización al que se aplican. La combinación <i>idioma_entorno_nacional</i> representa el formato de visualización, según se muestra en la columna LOCALENAME de la tabla LANGUAGE.

Estructura de directorio de una plantilla para todas las tiendas e idiomas

Este modelo es adecuado cuando el diseño de cada tienda y cada idioma es muy similar. Solamente necesita mantener un conjunto de plantillas JSP pero debe gestionar una serie de archivos de propiedades. Este método es el modelo de gestión de plantillas que se ha de mantener y permite realizar cambios en el diseño de las páginas de todo el sitio ya que sólo es necesario modificar una plantilla.

Por ejemplo, si tiene dos ubicaciones de tienda, cada una de las cuales visualiza inglés de Estados Unidos y francés del Canadá, puede organizar las plantillas JSP como se indica a continuación: /webapp/common/web/template/abc.jsp.

La vía de acceso de los archivos de propiedades de esta plantilla JSP se almacenarán del modo siguiente:

```
/webapp/common/web/properties/en_US/abc.properties  
/webapp/common/web/properties/fr_CA/abc.properties
```

En este caso, cuando se registren archivos JSP, solamente es necesario incluir el tipo de archivo en el registro de archivos. Utilizando este método, sólo es necesario registrar un conjunto de archivos JSP para todas las tiendas y entornos nacionales. De aquí que los archivos de propiedades se deban almacenar por separado porque contiene información sensible a la cultura, mientras que la plantilla propiamente dicha, que es completamente neutral, se almacena en un directorio común. Para obtener un ejemplo en funcionamiento de esta estrategia de gestión de plantillas, consulte una tienda de ejemplo.

Estructura de directorio de una plantilla por idioma

Para utilizar este modelo, debe crearse un directorio diferente dentro de cada directorio de plantillas para cada idioma soportado para dicha tienda. Se debe almacenar una plantilla diferente dentro de estos directorios dependientes del idioma. No son necesarios archivos de propiedades en este modelo, ya que cada combinación de tienda e idioma nacional tiene su propia plantilla JavaServer Page.

Por ejemplo, si tiene dos ubicaciones de tienda, cada una de las cuales muestra inglés de Estados Unidos y francés de Canadá, puede organizar las plantillas JSP del modo siguiente: `/webapp/StoreA/web/template/en_US/abc.jsp`
`/webapp/StoreA/web/template/fr_CA/abc.jsp`
`/webapp/StoreB/web/template/en_US/abc.jsp`
`/webapp/StoreB/web/template/fr_CA/abc.jsp` En este caso, cuando registre la plantilla JSP, el entorno nacional y el tipo de archivo se tendrán que incluir en el registro de archivos. Cada tienda y cada entorno nacional debe tener un conjunto completo de plantillas registradas.

Estructura de directorio de una plantilla para cada tienda

Las plantillas JSP de este modelo de compra se comparten en una tienda, pero son exclusivas de una sola tienda. Los archivos de inclusión JSP para que se puedan compartir las plantillas en los diferentes formatos de idioma.

Por ejemplo, si tiene dos ubicaciones de tienda, cada una de las cuales visualiza inglés de Estados Unidos y francés del Canadá, puede organizar sus plantillas JSP del modo siguiente: `/webapp/StoreA/web/template/abc.jsp`
`/webapp/StoreB/web/template/abc.jsp`

La vía de acceso para los archivos de propiedades dentro de este modelo de gestión de plantilla se parecerán a este ejemplo:

```
/webapp/StoreA/web/properties/en_US/abc.properties  
/webapp/StoreA/web/properties/fr_CA/abc.properties  
/webapp/StoreB/web/properties/en_US/abc.properties  
/webapp/StoreB/web/properties/fr_CA/abc.properties
```

Cuando se registran las plantillas JSP para este modelo, solamente el tipo de archivo necesita incluirse en el registro de archivos. Cada tienda tendrá que registrar su propia lista completa de archivos JSP.

Adición de un idioma a una tienda

Para añadir soporte para un idioma nuevo en una tienda existente, haga lo siguiente:

1. Asegúrese de que el idioma esté disponible en el sitio. Para obtener una lista de los diez idiomas soportados, consulte el soporte de varios idiomas. Si el idioma está disponible, vaya al paso 3, de no ser así, vaya al paso siguiente.

2. Cree un nuevo formato de visualización para el idioma. Utilice el cuadro Perfil de tienda para añadir el idioma a la lista de los idiomas a los que da soporte la tienda.
3. Copie el archivo del idioma nacional, por ejemplo, para FashionFlow será `infashiontext_locale.properties` en esta ubicación:
`AppServer/installedApps/host/WC_demo1.ear/Stores.war/WEB-INF/classes/storeDir`
 Hay muchos otros archivos XML que se han de traducir y rellenar utilizando Loader, por ejemplo, `asw tax.xml`, `store.xml`, `fulfillment.xml`, `catalog.xml`, `businesspolicy.xml`, `contract.xml`, `accesscontrol.xml`, `shipping.xml`.

Creación de un catálogo en línea globalizado

Para crear un catálogo en línea globalizado adecuado para un sitio globalizado, incluya varios detalles acerca de cada producto, uno para cada idioma o cultura a la que desea dar soporte. Tenga en cuenta que a menudo hay diferencias entre las culturas que van más allá de simplemente el idioma como, por ejemplo, el modo en que se representan determinados datos. Por ejemplo, en algunas culturas, se representa un número decimal mediante una coma, mientras que en las demás se representa utilizando un punto.

1. Para cada idioma a la que dé soporte la tienda debe crear un catálogo. Seleccione uno de los métodos de creación de catálogo siguientes:
 - Cree un catálogo utilizando Loader package o una herramienta de catálogo de su elección.
 - Cree los datos de catálogo en los archivos XML y cárguelos en la base de datos utilizando Loader package, o publíquelos en un formato de archivador de tienda utilizando la Consola de administración. Para obtener más información, consulte el apartado sobre Creación de un catálogo.
 - Convierta el catálogo existente a un formato de archivo XML, adecuado para utilizarlo con Loader y luego cargue la información en la base de datos. Para obtener más información, consulte Loader package.
 - Cree el catálogo utilizando los catálogos de tienda de ejemplo como base y cambie la información utilizando las herramientas de gestión de productos (esto funciona solamente para pequeñas cantidades de datos).
2. Para cada catálogo que cree tenga en cuenta cómo presentará los tipos de información siguientes:

Productos del catálogo en línea

Hay varias descripciones de idioma para cada entrada de catálogo.

Descripciones de productos

Se puede variar el idioma y las frases de las descripciones, resaltando las diferentes características para grupos diferentes de clientes.

Precios

Los precios se pueden variar para que reflejen las tarifas y los demás costes de envío y se pueden expresar en monedas diferentes.

Formatos culturales

Las fechas, nombres y unidades de medida y otros datos se pueden formatear para adaptarse a las expectativas culturales.

Imágenes de productos

Es posible que desee visualizar varias imágenes de productos a clientes diferentes.

Gestionar elementos de globalización

Para gestionar los elementos Web, se recomienda emplear un modelo de programación de globalización que utilice una plantilla JSP para todas las tiendas e idiomas incluido el diseño básico de cada página junto con cualquier información cultural que sea neutral. El texto sensible a la cultura que queda se añade a las páginas durante la ejecución utilizando los paquetes de recursos o los archivos de propiedades.


Determine qué elementos Web se han de traducir. Esta lista puede incluir banners, imágenes, applets, texto, mensaje y otro contenido sensible culturalmente en las páginas. Cree varias versiones de algunos de estos componentes, uno para cada idioma o cultura a la que dé soporte su sitio. Para obtener un ejemplo de cómo se pueden gestionar los elementos en una tienda globalizada, consulte una tienda de ejemplo.

Traducción de archivos de propiedades

Para traducir archivos de propiedades, haga lo siguiente:

1. Abra el archivo de propiedades utilizando cualquier editor de texto.
2. Traduzca el texto del archivo de propiedades, teniendo en cuenta lo siguiente:
 - No traduzca la palabra clave. La palabra clave es el contenido que hay a la izquierda del signo de igual. Source: lastName.Label=Last Name Translation: lastName.Label=Nom de famille
 - Para los atributos de opciones, traduzca solamente los valores que haya a la derecha del signo de punto y coma (;). Source: title.Options=MR;Mr. | MRS;Mrs. | MS;Ms. Translation: title.Options=MR;M. | MRS;Mme. | MS;Mlle. Source: publishPhone.Options=Y;Yes | N;No Translation: publishPhone.Options=Y;Oui | N;Non
 - Opcionalmente, traduzca los comentarios, esto es, cualquier línea que comience por un signo numérico (#).
3. Guarde el archivo de propiedades como texto. Si va a utilizar un modelo de programación en el que:
 - a. Los archivos de propiedades tienen el mismo nombre pero se almacenan en directorios específicos del entorno nacional.
 - Guarde el archivo en el directorio correcto
 - b. Los archivos de propiedades se almacenan en el mismo directorio pero el entorno nacional se añade al nombre
 - Añada el entorno nacional adecuado al nombre de archivo. La extensión debe ser .properties
4. Si el archivo de propiedades contiene caracteres que no son Latin 1 y Unicode, utilice el convertidor native2ascii para convertir datos desde un formato que no es ascii a representaciones ascii de Unicode. Este proceso hará que los datos que contenga el archivo de propiedades sean independientes de la plataforma. El convertidor native2ascii está en el directorio siguiente:

`dir_instalación_WC\jdk\bin`

 `400` `dir_instalación_WC/Java400/jdk13/bin`

Para obtener información adicional acerca del convertidor native2ascii, consulte el sitio siguiente: www.java.sun.com

Parte 9. Empaquetado de la tienda

Capítulo 35. Empaquetado de una tienda

Si desea utilizar la tienda como un ejemplo que entregará a otros, o publicarla mediante el programa de utilidad de publicación de la Consola de administración, o si desea desplegarla en otro servidor o plataforma, puede empaquetarla con el formato de archivador de tienda.

Normalmente, un archivador de tienda está compuesto de los archivos siguientes:

- **Elementos Web:** los archivos utilizados para crear las páginas de tienda como, por ejemplo, archivos HTML, archivos JSP, imágenes, gráficos y archivos de inclusión.
- **Paquete de recursos de propiedades:** Contiene el texto para las páginas de la tienda. Si la tienda da soporte a más de un idioma, el archivador de tienda contendrá varios paquetes de recursos, uno por idioma soportado, además de un paquete de recursos por omisión (que no incluye un entorno nacional). Por ejemplo, `AddressText_en_US.properties` y `AddressText.properties`.
- **Elementos de datos de tienda:** Los datos que se deben cargar en la base de datos. Los elementos de datos de tienda incluyen datos como las campañas, las entradas de catálogo, las monedas, la información de despacho de pedidos, de fijación de precios, de envío, de la tienda y de los impuestos. Para obtener una lista más detallada de los elementos de datos de tienda, consulte la Parte 6, “Desarrollo de los datos de la tienda”, en la página 117.

Los elementos de datos de tienda de los archivadores de tienda de ejemplo proporcionados con WebSphere Commerce forman parte de un archivo XML completo válido para Loader Package. Los archivos XML de los elementos de datos de tienda están pensados para ser transportables y no deben contener claves primarias generadas que sean específicas de una instancia determinada de la base de datos. En lugar de ello, utilizan alias internos que ID Resolver resuelve en el momento en que se publica la tienda. Si se utilizan estos convenios se pueden portar los archivadores de tienda de ejemplo. Para obtener más información, consulte la Parte 9, “Empaquetado de la tienda”, en la página 337.

Para obtener más información sobre Loader package, consulte el Capítulo 37, “Visión general de la carga de datos de tienda”, en la página 361.

Nota: Los elementos de datos de la tienda también incluyen información del contrato, que proporciona la información necesaria para crear un contrato. La información del contrato no se carga mediante Loader package; éste proporciona entrada para un mandato que crea los contratos.

- **Elementos de pago:** Información de configuración para WebSphere Commerce Payments. La información de pago no se carga mediante Loader package; éste proporciona entrada para un mandato que configura WebSphere Commerce Payments.
- **Descriptorios:** Archivos XML que describen el archivador de tienda e información sobre cómo se debe publicar. Estos archivos son `store-refs.xml`, `ibm-wc-load.xml`, `unpack.xml` y `ForeignKeys.dtd`.

Los archivos de los archivadores de tienda de ejemplo están agrupados en la estructura siguiente:

- *Directorio de tienda*

- Archivos JSP, HTML: se agrupan en subdirectorios por áreas de función. Por ejemplo, ShoppingArea, AuctionArea, CustomerServiceArea. Cada una de estas áreas se agrupa a su vez en secciones. Por ejemplo, el área ShoppingArea está subdividida en las secciones CatalogSection, CheckoutSection, DiscountSection y ShopcartSection. Cada una de estas secciones también se puede dividir en secciones, si es necesario.
- Imágenes: Las imágenes de los archivadores de tienda de ejemplo se agrupan según el entorno nacional.
- SAR-INF
 - Contiene información específica para publicar este archivador de tienda, incluidos los archivos siguientes:
 - store-refs.xml: define los parámetros de publicación que se utilizarán con este archivador de tienda.
 - archivos de propiedades: el texto que se utiliza para los parámetros de publicación de este archivador de tienda. Los archivadores de tienda de ejemplo también incluyen las versiones específicas del entorno nacional de este archivo de propiedades.
 - unpack.xml: determina qué elementos del archivador de tienda se desempaquetarán, el modo en que se desempaquetarán y dónde se desempaquetarán.

Nota: Para obtener más información acerca de los archivos store-refs.xml y unpack.xml, consulte el Capítulo 36, “Publicación de una tienda completa”, en la página 347.

- WEB-INF
 - Archivos de propiedades: Los archivos de propiedades de las tiendas están situados en la estructura de directorio siguiente en WEB-INF:
 - Classes
 - *Directorio de tienda*
 - Elementos de datos de la tienda: Los elementos de datos de la tienda, con el formato XML, se encuentran en la estructura de directorio siguiente en WEB-INF:
 - Tiendas

Nota: El archivador de tienda compuesto incluye un directorio de modelos de negocio adicional.

- *Directorio de tienda*
 - datos: Si la tienda contiene varios idiomas, los archivos XML específicos del entorno nacional se agrupan según el entorno nacional. El directorio de datos también contiene los siguientes archivos necesarios para publicar el archivador de tienda:
 - ForeignKeys.dtd: almacena los valores del parámetro de publicación. Estos parámetros son parejas de nombre-valor a los que hacen referencia los XML de elementos de datos de tienda.
 - ibm-wc-load.xml: controla la carga de datos.
 - store-data-assets.xml: un archivo XML que incluye todos los datos de la tienda que se han de cargar.

Notas:

1. Para obtener más información sobre los archivos ForeignKeys.dtd e ibm-wc-load.xml, consulte el Capítulo 36, “Publicación de una tienda completa”, en la página 347.

2. La información sobre la configuración de pagos, con formato de archivo XML, también se incluye en el directorio de datos.
- Archivos para cambiar el flujo de la tienda: Si el flujo de la tienda se puede modificar utilizando las herramientas de la tienda de WebSphere Commerce Accelerator, los archivos necesarios se encuentran en la estructura de directorio siguiente:
 - xml
 - tools
 - stores
 - *Directorio de tienda*
 - devtools
 - flow

Creación de un archivador de tienda

La estructura de empaquetado del archivador de tienda es flexible. Las instrucciones que se proporcionan aquí reflejan la estructura de los ejemplos proporcionados con WebSphere Commerce, no obstante, puede alterar esta estructura para que se ajuste a sus necesidades. El archivador de tienda debe incluir un directorio SAR-INF que contiene un archivo `store-refs.xml` y un archivo `unpack.xml`. Las vías de acceso definidas en estos dos archivos deben ser coherentes con la estructura del archivador de tienda.

Para empaquetar la tienda como un archivador de tienda, efectúe lo siguiente:

1. Revise la estructura y el contenido de los archivadores de tienda de ejemplo que se proporcionan con WebSphere Commerce.
 - Los archivos de archivador de tienda están ubicados en el directorio siguiente:
 - *dir_instalación_WC* /samplestores/*modelo_negocio*
 - Para ver el archivador de tienda, utilice un programa de descompresión.
2. Cree un directorio temporal en WebSphere Commerce Server para la tienda. Por ejemplo, *mitienda*.
3. Cree los subdirectorios siguientes:
 - *Directorio de tienda* (el nombre de la tienda)
 - SAR-INF
 - WEB-INF
4. En el *Directorio de tienda*, efectúe lo siguiente:
 - Cree los subdirectorios para utilizar los archivos JSP por áreas de funciones. Consulte los archivadores de tienda de ejemplo para obtener un ejemplo. Copie los archivos JSP y los archivos HTML necesarios en estos subdirectorios.
 - Cree un subdirectorio para los archivos de imágenes. Si la tienda da soporte a varios idiomas, cree subdirectorios para la información específica del idioma utilizando los nombres del entorno nacional. Por ejemplo, *es_ES*. Copie los archivos de imágenes en estos subdirectorios.
5. En el directorio SAR-INF, efectúe lo siguiente:
 - a. Cree un archivo `store-refs.xml` para el archivador de tienda. Utilizando como ejemplo un archivo `store-refs.xml` existente de un archivador de tienda de ejemplo, cree un archivo `store-refs.xml` para la tienda. Para obtener más información acerca de las especificaciones XML, consulte el descriptor, `store-refs.dtd`, en el directorio siguiente:

- *dir_instalación_WC/xml/sar*

Nota: Para obtener más información sobre el archivo *store-refs.xml*, consulte el Capítulo 36, “Publicación de una tienda completa”, en la página 347.

- (Opcional) Si desea que el archivador de tienda tenga parámetros que pueda seleccionar el usuario durante el proceso de publicación desde la Consola de administración, cree un archivo de propiedades que describa estos parámetros. Guarde este archivo en un subdirectorio denominado *properties*.
- Utilizando como ejemplo un archivo *unpack.xml* existente de un archivador de tienda de ejemplo, cree un archivo *unpack.xml* para la tienda. El archivo *unpack.xml* determina el modo en que se desempaquetará el archivador de tienda. Para obtener más información sobre las especificaciones XML, consulte el descriptor *unpack.dtd* en el directorio siguiente:
 - *dir_instalación_WC/xml/sar*

Nota: Para obtener más información sobre el archivo *unpack.xml*, consulte el Capítulo 36, “Publicación de una tienda completa”, en la página 347.

- En el directorio *WEB-INF*, efectúe lo siguiente:
 - Cree la estructura de subdirectorios siguiente para los archivos de propiedades de la tienda:
 - *classes*
 - *Directorio de tienda*
 - Copie los archivos de propiedades en el *Directorio de tienda*.
 - Cree la estructura de subdirectorios siguiente para los elementos de datos de la tienda:
 - *stores*
 - *Directorio de tienda*
 - *data*: Si la tienda da soporte a varios idiomas, cree subdirectorios para la información específica del idioma utilizando los nombres del entorno nacional. Por ejemplo, *es_ES*.
 - Copie los elementos de datos en el directorio de datos y los subdirectorios correspondientes.
 - Utilizando como ejemplo un archivo *ibm-wc-load.xml* existente de un archivador de tienda de ejemplo, cree un archivo *ibm-wc-load.xml* para la tienda. El archivo *ibm-wc-load.xml* determina el modo en que se cargarán los datos de la tienda. Para obtener más información acerca de las especificaciones XML, consulte el descriptor, *ibm-wc-load.dtd*, en el directorio siguiente:
 - *dir_instalación_WC/xml/sar*
 - Nota:** Para obtener más información sobre el archivo *ibm-wc-load.xml*, consulte el Capítulo 36, “Publicación de una tienda completa”, en la página 347.
 - (Opcional) Si desea que el archivador de tienda tenga parámetros que pueda seleccionar el usuario durante el proceso de publicación desde la Consola de administración, debe crear un archivo *ForeignKeys.dtd* para almacenar los valores de estos parámetros. Utilizando como ejemplo un archivo *ForeignKeys.dtd* existente de un archivador de tienda de ejemplo, cree un archivo *ForeignKeys.dtd* para la tienda.

Nota: Para obtener más información acerca del archivo `ForeignKeys.dtd`, consulte el Capítulo 36, “Publicación de una tienda completa”, en la página 347.

7. Cree un archivo ZIP compuesto por el *Directorio de tienda*, el directorio SAR—INF y el directorio WEB-INF. A este archivo ZIP le dará el nombre *nombre_archivador_tienda.sar*.
8. Si desea publicar el archivador de tienda utilizando la Consola de administración, consulte el apartado “Disponibilidad del archivador de tienda en la Consola de administración” en la página 359.

Creación de un archivador de tienda de ejemplo

Después de empaquetar la tienda como un archivador de tienda, puede optar por utilizarla como una tienda de ejemplo en la Consola de administración. Para utilizar el archivador de tienda como un archivador de tienda de ejemplo, efectúe lo siguiente:

1. Guarde el archivador de tienda en el directorio siguiente:
 - *dir_instalación_WC/samplestores*
2. (Opcional) Cree páginas de vista previa. Para poder visualizar las vistas previas de las páginas de la tienda en la Consola de administración, deberá crear las páginas de vista previa. Efectúe lo siguiente:
 - a. (Opcional) En el programa de utilidad de publicación de la Consola de administración, seleccione un archivador de tienda para publicarlo y luego pulse **Vista previa**. Las páginas que desea visualizar se denominan páginas de vista previa. Estas páginas son archivos HTML que presentan un flujo de compra de ejemplo definido previamente y actúan como una vista previa de la tienda de ejemplo.
 - b. Determine el flujo de compra que desea mostrar en las páginas de vista previa.
 - c. (Opcional) Cree algunos datos de ejemplo en una tienda publicada. Por ejemplo, añada artículos al carro de la compra y cree algunas direcciones de envío y direcciones de facturación. Creará las páginas de vista previa desde esta tienda y los datos harán que las páginas resulten más reales.
 - d. Con Internet Explorer, vaya hasta la tienda. Guarde el HTML de cada página, seleccionando Archivo, Guardar como. También deberá guardar la hoja de estilo (.css) y las imágenes. Guarde los archivos en los directorios siguientes:
 - *stylesheet.css*
 - *dir_instalación_WC/wc.ear/SiteAdministration.war/tools/devtools/preview/locale/modelo_negocio/dir_tienda/*
 - HTML
 - *dir_instalación_WC/wc.ear/SiteAdministration.war/tools/devtools/preview/locale/modelo_negocio/dir_tienda*
 - imágenes independientes del entorno nacional
 - *dir_instalación_WC/wc.ear/SiteAdministration.war/tools/devtools/preview/images/modelo_negocio/dir_tienda*
 - imágenes dependientes del entorno nacional

Parte 10. Publicación de la tienda

Para crear una tienda que funcione, los elementos Web del escaparate deben publicarse en WebSphere Commerce Server y los datos de la tienda deben publicarse en la base de datos de WebSphere Commerce.

Los capítulos de esta sección describen las opciones de publicación que proporciona WebSphere Commerce:

- Capítulo 36, "Publicación de una tienda completa", en la página 347 - Este capítulo describe la publicación de una tienda entera (elementos de datos de tienda y de escaparate), si la tienda está en forma de archivador de tienda, utilizando la Consola de administración o el programa de utilidad de publicación de la línea de mandatos.
- Capítulo 37, "Visión general de la carga de datos de tienda", en la página 361 - Este capítulo describe cómo se publican los elementos de datos de tienda en la base de datos utilizando Loader package.
- Capítulo 38, "Carga de grupos de elementos de base de datos de WebSphere Commerce", en la página 405 - Este capítulo describe cómo se publican los grupos de elementos de datos de tienda o todos los datos de tienda en la base de datos utilizando Loader package.
- Capítulo 39, "Publicación de cuentas de negocio y contratos", en la página 419 - Este capítulo describe la publicación de elementos de cuenta, contrato y conjunto de productos.
- Capítulo 40, "Publicación de los elementos de escaparate y de los archivos de configuración de tiendas", en la página 423 - Este capítulo describe la publicación de los elementos de escaparate y los archivos de configuración de la tienda.

Capítulo 36. Publicación de una tienda completa

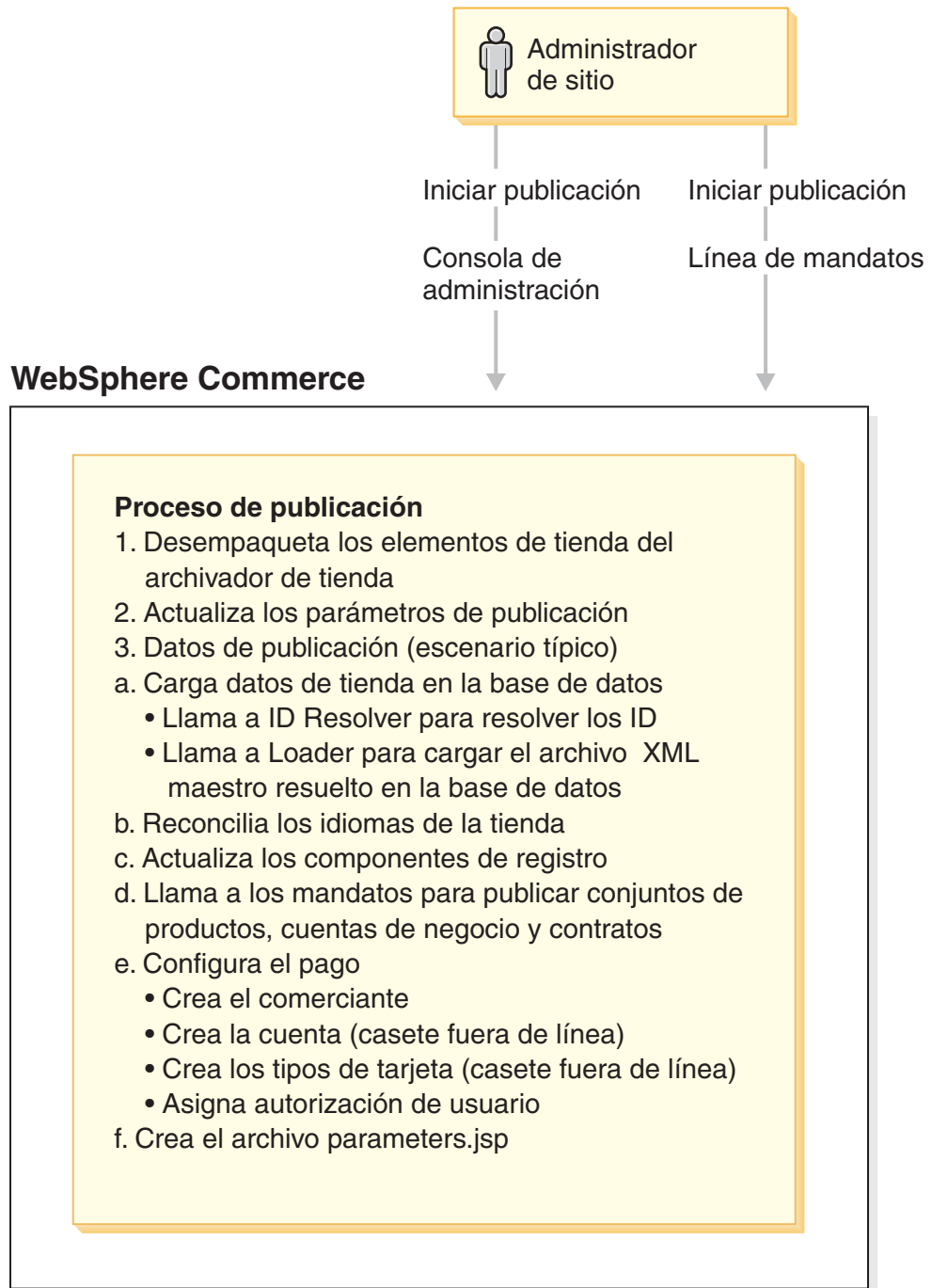
Para crear una tienda que funcione, los elementos Web del escaparate deben publicarse en WebSphere Commerce Server y los datos de la tienda deben publicarse en la base de datos de WebSphere Commerce. Este capítulo describe la publicación de una tienda entera (elementos de escaparate y de datos de tienda), si la tienda está en forma de archivador de tienda, utilizando el programa de utilidad de publicación en la Consola de administración o con la línea de mandatos.

Nota: Si prefiere no empaquetar la tienda como un archivador de tienda, puede publicar los elementos individualmente. Para obtener más información, consulte el Capítulo 37, “Visión general de la carga de datos de tienda”, en la página 361, el Capítulo 38, “Carga de grupos de elementos de base de datos de WebSphere Commerce”, en la página 405, el Capítulo 39, “Publicación de cuentas de negocio y contratos”, en la página 419 y el Capítulo 40, “Publicación de los elementos de escaparate y de los archivos de configuración de tiendas”, en la página 423.

Descripción de la publicación en WebSphere Commerce

La opción de publicación que está disponible en los Servicios de tienda o en la línea de mandatos le permite publicar una tienda completa (elementos de escaparate y de datos de tienda) de forma inmediata. Para utilizar esta opción, los elementos de tienda deben estar empaquetados en forma de archivador de tienda. Para obtener más información sobre cómo empaquetar la tienda como un archivador de tienda, consulte la Parte 9, “Empaquetado de la tienda”, en la página 337.

El diagrama siguiente describe los pasos del proceso de publicación.



Iniciar la publicación

Para publicar una tienda, debe tener autorización de Administrador de sitio. Los Administradores de sitio pueden iniciar el proceso de publicación utilizando uno de los métodos siguientes:

- Consola de administración
- Línea de mandatos

Ambos métodos de publicación requieren que se especifique el archivador de tienda que se desea publicar.

Nota: Para poder publicar un archivador de tienda utilizando la Consola de administración, debe estar en la ubicación correcta o estar registrado. Para obtener más información, consulte el apartado “Disponibilidad del archivador de tienda en la Consola de administración” en la página 359. A continuación, puede cambiar los valores de los parámetros seleccionados, si están disponibles, incluido el identificador de la tienda (el nombre que identifica de forma exclusiva la tienda), el directorio de tienda (la ubicación exclusiva en la que se publicarán los archivos JSP y las imágenes) y la organización (la organización en la que se publicará el archivador de tienda).

Parámetros de publicación en la Consola de administración

Los parámetros de publicación del programa de utilidad de publicación de la Consola de administración se definen mediante el archivo `store-refs.xml` de cada archivador de tienda.

Observe el ejemplo siguiente de un archivo `store-refs.xml` del archivo `ConsumerDirectStore.sar`.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE store-refs SYSTEM "store-refs.dtd">
<store-refs
  target-dtd="WEB-INF/stores/FashionFlow/data/ForeignKeys.dtd"
  deploy-descriptor="WEB-INF/stores/FashionFlow/data/ibm-wc-load.xml"
  resource-bundle="/SAR-INF/properties/publishNLS">
  ref id="storeDir" entity="STORE_DIR" >
    <input type="text"/>
  </ref>
  <ref id="storeIdent" entity="STORE_IDENTIFIER" >
    input type="text"/>
  </ref>
  <ref id = "parentOrg" entity="ORGANIZATION_DN">
    <input type="member" />
  </ref>
</store-refs>
```

Este archivo define tres parámetros de publicación para el archivo `ConsumerDirectStore.sar`:

- `<ref id="storeDir" entity="STORE_DIR" >`
 `<input type="text"/>`

Esta entidad crea el directorio de tienda de los parámetros de publicación.

- `<ref id="storeIdent" entity="STORE_IDENTIFIER" >`
 `< input type="text"/>`

Esta entidad crea el identificador de tienda de los parámetros de publicación.
store identifier.

- `<ref id = "parentOrg" entity="ORGANIZATION_DN">`
 `<input type="member" />`

Esta entidad crea la organización de los parámetros de publicación.

donde

- `ref id` se utiliza como la clave del archivo de propiedades especificado mediante el atributo `store-refs resource-bundle`. Se utiliza para obtener el nombre del parámetro traducible y la descripción que se visualiza en la página de parámetros de publicación.
- `entity` es el nombre de ENTITY en el archivo `target-dtd` que edita este parámetro.
- `input type` controla cómo se visualiza el parámetro en la pantalla. Si el tipo de entrada es texto, el parámetro se visualiza en un campo editable. Si el tipo de

entrada es miembro, todas las organizaciones existentes aparecen en la lista desplegable. Los parámetros que son de sólo lectura (read-only) no se pueden editar.

Los valores que especifique el usuario para estos parámetros se almacenan en el archivo que identifica el archivo target-dtd. El archivo target dtd se define con el código siguiente:

```
target-dtd="WEB-INF/stores/ConsumerDirect/data/ForeignKeys.dtd
```

Este archivo también forma parte de un archivador de tienda y se desempaqueta con los elementos de datos de tienda. El valor de entidad correspondiente a cada parámetro se actualiza en el archivo desempquetado. La DTD que contiene el archivador de tienda no se actualiza. Los valores de los parámetros se almacenan en este archivo (en este caso ForeignKeys.dtd) hasta que se crea una instancia de la publicación.

Finalmente, si se puede publicar una tienda en varios idiomas, como ocurre con las tiendas de ejemplo, los parámetros de publicación y sus descripciones se encuentran en archivos específicos del entorno nacional. La etiqueta de campo y la descripción de cada parámetro de publicación se encuentran en el archivo de propiedades definido en el atributo resource-bundle de store-refs.xml. Durante la publicación, se busca el entorno nacional específico del idioma que utiliza la Consola de administración. El archivo stores-ref.xml también define estos archivos:

- resource-bundle="/SAR-INF/properties/publishNLS"

Nota: Los parámetros de publicación solamente están disponibles mediante la Consola de administración. Si publica un archivador de tienda mediante la línea de mandatos, no puede especificar valores de parámetros. Se utilizarán los valores por omisión contenidos en el archivador de tienda.

El descriptor de despliegue especifica la ubicación del archivo (ibm-wc-load.xml) que controla la parte de datos de publicación correspondiente al proceso de publicación. Por ejemplo, deploy-descriptor="WEB-INF/stores/FashionFlow/data/ibm-wc-load.xml"

Después de seleccionar los parámetros, pulse **Finalizar** para iniciar la publicación. Después de haber iniciado el proceso de publicación utilizando la Consola de administración o la línea de mandatos, no tiene que hacer nada más. Todos los demás pasos listados en el diagrama anterior y en este capítulo los realiza el sistema WebSphere Commerce.

Para obtener información más detallada sobre cómo publicar un archivador de tienda utilizando la Consola de administración o la línea de mandatos, consulte la ayuda en línea de Ayuda en línea a la producción de WebSphere Commerce.

Desempaquetado de los elementos del archivador de tienda

Después de pulsar **Finalizar** en el asistente de publicación de la Consola de administración, o de ejecutar la publicación desde la línea de mandatos, WebSphere Commerce desempaqueta los elementos del archivador de tienda en WebSphere Commerce Server. El archivo unpack.xml, situado en el directorio SAR-INF del archivador de tienda, controla el desempaquetado de los elementos.

El archivo unpack.xml del ejemplo siguiente procede del archivo ConsumerDirect.sar:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ibm-wc-unpack SYSTEM "unpack.dtd">
<ibm-wc-unpack>
  <unpack>
    <include file="*" />
    <exclude file="SAR-INF/*" />
    <exclude file="*.zip" />
    <exclude file="*.war" />
    <exclude file="FashionFlow/devtools/flow/ui/*.properties" />
    <exclude file="xml/*" />
    <rename-store-dir target-name="FashionFlow">
      <store-dir path="WEB-INF/stores/FashionFlow" />
      <store-dir path="FashionFlow" />
      <store-dir path="WEB-INF/classes/FashionFlow" />
      <store-dir path="WEB-INF/xml/tools/stores/FashionFlow" />
    </rename-store-dir>
  </unpack>
  <unpack dest="{wc:ToolsStoresPropertiesPath}">
    <include file="FashionFlow/devtools/flow/ui/*.properties" />
    <rename-store-dir target-name="FashionFlow">
      <store-dir path="FashionFlow" />
    </rename-store-dir>
  </unpack>
  <unpack dest="{wc:instanceDir}">
    <include file="xml/member/MemberRegistrationAttributes.xml" />
  </unpack>
</ibm-wc-unpack>

```

El archivo `unpack.xml` determina qué archivos se han de desempaquetar (utilizando los elementos `include` y `exclude`), en los que se desempaquetarán los archivos (utilizando la entidad `unpack dest`), y también cambia el nombre de los directorios (utilizando la entidad `rename-store-dir`).

Por omisión, `unpack` desempaqueta todos los archivadores de tienda en el que está situado. No obstante, `unpack` también puede desempaquetar determinados archivos del archivador de tienda, si se especifica. Asimismo, por omisión, `unpack` desempaqueta los archivos en la vía de acceso obtenida mediante la combinación de las vías de acceso `StoreDocRoot` y `StoreWebPath` del elemento `DevTools` del XML de la instancia. Esta vía de acceso apunta al directorio raíz de documentos del módulo `Web Stores`. No obstante, si se especifica, como en el ejemplo anterior, `<unpack dest="{wc:ToolsStoresPropertiesPath}">`, `unpack` desempaquetará los archivos en otra ubicación. Tenga en cuenta que `unpack` acepta variables. En `"{wc:ToolsStoresPropertiesPath}"`, la variable `ToolsStoresPropertiesPath` es un atributo del elemento `devtools` del `xml` de la *instancia*.

Actualización de los parámetros de publicación

Una vez desempaquetados los elementos del archivador de tienda, `WebSphere Commerce` actualiza el archivo DTD (en las tiendas de ejemplo, `ForeignKeys.dtd`) con los valores de los parámetros de publicación que se han creado o seleccionado en el asistente de publicación. Por ejemplo, si el archivo original contenía `<!ENTITY STORE_IDENTIFIER "FashionFlow">`, el archivo actualizado contendrá `<!ENTITY STORE_IDENTIFIER "MyFashion">`.

Nota: Los parámetros de publicación solamente están disponibles mediante la Consola de administración. Si publica un archivador de tienda mediante la línea de mandatos, no puede seleccionar parámetros de publicación

Datos de publicación

Una vez desempaquetados los archivos y actualizados los parámetros de publicación, se crea un trabajo planificado para el proceso de publicación. El número de trabajo planificado para publicar el archivador de tienda se visualiza en el programa de utilidad de publicación de la Consola de administración.

Cuando el planificador ejecuta el trabajo de publicación, generalmente WebSphere Commerce realiza las acciones siguientes:

- Carga en la base de datos los datos de tienda del XML del archivador de tienda
- Reconcilia los datos de tienda
- Actualiza los componentes de registro
- Llama a los mandatos para publicar cuentas de negocio y contratos
- Configura el pago
- Crea el archivo parameters.jsp

El trabajo de publicación se controla mediante el archivo `ibm-wc-load.xml` que contiene cada archivador de tienda. Este archivo se especifica mediante el atributo `deploy-descriptor` del archivo `stores-refs.xml`.

ibm-wc-load.xml

El archivo `ibm-wc-load.xml` determina las tareas que se completarán en el trabajo de publicación y la secuencia de estas tareas. El siguiente es un ejemplo de un archivo `ibm-wc-load.xml`:

```
<data-deploy base-dir="." default-target="all">
  <asset id="master" location="store-data-assets.xml"/>
  <asset id="resolved.master" location="store-data-assets.resolved.xml"/>
  <asset id="foreignKeys" location="ForeignKeys.dtd" type="dtd"/>
  <asset id="pmconfigfile" location="paymentinfo.xml"/>
  <deploy-task-cmd name="configPM" class="com.ibm.commerce.tools.devtools.
publish.tasks.payment.ConfigurePaymentTaskCmd"/>
  <target id="all">
    <task name="idresolve">
      param name="infile" value="{asset:master}" />
      param name="outfile" value="{asset:resolved.master}" />
    </task>
    <task name="massload">
      param name="infile" value="{asset:resolved.master}" />
      param name="maxerror" value="1" />
      param name="noprimary" value="error" />
    </task>
    <task name="configPM">
      param name="paymentConfigFilename" value="{asset:pmconfigfile}" />
      param name="storeIdentifier" value="
${asset:foreignKeys#STORE_IDENTIFIER}" />
      param name="organizationDN" value="
${asset:foreignKeys#ORGANIZATION_DN}" />
    </task>
  </target>
</data-deploy>
```

fonde

- `base-dir` es el directorio de la información que se va a publicar. `."` indica que la información se encuentra en el mismo directorio que el archivo `ibm-wc-load.xml`.
- `default-target` es el ID de un destino que se va a ejecutar. Solamente se ejecutará un destino durante la publicación.

- `asset id` es el ID asignado a los elementos que se van a publicar. Se asignan ID a los elementos ya que puede ejecutarse más de una acción en los mismos durante el proceso de publicación.
- `location` es la ubicación del elemento al que se ha asignado el ID, relativo al directorio base (`base-dir`).
- `deploy-task-cmd name` es el nombre abreviado asignado a los mandatos que se utilizan en el proceso de publicación.
- `deploy-task-cmd class` es el nombre completo de los mandatos de tarea que se utilizan en el proceso de publicación.
- `target id` es un nombre asignado a una secuencia de tareas que se ejecutan como un grupo. Se pueden definir varios destinos pero solamente se ejecutará durante la publicación aquel al que hace referencia el destino por omisión.
- `task name` es el nombre de la tarea que se va a completar. Tenga en cuenta que en este ejemplo se utiliza el nombre abreviado de la tarea. Puede añadir tareas nuevas al archivo `ibm-wc-load.xml`, pero cualquier tarea nueva debe ampliar el mandato siguiente:
`com.ibm.commerce.tools.devtools.publish.tasks.DeployTaskCmd`.
- `param name` es el nombre del parámetro de la tarea. Los parámetros de entrada se pasan a la tarea como series de parejas de nombre-valor.
- `value` es el valor del parámetro. Tenga en cuenta que pueden ser variables. Estas variables se resolverán cuando se ejecute la tarea.

Carga en la base de datos los datos de tienda de los archivos XML del archivador de tienda

Para cargar en la base de datos los datos de tienda de los archivos XML del archivador de tienda, WebSphere Commerce realiza lo siguiente:

Llamada a ID Resolver para resolver los ID: ID Resolver, que es un programa de utilidad de Loader package, genera identificadores exclusivos para elementos XML de los archivos XML del archivador de tienda. Por ejemplo, ID Resolver sustituye el alias `@` utilizado en los archivos XML de la tienda de ejemplo por un valor exclusivo. Para obtener un ejemplo de la resolución de los alias internos utilizados en las tiendas de ejemplo, consulte el Apéndice B, “Creación de los datos”, en la página 465.

Nota: ID Resolver también puede resolver identificadores para tiendas ya publicadas, cuando las vuelva a publicar. Por ejemplo, si ha publicado el archivador de tienda una vez y tiene que volver a publicarlo, todo o partes del mismo, ID Resolver recupera los identificadores exclusivos de la base de datos y los utiliza cuando vuelve a publicar.

Para obtener más información sobre ID Resolver y los demás componentes de Loader package, consulte el Capítulo 37, “Visión general de la carga de datos de tienda”, en la página 361.

Cuando la publicación llama ID Resolver, debe especificar qué método de ID Resolver se debe utilizar. ID Resolver tiene varios métodos, que se pueden utilizar para procesar la entrada de ID Resolver; específicamente consisten en tratar los datos como si existieran identificadores en los datos originales (método de actualización) o tratar dichos datos como si no existieran identificadores en los datos originales (método de carga). El método combinado se utiliza cuando algunos identificadores existen y otros no. Puede especificar el método que utilizará la configuración en el archivo de configuración de WebSphere Commerce, `nombre_instancia.xml`. Por omisión, el proceso de publicación utiliza el método combinado. Para obtener más información sobre los métodos de ID Resolver, consulte el apartado “Mandato ID Resolve” en la página 365.

La publicación también debe especificar un archivo de personalización para utilizarlo con ID Resolver. Los archivos de personalización por omisión son los siguientes: DBConnectionCustomizer u OracleConnectionCustomizer.

Oracle

El archivo de personalización OracleConnectionCustomizer se encuentra en el siguiente directorio:

- *dir_instalación_WAS/installedApps/nombre_célula/nombre_instancia_WC.ear/properties*

DB2

El archivo DBConnectionCustomizer se encuentra en el siguiente archivo ZIP:

- *dir_instalación_WAS/installedApps/nombre_célula/nombre_instancia_WC.ear/properties*
- **400** *dir_usuario_WAS/WAS_nombre_instancia/installedApps/nombre_célula/nombre_instancia_WC.ear/lib/loader/IdResGen.zip*

Nota: Si desea especificar su propio archivo de personalización, debe modificar el valor del atributo siguiente en la sección DevTools del archivo *nombre_instancia.xml*:

- `IDResolverCustomizerFile="myIDResolverCustomizerFile"`

store-data-asset.xml: Cada archivador de tienda de ejemplo contiene un archivo *store-data-asset.xml*. El archivo *store-data-asset.xml* incluye contenedores para los archivos de elementos de datos del archivador de tienda que se incluirán durante la publicación.

La siguiente es una parte del archivo *store-data-asset.xml* para *ConsumerDirect.sar*, que ilustra los contenedores:

```
<?xml version="1.0"?>
<!DOCTYPE import SYSTEM "store-data-assets.dtd">

<import>
  &modelorg.xml;
  &modelorgrole.xml;
  &storeorg.xml;
  &storeorgrole.xml;
  &fulfillment.xml;
  &store.xml;
  &en_US_store.xml;
  &en_US_fulfillment.xml;
  &catalog.xml;
  &en_US_catalog.xml;
  &tax.xml;
```

Durante la publicación, todos los elementos de datos identificados con contenedores en el archivo *store-data-asset.xml* se consolidan en el archivo *store-data-asset.xml* y crean un archivo de datos más grande.

ID Resolver utiliza *store-data-asset.xml* y el archivo DTD correspondiente *store-data-asset.dtd* para resolver los ID. Una vez resueltos los ID, ID Resolver crea el archivo siguiente, el archivo *store-data-asset.resolved.xml*, que contiene los identificadores exclusivos. Si se produce un error durante el proceso de resolución de los ID, Loader package añade una entrada al archivo *messages.txt*.

Para obtener más información, consulte el apartado “Archivos de anotaciones cronológicas de la publicación” en la página 357.

Llamada a Loader package para cargar el archivo XML maestro resuelto en la base de datos: Loader package carga el archivo `store-data-asset.resolved.xml` resuelto en la base de datos. Si se produce un error durante el proceso de carga, Loader package añade una entrada al archivo `messages.txt`.

Para obtener más información sobre Loader package, consulte el Capítulo 37, “Visión general de la carga de datos de tienda”, en la página 361.

Cuando la publicación mediante la Consola de administración o la línea de mandatos llama a Loader package, debe especificar qué método de Loader se ha de utilizar. La Consola de administración puede utilizar los métodos siguientes de Loader:

- SQL import (Importación de SQL)
- Import (Importación)
- Load (Carga)

Nota: Por omisión, la Consola de administración utilizan el método de importación de SQL.

Puede especificar el método al que llamará la publicación mediante la Consola de administración o la línea de mandatos en el archivo de configuración de WebSphere Commerce, `nombre_instancia.xml`, utilizando el atributo `LoaderMode` del elemento `DevTools`.

- SQL import: Este método utiliza JDBC (Java Database Connectivity) para insertar y actualizar datos, proporcionando el método más flexible de operación pero también el más lento para importar grandes cantidades de datos a un pequeño número de tablas. Permite la actualización a nivel de columna. Se recomienda utilizar SQL import.

Nota: El método de importación de SQL es el método más seguro porque no corromperá la base de datos si los datos no son válidos. Para poder efectuar la carga utilizando la importación de SQL, los registros deben satisfacer las restricciones de esquema de base de datos. Los otros métodos de Loader son más rápidos porque los datos se cargan masivamente en la base de datos sin efectuar muchas comprobaciones. Por tanto, para utilizar los otros métodos debe estar seguro de que los datos sean correctos.

- Import: Este método utiliza funciones de importación nativas de DB2 y permite la actualización a nivel de casilla con una velocidad y flexibilidad medias. Este método no está disponible con Oracle.
- Load: Este método utiliza los recursos nativos de RDBMS (DB2 Load o SQLLoad) y es el método más rápido para cargar grandes cantidades de datos en un pequeño número de tablas.

Para obtener más información sobre los métodos del mandato load, consulte el apartado “Mandato Load” en la página 374.

La publicación mediante la Consola de administración y la línea de mandatos también debe especificar un archivo de personalización para utilizarlo con Loader. Si no especifica ningún archivo de personalización en el archivo de configuración de WebSphere Commerce, `nombre_instancia.xml`, el código de publicación utilizará el archivo de personalización por omisión: `MassLoadCustomizer`.

Nota: Si desea especificar su propio archivo de personalización, debe modificar el valor del atributo siguiente en la sección DevTools del archivo `nombre_instancia.xml`:

- `LoaderCustomizerFile="miArchivoPersonalizaciónLoader"`

Por omisión, el archivo de configuración de WebSphere Commerce, `nombre_instancia.xml`, no especifica un valor para este atributo.

Reconciliación de los idiomas de la tienda

Los archivadores de tienda de ejemplo contienen datos para todos los idiomas a los que WebSphere Commerce da soporte. Como resultado, cuando Loader package carga los datos de tienda, se carga toda la información para la tienda. No obstante, una tienda solamente da soporte a los idiomas a los que da soporte la instancia en la que reside la tienda. La tarea de reconciliación de idiomas de tienda garantiza que solamente los idiomas a los que se da soporte en la instancia estén habilitados en la tienda. Si desea añadir soporte de idiomas adicionales a la instancia, efectúelo utilizando el Gestor de configuración. Para obtener más información, consulte el tema "Gestor de configuración" de la Ayuda en línea a la producción de WebSphere Commerce.

Actualización de los componentes de registro

El proceso de publicación también actualiza los componentes de registro. La publicación actualiza todos los registros de WebSphere Commerce llamando a `com.ibm.commerce.scheduler.commands.RefreshRegistryCmd`. Para obtener más información sobre los registros, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

Llamada a los mandatos para publicar cuentas de negocio y contratos

Loader package no puede cargar algunos de los elementos de base de datos de tienda (contratos y cuentas de negocio), de modo que la publicación también llama a los mandatos correspondientes para publicar dichos elementos en WebSphere Commerce Server. Estos mandatos son los siguientes:

- `AccountImport` — Crea una cuenta de negocio a partir del archivo `businessaccount.xml` del archivador de tienda.
- `ContractImportApprovedVersion` — Importa un contrato del archivo `contract.xml` del archivador de tienda.
- `ProductSetPublish` — Sincroniza los datos del conjunto de productos en la base de datos con el catálogo, antes de crear las cuentas de negocio y los contratos. La publicación a través de los Servicios de tienda o la línea de mandatos llama al mandato `ProductSetPublish` que, a su vez, llama a los mandatos `AccountImport` y `ContractImportApprovedVersion`.

Para obtener más información sobre la publicación de cuentas de negocio y de contratos, consulte el Capítulo 39, "Publicación de cuentas de negocio y contratos", en la página 419.

Configuración del pago

El proceso de publicación incluye también un paso para configurar el pago. WebSphere Commerce da soporte a WebSphere Commerce Payments. Si piensa utilizar WebSphere Commerce Payments como método para procesar pagos, debe crear un archivo XML de pagos como se describe en el Capítulo 21, "Herramientas de Payments", en la página 227. Si se incluye un archivo XML de pago en el archivador de tienda que se está publicando, WebSphere Commerce realizará la configuración de pago siguiente durante la publicación:

- Crea el comerciante.


- Crea la cuenta (sólo para casetes fuera de línea).
- Crea los tipos de tarjeta especificados en `paymentinfo.xml` (sólo para casetes fuera de línea).
- Asigna autorización de usuario.

Manejo de errores: Si se produce un error durante la fase de configuración de pago del proceso de publicación, puede ver el mensaje de error en las anotaciones cronológicas de publicación (consulte el apartado “Archivos de anotaciones cronológicas de la publicación”).

Creación del archivo `parameters.jsp`

El proceso de publicación crea el archivo `parameters.jsp`. Este archivo incluye el parámetro `storeId`. El archivo `index.jsp` de las tiendas de ejemplo utiliza este parámetro para iniciar la tienda.

`parameters.jsp` está ubicado en el directorio siguiente:




- `dir_instalación_WAS/installedApps/nombre_célula/WC_nombre_instancia.ear/Stores.war/dir_tienda/include`
-  `dir_usuario_WAS/nombre_instancia_WAS/installedApps/nombre_célula/nombre_instancia_WC.ear/Stores.war/dir_tienda/include`

Manejo de errores



Si se produce un error durante la fase de publicación de elementos del proceso de publicación, puede ver el mensaje de error en las anotaciones cronológicas de publicación (consulte el apartado “Archivos de anotaciones cronológicas de la publicación”) o mediante la página Resumen de la publicación de la Consola de administración.

Archivos de anotaciones cronológicas de la publicación


Los errores encontrados durante la fase de publicación de elementos del proceso de publicación se graban en los archivos de anotaciones cronológicas y de rastreo siguientes:

- `activity.log`: el archivo de anotaciones cronológicas de WebSphere Application Server. Todos los mensajes de error de WebSphere Commerce se graban en `activity.log`. Cuando falle la publicación, compruebe primero esta anotación cronológica. `activity.log` está situado en el directorio siguiente:
 - `dir_instalación_WAS/logs`
 -  `dir_usuario_WAS/logs`
 -  `dir_instalación_WCStudio/nombre_espacio_trabajo/.metadata/plugins/com.ibm.etools.server.core/tmp0/logs`
- `SystemOut.log` y `SystemErr.log`: `SystemOut.log` está situado en el directorio siguiente. Contiene cualquier información que se ha grabado en la salida estándar y en los errores estándar producidos durante el proceso de publicación. `SystemOut.log` y `SystemErr.log` están situados en el directorio siguiente:
 - `dir_instalación_WAS/logs/nombre_instancia`
 -  `dir_usuario_WAS/nombre_instancia_WAS/logs/nombre_instancia_WC`





Nota: `SystemOut.log` y `SystemErr.log` no están disponibles en el entorno WebSphere Commerce Studio. La información que se ha grabado en la salida estándar y en los errores estándar se muestra en la consola de WebSphere Studio y se captura en la ubicación siguiente:

- workspaceDir\.metadata\.plugins\com.ibm.etools.server.core\tmp0\logs\server1\trace.log
- messages.txt: Contiene mensajes de error de la parte de Loader package del proceso de publicación. Cuando falle la publicación, compruebe primero esta anotación cronológica. Los números de línea o columna mencionados en estos mensajes de error hacen referencia al archivo store-data-asset.resolved.xml . El archivo messages.txt está situado en el directorio siguiente:
 - dir_instalación_WC/instances/nombre_instancia/logs
 -  dir_usuario_WC/instances/nombre_instancia/logs
 -  dir_instalación_WCStudio/Commerce/logs


Los números de línea o columna mencionados en estos mensajes de error hacen referencia al archivo store-data-asset.resolved.xml.

- trace.txt: Contiene información de rastreo para las partes de Loader package e ID Resolver del proceso de publicación. Este archivo contiene también mensajes acerca de la parte correspondiente a IDResolver del proceso de publicación. Por omisión, trace.txt está desactivado. El archivo trace.txt está situado en el directorio siguiente:
 - dir_instalación_WC/instances/nombre_instancia/logs
 -  dir_usuario_WC/instances/nombre_instancia/logs

Nota: Por omisión, se activa el rastreo y se genera como un archivo de anotaciones cronológicas circular.

-  dir_instalación_WCStudio/Commerce/instances/nombre_instancia/logs
- trace.log: Forma parte de los archivos de anotaciones cronológicas de rastreo de WebSphere Application Server. Contiene información de rastreo para el proceso de publicación, si se ha habilitado el componente de rastreo WC_DEVTOOLS. Para obtener información acerca de cómo habilitar las anotaciones cronológicas, consulte el capítulo sobre configuración, de la Guía de administración de WebSphere Commerce. trace.log está situado en el directorio siguiente:
 - dir_instalación_WAS/logs/nombre_instancia_WC
 -  dir_usuario_WASlogs/nombre_instancia_WC
 -  dir_espacio_trabajo\.metadata\.plugins\com.ibm.etools.server.core\tmp0\logs\server1
-  RESWCSID.txt: Contiene mensajes de la parte correspondiente a IDResolver del proceso de publicación. Los números de línea o columna mencionados en los mensajes hacen referencia al archivo de entrada, por ejemplo, al archivo store-data-assets.xml. RESWCSID.txt está situado en el directorio siguiente:
 - dir_usuario_WC/instances/nombre_instancia/logs

Para configurar los archivos de anotaciones cronológicas trace.txt y messages.txt (es decir, ajustar el nivel de anotación cronológica u otras opciones), edite el archivo siguiente:

- dir_instal_WC/instances/nombre_instancia/xml/loader/WCALoggerConfig.xml
-  dir_usuario_WC/instances/nombre_instancia/xml/loader/WCALoggerConfig.xml

-  `dir_instalación_WCStudio/Commerce/instances/nombre_instancia/xml/loader/WCALoggerConfig.xml`


Disponibilidad del archivador de tienda en la Consola de administración

Para poder publicar un archivador de tienda utilizando la Consola de administración, el archivador de tienda debe estar disponible en la Consola de administración mediante uno de los métodos siguientes:

- Registrando el archivador de tienda en el archivo SARRegistry.xml
- Copiando el archivador de tienda en el directorio de archivador de tienda aplicable

Registro del archivador de tienda en el archivo SARRegistry.xml

Para poder publicar el archivador de tienda y obtener una vista previa de la tienda desde la Consola de administración, el archivador de tienda debe estar registrado en el archivo SARRegistry.xml. El archivo SARRegistry.xml está situado en el directorio siguiente:

- `dir_instalación_WC/xml/tools/devtools`
-  `dir_usuario_WC/xml/tools/devtools`

Para registrar el archivador de tienda, incluya la vía de acceso del archivador de tienda y la vía de acceso a cualquier archivo de vista previa del archivo SARRegistry.xml. El ejemplo siguiente ilustra la entrada que registra el archivador de tienda de ejemplo Directo al consumidor en el archivo SARRegistry.xml.

```
<!-- Directo al consumidor -->
<SampleSAR fileName="ConsumerDirect.sar" relativePath="ConsumerDirect">
  <view name="ConsumerDirect" />
  <view name="default" />
  <html locale="de_DE" featureFile="" sampleSite="de_DE/B2C/FashionFlow/index.html"/>
  <html locale="en_US" featureFile="" sampleSite="en_US/B2C/FashionFlow/index.html"/>
  <html locale="es_ES" featureFile="" sampleSite="es_ES/B2C/FashionFlow/index.html"/>
  <html locale="fr_FR" featureFile="" sampleSite="fr_FR/B2C/FashionFlow/index.html"/>
  <html locale="it_IT" featureFile="" sampleSite="it_IT/B2C/FashionFlow/index.html"/>
  <html locale="ja_JP" featureFile="" sampleSite="ja_JP/B2C/FashionFlow/index.html"/>
  <html locale="ko_KR" featureFile="" sampleSite="ko_KR/B2C/FashionFlow/index.html"/>
  <html locale="pt_BR" featureFile="" sampleSite="pt_BR/B2C/FashionFlow/index.html"/>
  <html locale="zh_CN" featureFile="" sampleSite="zh_CN/B2C/FashionFlow/index.html"/>
  <html locale="zh_TW" featureFile="" sampleSite="zh_TW/B2C/FashionFlow/index.html"/>
</SampleSAR>
```

donde

- `fileName` es el nombre del archivador de tienda.

- `relativePath` es la vía de acceso del directorio relativa al atributo `sampleSarPath` del elemento `DevTools` en el archivo de configuración de WebSphere, *nombre_instancia.xml*.
- `view name` es el nombre de la vista que el archivador de tienda visualizará desde la interfaz de usuario de publicación. Tenga en cuenta que un archivador de tienda puede visualizarse en varias vistas. Por ejemplo, se visualizará en la vista Directo al consumidor y en la vista por omisión.
- `html locale` es el entorno nacional para los archivos de vista previa HTML.
- `featureFile` es el archivo de característica que ha quedado en desuso.

Copia del archivador de tienda en el directorio de archivador de tienda aplicable

Si desea publicar el archivador de tienda desde la Consola de administración y no piensa incluir las páginas de vista previa ni especificar la vista de publicación en la que se visualizará la tienda en la Consola de administración, simplemente puede copiar el archivador de tienda en el directorio siguiente:

- *dir_instalación_WC/instances/instance_name/sar*

El archivador de tienda se visualizará en la vista por omisión.

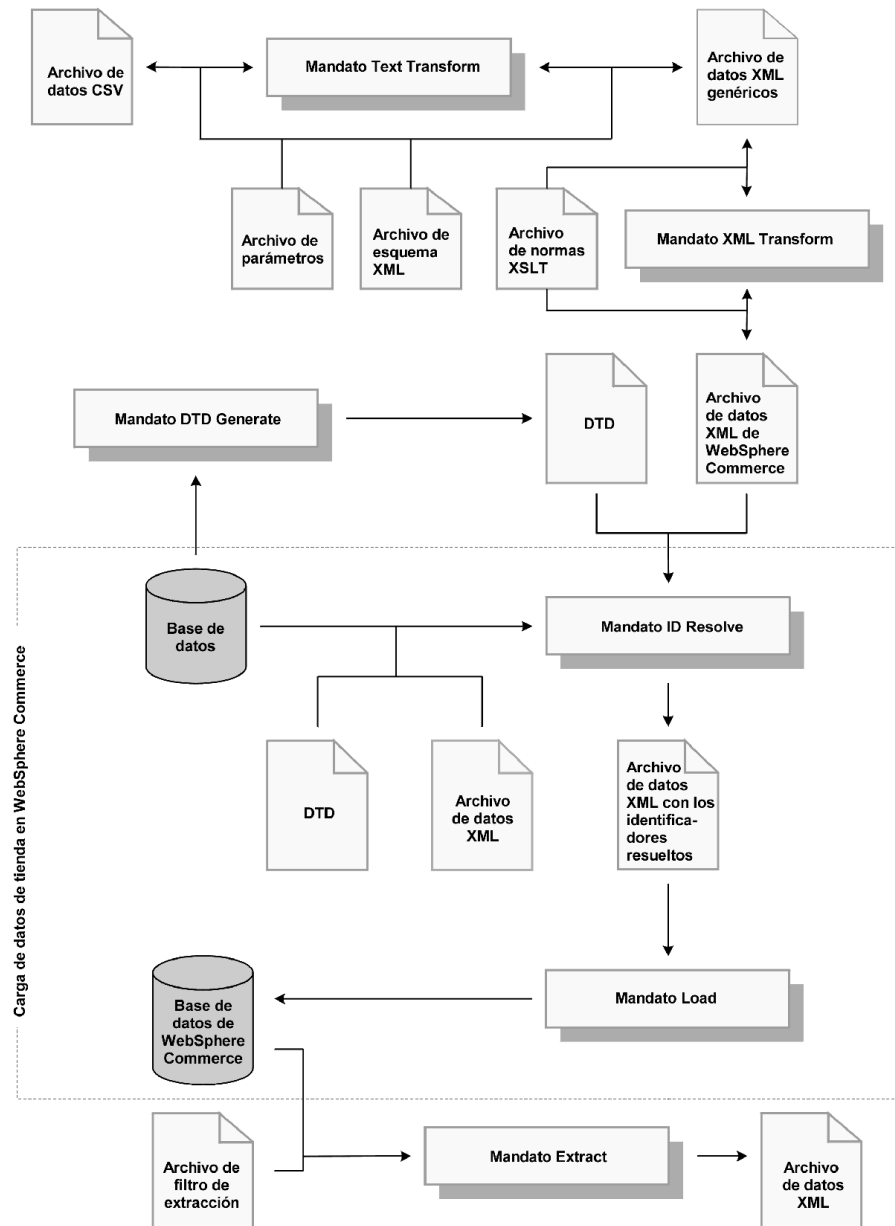
Capítulo 37. Visión general de la carga de datos de tienda

Después de crear los datos de la tienda, puede optar por empaquetarlos como un archivador de tienda y publicarlos utilizando la Consola de administración de WebSphere Commerce o puede cargarlos directamente en la base de datos de WebSphere Commerce Server utilizando Loader package de WebSphere Commerce. Consulte el Capítulo 38, “Carga de grupos de elementos de base de datos de WebSphere Commerce”, en la página 405 y el apartado “Carga de grupos de elementos de base de datos” en la página 413 si desea obtener información sobre el proceso de carga para grupos de elementos de base de datos de WebSphere Commerce.

Loader Package proporciona seis programas de utilidad de línea de mandatos y dos herramientas administrativas relacionadas que se pueden utilizar para preparar los datos así como para cargarlos en la tienda. Estos mandatos y estas herramientas utilizan archivos de datos XML (Extensible Markup Language) para gestionar la información.

Descripción de la carga de datos en WebSphere Commerce

En la figura siguiente se muestran los procesos de preparación, carga y extracción de datos que puede realizar utilizando los mandatos de Loader Package.



Observe que la línea de puntos indica los dos procesos que se utilizan más comúnmente para cargar datos de tienda en una base de datos de WebSphere Commerce Server: la resolución de los identificadores y la carga de los datos. Estos procesos son el tema central de este capítulo.

Para obtener más información sobre cómo preparar los datos para cargarlos en una base de datos de WebSphere Commerce Server, consulte la Parte 5, “Visión general de los datos de la tienda”, en la página 103.

Los dos programas de utilidad de línea de mandatos siguientes de Loader Package se utilizan normalmente para cargar datos en una base de datos de WebSphere Commerce Server:

- **Mandato ID Resolve**

Para cargar datos XML en una base de datos de WebSphere Commerce Server utilizando Loader Package, los elementos XML deben correlacionarse directamente con el esquema de la base de datos de WebSphere Commerce Server de destino. Todos los elementos XML que tengan atributos correspondientes a claves exclusivas o primarias en el esquema de base de datos deben tener identificadores exclusivos; y todas las columnas que no puedan tener el valor nulo del esquema de base de datos deben tener atributos correspondientes definidos con valores no nulos. ID Resolver puede generar identificadores exclusivos para atributos de clave exclusiva o primaria de elementos XML calificadores.

Nota: En este documento, un identificador es un valor en una sola columna de una tabla de base de datos que da a cada fila una identidad exclusiva. Si utiliza ID Resolver para generar identificadores, obtiene un valor base de la tabla KEYS o SUBKEYS e incrementa el valor de forma secuencial para determinar un identificador para cada fila en la tabla de base de datos.

Para obtener información sobre este mandato, consulte los apartados “Mandato ID Resolve” en la página 365, “Utilización de los mandatos y scripts de Loader package” en la página 394 y “Ejemplos de resolución de identificadores” en la página 394.

- **Mandato Load**

Loader utiliza archivos XML válidos y con formato correcto como entrada para cargar datos en la base de datos. Los elementos del documento XML se correlacionan con los nombres de tabla de la base de datos y los atributos de elemento se correlacionan con las columnas.

Nota: Consulte las directrices XML de World Wide Web Consortium (W3C) si desea obtener una descripción de las restricciones en cuanto a validez y construcción correcta.

Para obtener información sobre este mandato, consulte los apartados “Mandato Load” en la página 374, “Utilización de los mandatos y scripts de Loader package” en la página 394 y “Ejemplo de carga de datos” en la página 402.

Estos mandatos son el tema principal de este capítulo.

Los siguientes programas de utilidad de línea de mandatos de Loader Package también se pueden utilizar para gestionar los datos:

- **Mandato DTD Generate**

DTD Generator genera una definición de tipo de documento (DTD) que describe las tablas y las columnas de la base de datos de destino en la que se deben cargar los datos XML. DTD Generator también puede generar un esquema XML para la base de datos.

DTD Generator puede crear una DTD basándose en el esquema de base de datos de WebSphere Commerce. Si utiliza las DTD proporcionadas con los archivadores de tienda de ejemplo y no modifica el esquema de base de datos, normalmente no necesitará generar una DTD utilizando DTD Generator.

Consulte el apartado “Mandato DTD Generate” en la página 384 para obtener más información.

- **Mandato Extract**

Extractor utiliza una consulta en una base de datos para extraer subconjuntos de datos seleccionados de la base de datos en forma de documento XML.

Puede utilizar este mandato para extraer datos de la base de datos en formato XML.

Consulte el apartado “Mandato Extract” en la página 387 para obtener más información.

- **Mandato Text Transform**

Text Transformer transforma los datos de un formato variable delimitado por caracteres a un formato de datos XML.

Si los datos no se pueden extraer directamente de una base de datos en formato XML, por ejemplo, puede guardar los datos en formato variable delimitado por caracteres y, a continuación, utilizar este mandato para transformarlos a formato XML.

Consulte el apartado “Mandato Text Transform” en la página 390 para obtener más información.

- **Mandato XML Transform**

XML Transformer transforma los datos de un documento XML a un formato XML alternativo. Utiliza XSL (Extensible Stylesheet Language) para definir las normas de correlación para la transformación.

Puede utilizar este mandato para convertir los datos XML a un formato que se correlacione directamente con el esquema de la base de datos de WebSphere Commerce en la que desea cargar los datos.

Consulte el apartado “Mandato XML Transform” en la página 392 para obtener más información.

Estos mandatos no son el tema principal de este capítulo. Para obtener más información sobre estos mandatos, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

Loader Package de WebSphere Commerce también incluye las herramientas siguientes para facilitar la administración de las funciones de gestión de datos:

- **Herramienta de Transformación de texto**

La herramienta de Transformación de texto le ayuda a procesar una transformación de datos entre un formato variable delimitado por caracteres y un formato de datos XML, utilizando el mandato Text Transform.

- **Editor XSL**

El editor XSL le proporciona una interfaz visual para editar archivos XSL que pueden ser utilizados por XML Transformer. Mediante la utilización del editor XSL, se establece la asociación de un elemento de una DTD de origen a un elemento de una DTD de destino al definir las normas de correlación para transformar datos entre formatos XML.

Estas herramientas no son el tema principal de este capítulo. Para obtener más información sobre estas herramientas, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

Mandatos de Loader Package para cargar datos de tienda

Mandato ID Resolve

Este mandato genera identificadores exclusivos para los elementos de datos XML que los necesitan para que dichos elementos puedan cargarse en una base de datos. Si los datos XML fuente ya proporcionan los identificadores exclusivos necesarios, no tiene que ejecutar ID Resolver.

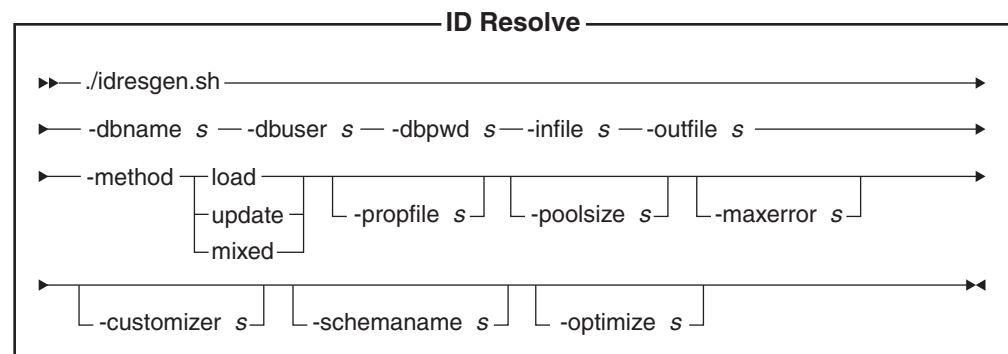
El esquema de base de datos de WebSphere Commerce define en las tablas las claves primarias y externas que se utilizan para representar diversas relaciones entre las tablas. Por este motivo, los elementos XML de WebSphere Commerce deben contener atributos que se correspondan con identificadores exclusivos. En la base de datos de WebSphere Commerce Server, las tablas cuyos identificadores deben resolverse son las definidas en las tablas KEYS y SUBKEYS. Estas tablas se denominan *tablas primarias* dentro de WebSphere Commerce. Para obtener más información sobre las tablas KEYS y SUBKEYS, consulte la ayuda en línea de WebSphere Commerce.

Nota: Si es necesario resolver los identificadores de una tabla que no está definida en la tabla KEYS o SUBKEYS, añada la tabla a la tabla SUBKEYS antes de ejecutar ID Resolver.

Dado que los atributos y elementos XML de WebSphere Commerce están diseñados para poder desplazarse entre bases de datos y entre instancias de base de datos, sus identificadores normalmente se representan utilizando alias internos. Para poder cargar los datos en cualquier base de datos de WebSphere Commerce Server, estos alias deben transformarse en identificadores numéricos válidos. Para obtener más información, consulte el Apéndice B, "Creación de los datos", en la página 465.

Notas:





1. El diagrama siguiente está pensado principalmente para servir de referencia para los parámetros del mandato.
2. Los nombres de archivo especificados como parámetros para este mandato pueden ir precedidos por vías de acceso relativas o absolutas.







Nota: Los nombres de archivo especificados como parámetros para este mandato pueden ir precedidos por vías de acceso relativas o absolutas.

Valores de los parámetros:

-dbname

   El nombre de la base de datos de destino.
 Este es el nombre de la base de datos como se muestra en el directorio de base de datos relacional (WRKRDBDIRE).

-dbuser

   El nombre del usuario que se conecta a la base de datos.  Generalmente, es el mismo que el nombre de usuario de la instancia.

-dbpwd

Contraseña del usuario que se conecta a la base de datos

-infile Nombre del documento XML de entrada que contiene los registros de tabla

-outfile

Nombre del archivo XML de salida que se genera; este archivo puede utilizarse como entrada a Loader

-method

Método que se ha de utilizar al procesar el archivo de entrada. El mandato puede tratar el archivo de entrada como si los registros no existieran en la base de datos (carga) o como si ya hubieran identificadores para los objetos de entrada (actualización). Utilice el método combinado cuando algunos registros no existan en la base de datos y otros sí. El método por omisión es el método load.

-profile

Archivo de texto que contiene las propiedades Java con el formato de parejas nombre=valor. Este archivo de propiedades establece la forma en que ID Resolver resuelve los identificadores. Se utiliza para describir qué columnas de una entrada primaria deben utilizarse como consultas para las tablas que necesitan el identificador de una fila primaria. Este archivo define los nombres de columna para búsquedas del identificador de clave externa y el predicado select en consultas de la tabla principal (por ejemplo, CATEGORY y PRODUCT). Puede omitir las entradas de este archivo para tablas que tengan definido un índice exclusivo que no incluya el identificador. Este parámetro es opcional. IdResolveKeys.properties es el archivo por omisión. Se puede especificar el archivo de propiedades como se muestra en cualquiera de los ejemplos siguientes:

```
-profile dir_instalación_WC/mi_directorio/nombre_archivo.properties  
-profile dir_instalación_WC/mi_directorio/nombre_archivo
```



```
-profile dir_usuario_WC/mi_directorio/nombre_archivo.properties  
-profile dir_usuario_WC/mi_directorio/nombre_archivo
```

Si este archivo existe en el directorio actual, se puede especificar el mismo archivo tal como se muestra en el ejemplo siguiente:

```
-profile nombre_archivo.properties
```

Si este archivo se encuentra en el directorio especificado en la variable de entorno del sistema CLASSPATH, se puede especificar el mismo archivo, como se muestra en el ejemplo siguiente:

```
-profile nombre_archivo
```

donde, *mi_directorio* es un directorio definido por el usuario y *nombre_archivo* es el nombre del archivo de propiedades que va a utilizar.

Para obtener más información sobre la creación y especificación de un nuevo archivo de propiedades para utilizarlo con ID Resolver, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

-poolsize





Número de identificadores que se han de reservar. Este parámetro es opcional. El número por omisión es 50.

-maxerror

Número de errores después del cual ID Resolver finalizará. Este parámetro es opcional. El valor por omisión es 1.

-customizer




Nombre del archivo de propiedades de personalización que se va a utilizar. Este parámetro es opcional. El archivo de propiedades de personalización establece la forma de funcionamiento de ID Resolver. El archivo por omisión es:


   DB2ConnectionCustomizer.properties
 ISeries_RESWCSID_Customizer.properties

Si ha configurado la instancia para que utilice el controlador Toolbox, entonces utilice el archivo de personalización `Toolbox_RESWCSID_Customizer` que se proporciona para el controlador Toolbox. Asimismo debe especificar el nombre de sistema principal para el parámetro `-dbname`. El siguiente es un ejemplo de cómo invocar el script `idresgen.sh`:

```
./idresgen.sh -dbname MI.NOMBSISPRAL.CA -dbuser instance -dbpwd contraseña  
-infile /path/infile.xml -outfile /path/outfile.xml -method sqlimport  
-customizer Toolbox_RESWCSID_Customizer
```

Se puede especificar el archivo de propiedades de personalización como se muestra en los ejemplos siguientes:

```
-customizer dir_instalación_WC/mi_directorio/nombre_archivo.properties  
-customizer dir_instalación_WC/mi_directorio/nombre_archivo  
  
-customizer WC_userdir/mi_directorio/nombre_archivo.properties  
-customizer dir_usuario_WC/mi_directorio/nombre_archivo
```

donde, *mi_directorio* es un directorio definido por el usuario y *nombre_archivo* es el nombre del archivo de propiedades que desea utilizar.

Si este archivo existe en el directorio actual, se puede especificar el mismo archivo tal como se muestra en el ejemplo siguiente:

```
-customizer nombre_archivo.properties
```

Si este archivo se encuentra en el directorio especificado en la variable de entorno del sistema `CLASSPATH`, se puede especificar el mismo archivo, como se muestra en el ejemplo siguiente:

```
-customizer nombre_archivo
```

Para obtener más información sobre la creación y especificación de un nuevo archivo de propiedades, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

-schemaname

Nombre del esquema de base de datos de destino. Este parámetro es opcional.

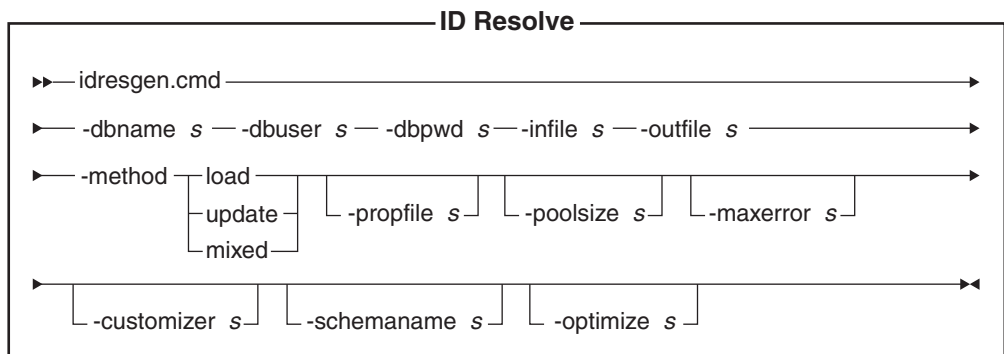
Si no se especifica este parámetro al ejecutar el mandato, éste busca una pareja nombre=valor en el archivo de propiedades de personalización que especifica el valor de SchemaName. Si esta pareja se encuentra en el archivo de propiedades, el mandato utiliza el valor especificado. Si no se especifica ni en un archivo de propiedades ni en la línea de mandatos, el mandato toma como valor por omisión el nombre de esquema de la tabla KEYS de la base de datos. 400 Si no se especifica el parámetro -schemaname ni en un archivo de propiedades ni en la línea de mandatos, el mandato toma por omisión el valor del parámetro -dbuser.

-optimize

-optimize no

El IdResolver ignorará la comprobación de registros duplicados antes de grabar registros resueltos en el archivo de salida. Esta opción permite al usuario desactivar la característica de optimización en IdResolver.

► Windows



Valores de los parámetros:

-dbname

Nombre de la base de datos de destino

-dbuser

Nombre del usuario que se conecta a la base de datos

-dbpwd

Contraseña del usuario que se conecta a la base de datos

-infile Nombre del documento XML de entrada que contiene los registros de tabla

-outfile

Nombre del archivo XML de salida que se genera; este archivo puede utilizarse como entrada a Loader

-method

Método que se ha de utilizar al procesar el archivo de entrada. El mandato puede tratar el archivo de entrada como si los registros no existieran en la base de datos (carga) o como si ya hubieran identificadores para los objetos de entrada (actualización). Utilice el método combinado cuando algunos registros no existan en la base de datos y otros sí. El método por omisión es el método load.

-profile

Archivo de texto que contiene las propiedades Java con el formato de parejas nombre=valor. Este archivo de propiedades establece la forma en que ID Resolver resuelve los identificadores. Se utiliza para describir qué columnas de una entrada primaria deben utilizarse como consultas para las tablas que necesitan el identificador de una fila primaria. Este archivo define los nombres de columna para búsquedas del identificador de clave externa y el predicado select en consultas de la tabla principal (por ejemplo, CATEGORY y PRODUCT). Puede omitir las entradas de este archivo para tablas que tengan definido un índice exclusivo que no incluya el identificador. Este parámetro es opcional. El archivo por omisión es `IdResolveKeys.properties`. Se puede especificar el archivo de propiedades como se muestra en cualquiera de los ejemplos siguientes:

```
-profile dir_instalación_WC\mi_directorio\ nombre_archivo.properties
```

```
-profile dir_instalación_WC\mi_directorio\nombre_archivo
```

Si este archivo existe en el directorio actual, se puede especificar el mismo archivo tal como se muestra en el ejemplo siguiente:

```
-profile nombre_archivo.properties
```

Si este archivo se encuentra en el directorio especificado en la variable de entorno del sistema CLASSPATH, se puede especificar el mismo archivo, como se muestra en el ejemplo siguiente:

```
-profile nombre_archivo
```

donde, *mi_directorio* es un directorio definido por el usuario y *nombre_archivo* es el nombre del archivo de propiedades que desea utilizar.

Para obtener más información sobre la creación y especificación de un nuevo archivo de propiedades para utilizarlo con ID Resolver, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

-poolsize

Número de identificadores que se han de reservar. Este parámetro es opcional. El número por omisión es 50.

-maxerror

Número de errores después del cual ID Resolver finalizará. Este parámetro es opcional. El valor por omisión es 1.

-customizer

Nombre del archivo de propiedades de personalización que se va a utilizar. Este parámetro es opcional. El archivo de propiedades de personalización establece la forma de funcionamiento de ID Resolver.

DB2ConnectionCustomizer.properties es el archivo por omisión. Se puede especificar el archivo de propiedades de personalización como se muestra en los ejemplos siguientes:

```
-customizer dir_instalación_WC\mi_directorio\nombre_archivo.properties
```

```
-customizer dir_instalación_WC\mi_directorio\nombre_archivo
```

Si este archivo existe en el directorio actual, se puede especificar el mismo archivo tal como se muestra en el ejemplo siguiente:

```
-customizer nombre_archivo.properties
```

Si este archivo se encuentra en el directorio especificado en la variable de entorno del sistema CLASSPATH, se puede especificar el mismo archivo, como se muestra en el ejemplo siguiente:

```
-customizer nombre_archivo
```

donde, *mi_directorio* es un directorio definido por el usuario y *nombre_archivo* es el nombre del archivo de propiedades que va a utilizar. Para obtener más información sobre la creación y especificación de un nuevo archivo de propiedades, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

-schemaname

Nombre del esquema de base de datos de destino. Este parámetro es opcional.

Si no se especifica este parámetro al ejecutar el mandato, éste busca una pareja nombre=valor en el archivo de propiedades de personalización que especifica el valor de SchemaName. Si esta pareja se encuentra en el archivo de propiedades, el mandato utiliza el valor especificado. Si no se especifica ni en un archivo de propiedades ni en la línea de mandatos, el mandato toma como valor por omisión el nombre de esquema de la tabla KEYS de la base de datos.

-optimize

```
-optimize no
```

El IdResolver ignorará la comprobación de registros duplicados antes de grabar registros resueltos en el archivo de salida. Esta opción permite al usuario desactivar la característica de optimización en IdResolver.

Técnicas de resolución:

ID Resolver resuelve los identificadores utilizando una combinación de dos o tres de las técnicas siguientes, en función de si se utiliza un archivo de propiedades o no.

• Resolución de alias interno

Cuando se utiliza la resolución de ID de alias interno, se sustituye un alias por la clave (identificador) exclusiva en el documento XML fuente. Este alias se utiliza luego en cualquier otra parte del archivo XML para hacer referencia a ese elemento.

Los alias internos deben utilizarse de forma coherente en todo el archivo XML. Por ejemplo, si el alias de un ID de listín, ADDRBOOK_ID, es @addrbook_1, todas las referencias de clave externa a dicho ID del archivo deberán utilizar @addrbook_1.

Tenga en cuenta que los alias son transitorios para el archivo XML específico. No se guardan y no se pueden utilizar en un archivo XML independiente sin entrar el alias otra vez. Durante la publicación en la Consola de administración, la publicación concatena los archivos XML para que pueda producirse la resolución en todos los datos.

- **Resolución de índice exclusivo**

ID Resolver también puede analizar el esquema de base de datos para determinar si existe un índice exclusivo que satisfaga los requisitos. ID Resolver sólo busca un índice exclusivo cuando no hay ninguna entrada en el archivo de propiedades para la tabla que se está analizando o cuando no hay ningún archivo de propiedades. Si estas condiciones son verdaderas, se realiza una comprobación de índice exclusivo. El índice exclusivo se considera válido si existe y no incluye la clave primaria para la tabla.

- **Especificación de archivo de propiedades**

ID Resolver le permite utilizar un archivo de propiedades Java alternativo para describir qué columnas de una entrada primaria deben utilizarse como consultas para las tablas que necesitan el identificador de una fila primaria.

Los archivadores de tienda de ejemplo proporcionados con WebSphere Commerce utilizan alias internos en sus archivos XML. Esto permite que los archivadores de tienda se puedan transportar entre bases de datos. Aunque las técnicas de especificación de índice exclusivo y de archivos de propiedades también permiten la portabilidad entre bases de datos, un usuario puede cambiar las columnas exclusivas y causar problemas cuando estas técnicas se utilicen posteriormente para la resolución de ID. Si, por ejemplo, un usuario cambia una columna exclusiva, el nombre de la columna debe cambiarse en la definición del archivo de propiedades. Sin embargo, con la técnica de alias interno, un cambio en la base de datos no requiere un cambio en los archivos XML o de propiedades. Durante la publicación desde la Consola de administración de WebSphere Commerce o mediante Loader package, ID Resolver sustituye el alias por un valor exclusivo. Una vez que se han cargado los datos, los alias son transparentes para el usuario. Para obtener más información, consulte el Apéndice B, "Creación de los datos", en la página 465.

ID Resolver utiliza el proceso siguiente:

- Si los datos XML de entrada tienen un elemento de una tabla primaria que ya tiene un identificador con codificación fija (por ejemplo "12345"), ID Resolver no crea un identificador nuevo para dicho elemento.
- Si los datos XML de entrada tienen un elemento de una tabla primaria que no tiene ningún identificador, ID Resolver examina la base de datos para ver si ya existe una fila para dicho elemento.

Para buscar el elemento en la base de datos es necesario que se utilicen otras columnas del elemento para formar una clave exclusiva. Estas otras columnas pueden especificarse en el archivo de propiedades o se puede permitir que ID Resolver determine qué columnas se deben utilizar.

- Si se está utilizando un archivo de propiedades y existe una entrada en el archivo de propiedades para la tabla que se está analizando, ID Resolver utiliza las columnas especificadas en el archivo de propiedades para formar la clave exclusiva.

- Si no se está utilizando ningún archivo de propiedades o no hay ninguna entrada en el archivo de propiedades para la tabla que se está analizando, ID Resolver utiliza la resolución de índice exclusivo.

La resolución de índice exclusivo utiliza cualquiera de los índices exclusivos especificados de una tabla como medio para localizar el identificador. Por ejemplo, MEMBER_ID más IDENTIFIER es un índice exclusivo de la tabla CATALOG y por consiguiente se puede utilizar como punto de resolución para la clave externa CATALOG_ID de la tabla CATALOGDSC.

Se considera que el elemento ya existe en la base de datos si hay una fila con la misma clave exclusiva; de lo contrario, se considera como una parte nueva de datos.

- Si el elemento ya existe en una fila de la base de datos, su identificador se recupera y se guarda para poderse utilizar posteriormente. De lo contrario, ID Resolver genera un identificador nuevo utilizando un valor disponible de la tabla KEYS o SUBKEYS.
- Si ha especificado un alias interno para el elemento (por ejemplo "@store_id_1") en el documento XML, dicho alias se asocia con el identificador para poder buscar ese identificador posteriormente utilizando el mismo alias interno.
- Los elementos de documento XML subsiguientes que necesitan hacer referencia a un elemento de la tabla primaria utilizan el alias interno, si el elemento de tabla primaria tenía alias interno (por ejemplo "@store_id_1") o utilizan los valores de las columnas de búsqueda si dicho elemento no tenía alias interno (por ejemplo "@WC2001@100"). En cualquiera de los casos, se utiliza el valor especificado para buscar el identificador real y el valor se sustituye por dicho identificador.
- Cuando se genera el documento XML de salida, todos los elementos de tabla primaria contienen identificadores reales y todos los elementos que hacen referencia a dichos elementos de tabla primaria hacen referencia a ellos utilizando los identificadores reales, no los alias internos ni los valores de columna de búsqueda mencionados anteriormente. Esto constituye el documento XML totalmente resuelto.

Métodos para el mandato ID Resolve:

El mandato ID Resolve le permite elegir entre el método load, el método update o el método combinado para procesar el archivo de entrada.

Método load:

El método load para ID Resolver se utiliza para generar identificadores nuevos para todos los registros nuevos que se cargan en la base de datos.

Nota: Si especifica el método load para ID Resolver, los registros del archivo de entrada no deben existir aún en la base de datos. Si se utiliza el método de carga con ID Resolver y ya existe un registro del archivo XML fuente en la base de datos de destino, Loader generará un error al cargar los datos. ID Resolver asignará una clave primaria nueva al registro del archivo XML durante la resolución de ID; pero cuando cargue los datos en la base de datos, se generará un error. Loader no se detendrá en el momento de procesar el registro duplicado, pero informará de un error y el registro duplicado no se cargará en la base de datos.

El ejemplo siguiente se utiliza para generar identificadores para elementos de datos que son nuevos en la base de datos:

-    

```
./idresgen.sh -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method load -customizer customizer -schemaname wcsadmin
```

- ▶ **Windows**

```
idresgen -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method load -customizer customizer -schemaname wcsadmin
```

Nota: Consulte el apartado “Utilización de los mandatos y scripts de Loader package” en la página 394 para obtener información sobre la ubicación del mandato o script apropiados de ID Resolve.

Método update:

Si especifica el método update para ID Resolver, los registros del archivo de entrada ya deben existir en la base de datos. ID Resolver localiza los identificadores en la base de datos, tal como se describe en la página 371. Si no existe un registro en la base de datos, ID Resolver no puede resolver el identificador para este registro e indica que se ha producido un error.

El ejemplo siguiente se utiliza para localizar identificadores para elementos de datos que ya existen en la base de datos:

- ▶ **AIX** ▶ **Linux** ▶ **400** ▶ **Solaris**

```
./idresgen.sh -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method update -customizer customizer -schemaname wcsadmin
```

- ▶ **Windows**

```
idresgen -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method update -customizer customizer -schemaname wcsadmin
```

Nota: Consulte el apartado “Utilización de los mandatos y scripts de Loader package” en la página 394 para obtener información sobre la ubicación del mandato o script apropiados de ID Resolve.

Método combinado:

Si el archivo de datos de entrada contiene registros que ya existen en la base de datos así como algunos registros que son nuevos, se deberá ejecutar ID Resolver utilizando el método combinado. Con este método, ID Resolver crea identificadores nuevos para los registros sólo si los registros no existen en la base de datos. De lo contrario, se obtiene el identificador existente de la base de datos. El ejemplo siguiente se utiliza para generar identificadores para datos nuevos y para localizar identificadores para elementos de datos que ya existen en la base de datos:

- ▶ **AIX** ▶ **Linux** ▶ **400** ▶ **Solaris**

```
./idresgen.sh -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method mixed -customizer customizer -schemaname wcsadmin
```

- ▶ **Windows**

```
idresgen -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method mixed -customizer customizer -schemaname wcsadmin
```

Notas:

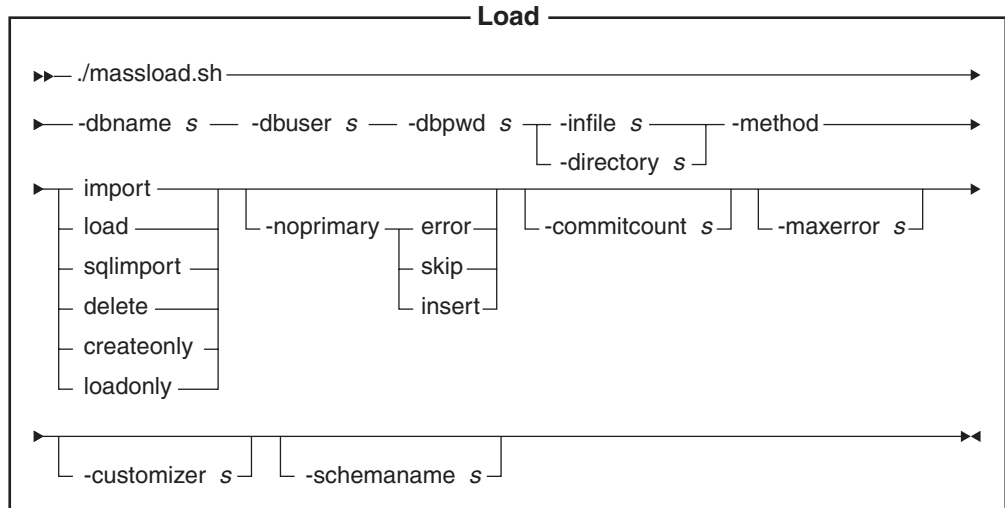
1. Consulte el apartado “Utilización de los mandatos y scripts de Loader package” en la página 394 para obtener información sobre la ubicación del mandato o script apropiados de ID Resolve.
2. Se recomienda el método combinado para la Consola de administración.

Si desea obtener información detallada sobre cómo configurar y personalizar los archivos utilizados para ejecutar este mandato, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

Mandato Load

Este mandato carga un archivo de entrada XML en una base de datos de destino.

▶ AIX ▶ Linux ▶ 400 ▶ Solaris



Notas:

1. El diagrama anterior está pensado principalmente para servir de referencia para los parámetros de mandato.
2. Los nombres de archivo especificados como parámetros para este mandato pueden ir precedidos por vías de acceso relativas o absolutas.

Valores de los parámetros:

-dbname

▶ AIX ▶ Linux ▶ Solaris El nombre de la base de datos de destino

▶ 400 Este es el nombre de la base de datos como se muestra en el directorio de base de datos relacional (WRKRDBDIRE).

-dbuser

▶ AIX ▶ Linux ▶ Solaris El nombre del usuario que se conecta a la base de datos.

▶ 400 Generalmente, es el mismo que el nombre de usuario de la instancia.

-dbpwd

Contraseña del usuario que se conecta a la base de datos

-infile Nombre del archivo XML de entrada

-directory

Puede utilizar el parámetro `-infile` si está utilizando cualquier opción que no sea `loadonly` para el parámetro `-method` como se describe a continuación. Si utiliza el método `loadonly`, debe sustituir el parámetro `-infile` por el parámetro `-directory` o se producirá un error. Para el valor del parámetro `-directory` cuando utilice el método `loadonly`, especifique la vía de acceso totalmente calificada del directorio `MassLoadOutputFiles` que se ha creado utilizando el método `createonly` como se describe a continuación.

-method

Modalidad de operación que utilizará Loader al modificar la base de datos con datos de entrada

load El método load utiliza el cargador nativo del proveedor de la base de datos. Puede utilizar el método load para bases de datos Oracle locales y remotas, pero el método load sólo se puede utilizar para bases de datos DB2 locales.

▶ 400 El método load no soporta los datos de bits ni los campos DBCLOB.

import

El método import utiliza la opción import o update si el proveedor de la base de datos la facilita. Si no está disponible la opción de importar o actualizar, se utilizan las sentencias SQL mediante JDBC para actualizar la base de datos. El valor por omisión es import.

▶ 400 El método de importación solamente se puede utilizar en las bases de datos locales.

sqlimport

El método import de SQL (sqlimport) se puede utilizar con bases de datos locales y remotas.

delete

El método delete suprime los datos de la base de datos.

createonly

Para mejorar el rendimiento durante la creación de la instancia, utilice el método createonly. Utilice el método createonly para crear archivos de datos de carga masiva (MLD) sin cargar los datos en la base de datos. Los archivos que se crean cuando se utiliza este método (archivos .mld y .cmd) se colocan en un directorio llamado MassLoadOutputFiles. ▶ AIX ▶ Linux ▶ Solaris Este directorio se crea como un subdirectorio del directorio desde el que se ha ejecutado el mandato Load, que es su directorio de trabajo. Como resultado, el directorio de trabajo debe ser regrabable. ▶ 400

Este directorio se crea como subdirectorio del directorio temp y está situado en el directorio raíz de la instancia. La ubicación del directorio por omisión será *dir_usuario_WC/instances/nombre_instancia/temp/MassLoadOutputFiles*.

El siguiente es un ejemplo de una ejecución del mandato Load utilizando el método createonly:

```
./massload.sh -dbname mall -dbuser db2admin -dbpwd db2admin  
-infile dir_instalación_WC/data/example.xml  
-method createonly
```

▶ 400

```
./massload.sh -dbname mall -dbuser db2admin -dbpwd db2admin  
-infile dir_usuario_WC/data/example.xml  
-method createonly
```

Después puede utilizar su programa de utilidad de carga de base de datos nativo para cargar los archivos MLD que ha creado en

una base de datos WebSphere Commerce ejecutando el mandato Load con el método loadonly que se describe a continuación.

Nota: El programa obtiene información sobre el programa de utilidad de carga de base de datos nativo que su producto de base de datos utiliza del archivo de propiedades de personalización.

loadonly

Utilice el método loadonly para cargar los archivos MLD que se han creado utilizando el método createonly descrito anteriormente. Cuando utilice el método loadonly, debe utilizar también el parámetro -directory o se producirá un error.

Nota: El parámetro -directory sustituye al parámetro -infile que se especifica cuando se utiliza cualquier método que no sea loadonly.

Para el valor del parámetro -directory, debe especificar la vía de acceso totalmente calificada del directorio MassLoadOutputFiles que se ha creado mediante el método createonly.

A continuación se muestra un ejemplo de cómo ejecutar el mandato Load utilizando el método loadonly (y el parámetro -directory obligatorio):

```
./massload.sh -dbname mall -dbuser db2admin -dbpwd db2admin  
-method loadonly -directory dir_instal_WC/bin/MassLoadOutputFiles  
-schemaname wcsadmin
```

Especifique siempre el nombre del esquema de base de datos de destino utilizando el parámetro -schemaname cuando ejecute el mandato Load con este método. De lo contrario, el programa utiliza el nombre del esquema de base de datos que se ha obtenido al crear originalmente el directorio MassLoadOutputFiles y sus archivos. Cuando utilice el método loadonly, los errores y otros tipos de mensajes se guardan en archivos que tienen una extensión .log. Estos archivos de anotaciones cronológicas se graban en el directorio MassLoadOutputFiles especificado para el parámetro -directory.

Utilice el método loadonly **solamente** para la creación de instancias. Si lo utiliza en cualquier otra ocasión, el resultado puede no ser el deseado.

-noprimary

Acción que Loader debe realizar cuando falte la clave primaria para un registro del archivo de entrada

- La opción "error" indica que se debe informar de que falta la clave primaria como un error y que se debe terminar.
- La opción "skip" se salta los registros del archivo de entrada que no tengan una clave primaria.
- La opción "insert" intenta insertar o suprimir los datos.

Este parámetro es opcional. La acción por omisión es "error".

-commitcount

Número de registros procesados antes de que se comprometa la base de datos cuando se utiliza el método import de SQL. Este parámetro es opcional. El número por omisión es 1.

-maxerror

Número de errores después de los cuales Loader finalizará el método de operación de importación de SQL. Este parámetro es opcional. El valor por omisión es 1.

-customizer

Nombre del archivo de propiedades de personalización que se va a utilizar. Este parámetro es opcional. El archivo de propiedades de personalización establece el funcionamiento de Loader. El archivo por omisión es:

   MassLoadCustomizer.properties

 ISeries_LODWCSDTA_Customizer.properties

Si ha configurado la instancia para que utilice el controlador Toolbox, entonces utilice el archivo de personalización

Toolbox_LODWCSDTA_Customizer que se proporciona para el controlador Toolbox. Asimismo debe especificar el nombre de sistema principal para el parámetro `-dbname`. El siguiente es un ejemplo de invocación del script `massload.sh`:

```
./massload.sh -dbname MI.NOMBSISPRAL.CA -dbuser instance -dbpwd micontraseña  
-method sqlimport -customizer Toolbox_LODWCSDTA_Customizer  
-infile /path/file.xml
```

Se puede especificar el archivo de propiedades de personalización como se muestra en el ejemplo siguiente:

```
-customizer dir_instalación_WC/mi_directorio/nombre_archivo.properties
```

```
-customizer dir_instalación_WC/mi_directorio/nombre_archivo
```



```
-customizer dir_usuario_WC/mi_directorio/nombre_archivo.properties
```

```
-customizer dir_usuario_WC/mi_directorio/nombre_archivo
```

Si este archivo existe en el directorio actual, se puede especificar el mismo archivo tal como se muestra en el ejemplo siguiente:

```
-customizer nombre_archivo.properties
```

Si este archivo se encuentra en el directorio especificado en la variable de entorno del sistema `CLASSPATH`, se puede especificar el mismo archivo, como se muestra en el ejemplo siguiente:

```
-customizer nombre_archivo
```

donde, *mi_directorio* es un directorio definido por usuario y *nombre_archivo* es el nombre del archivo de propiedades que va a utilizar. Para obtener más información acerca de cómo crear y especificar un nuevo archivo de propiedades de personalización, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

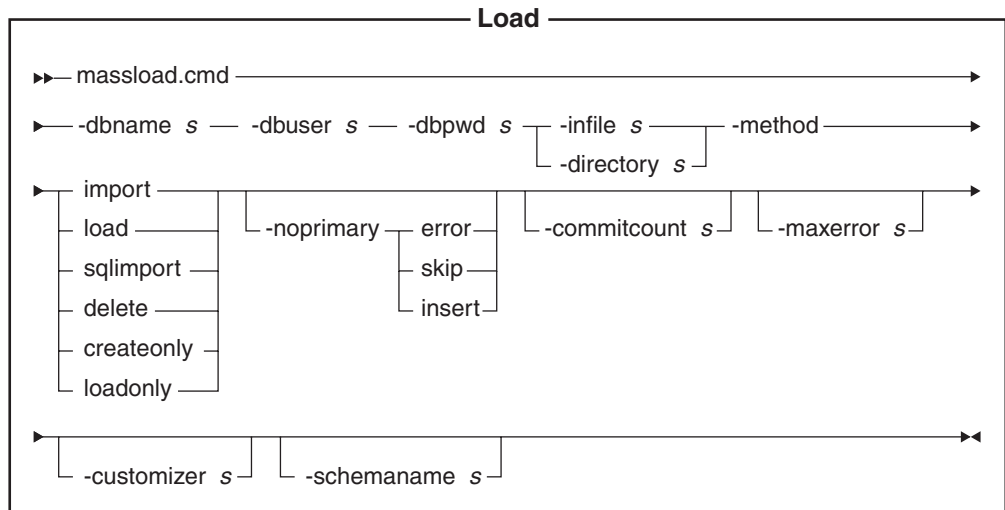
-schemaname

Nombre del esquema de base de datos de destino. Este parámetro es opcional.

Si no se especifica este parámetro al ejecutar el mandato, éste busca una pareja nombre=valor en el archivo de propiedades de personalización que especifica el valor de `SchemaName`. Si esta pareja se encuentra en el archivo de propiedades, el mandato utiliza el valor especificado. Si no se especifica ni en un archivo de propiedades ni en la línea de mandatos, el

mandato toma como valor por omisión el nombre de esquema de la tabla KEYS de la base de datos. **400** Si no se especifica el parámetro `-schemaname` ni en un archivo de propiedades ni en la línea de mandatos, el mandato toma por omisión el valor del parámetro `-dbuser`.

Windows



Nota: Los nombres de archivo especificados como parámetros para este mandato pueden ir precedidos por vías de acceso relativas o absolutas.

Valores de los parámetros:

-dbname

Nombre de la base de datos de destino

-dbuser

Nombre del usuario que se conecta a la base de datos

-dbpwd

Contraseña del usuario que se conecta a la base de datos

-infile Nombre del archivo XML de entrada

-directory

Puede utilizar el parámetro `-infile` si está utilizando cualquier opción que no sea `loadonly` para el parámetro `-method` como se describe a continuación. Si utiliza el método `loadonly`, debe sustituir el parámetro `-infile` por el parámetro `-directory` o se producirá un error. Para el valor del parámetro `-directory` cuando utilice el método `loadonly`, especifique la vía de acceso totalmente calificada del directorio `MassLoadOutputFiles` que se ha creado utilizando el método `createonly` como se describe a continuación.

-method

Modalidad de operación que utilizará Loader al insertar datos en la base de datos.

load El método `load` utiliza el cargador nativo del proveedor de la base de datos. Puede utilizar el método `load` para bases de datos Oracle locales y remotas, pero el método `load` sólo se puede utilizar para bases de datos DB2 locales. Aunque el método de importación se

puede utilizar para cargar datos en bases de datos locales o remotas, normalmente se utiliza para cargar datos en bases de datos DB2 remotas.

import

El método import utiliza la opción import o update si el proveedor de la base de datos la facilita. Si no está disponible la opción de importar o actualizar, se utilizan las sentencias SQL mediante JDBC para actualizar la base de datos. El valor por omisión es import.

sqlimport

El método import de SQL (sqlimport) se puede utilizar con bases de datos locales y remotas.

delete

El método de supresión (delete) suprime los datos de la base de datos.

createonly

Para mejorar el rendimiento durante la creación de la instancia, utilice el método createonly. Utilice el método createonly para crear archivos de datos de carga masiva (MLD) sin cargar los datos en la base de datos. Los archivos que se crean cuando se utiliza este método (archivos .mld y .cmd) se colocan en un directorio llamado MassLoadOutputFiles. Este directorio se crea como un subdirectorio del directorio desde el que ejecuta el mandato Load, que es su directorio de trabajo. Como resultado, el directorio de trabajo debe ser regrabable.

El siguiente es un ejemplo de una ejecución del mandato Load utilizando el método createonly:

```
massload -dbname mall -dbuser db2admin -dbpwd db2admin -infile
dir_instalación_WC\data\example.xml
-method createonly
```

Después puede utilizar su programa de utilidad de carga de base de datos nativo para cargar los archivos MLD que ha creado en una base de datos WebSphere Commerce ejecutando el mandato Load con el método loadonly que se describe a continuación.

Nota: El programa obtiene información sobre el programa de utilidad de carga de base de datos nativo que su producto de base de datos utiliza del archivo de propiedades de personalización.

loadonly

Utilice el método loadonly para cargar los archivos MLD que se han creado utilizando el método createonly descrito anteriormente. Cuando utilice el método loadonly, debe utilizar también el parámetro -directory o se producirá un error.

Nota: El parámetro -directory sustituye al parámetro -infile que se especifica cuando se utiliza cualquier método que no sea loadonly.

Para el valor del parámetro -directory, debe especificar la vía de acceso totalmente calificada del directorio MassLoadOutputFiles que se ha creado mediante el método createonly.

A continuación se muestra un ejemplo de cómo ejecutar el mandato Load utilizando el método loadonly (y el parámetro -directory obligatorio):

```
massload -dbname mall -dbuser db2admin -dbpwd db2admin -method loadonly -directory dir_instalación_WC\bin\MassLoadOutputFiles -schemaname wcsadmin
```

Especifique siempre el nombre del esquema de base de datos de destino utilizando el parámetro -schemaname cuando ejecute el mandato Load con este método. De lo contrario, el programa utiliza el nombre del esquema de base de datos que se ha obtenido al crear originalmente el directorio MassLoadOutputFiles y sus archivos. Cuando utilice el método loadonly, los errores y otros tipos de mensajes se guardan en archivos que tienen una extensión .log. Estos archivos de anotaciones cronológicas se graban en el directorio MassLoadOutputFiles especificado para el parámetro -directory.

Utilice el método loadonly **solamente** para la creación de instancias. Si lo utiliza en cualquier otra ocasión, el resultado puede no ser el deseado.

-noprimary

Acción que Loader debe realizar cuando falte la clave primaria para un registro del archivo de entrada

- La opción "error" indica que se debe informar de que falta la clave primaria como un error y que se debe terminar.
- La opción "skip" se salta los registros del archivo de entrada que no tengan una clave primaria.
- La opción insert intenta procesar (insertar o suprimir) los datos.

Este parámetro es opcional. La acción por omisión es "error".

-commitcount

Número de registros procesados antes de que se comprometa la base de datos cuando se utiliza el método import de SQL. Este parámetro es opcional. El número por omisión es 1.

-maxerror

Número de errores después de los cuales Loader finalizará el método de operación de importación de SQL. Este parámetro es opcional.

-customizer

Nombre del archivo de propiedades de personalización que se va a utilizar. Este parámetro es opcional. El archivo de propiedades de personalización establece el funcionamiento de Loader. El archivo por omisión es MassLoadCustomizer.properties. Se puede especificar el archivo de propiedades de personalización como se muestra en el ejemplo siguiente:

```
-customizer dir_instalación_WC\mi_directorio\nombre_archivo.properties
```

Si este archivo existe en el directorio actual, se puede especificar el mismo archivo tal como se muestra en el ejemplo siguiente:

```
-customizer nombre_archivo.properties
```

Si este archivo se encuentra en el directorio especificado en la variable de entorno del sistema CLASSPATH, se puede especificar el mismo archivo, como se muestra en el ejemplo siguiente:

```
-customizer nombre_archivo
```

donde, *mi_directorio* es un directorio definido por el usuario y *nombre_archivo* es el nombre del archivo de propiedades que va a utilizar.

Para obtener más información sobre cómo crear y especificar un nuevo archivo de propiedades de personalización, consulte la versión más reciente de la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

-schemaname

Nombre del esquema de base de datos de destino. Este parámetro es opcional.

Si no se especifica este parámetro al ejecutar el mandato, éste busca una pareja nombre=valor en el archivo de propiedades de personalización que especifica el valor de SchemaName. Si esta pareja se encuentra en el archivo de propiedades, el mandato utiliza el valor especificado. Si no se especifica ni en un archivo de propiedades ni en la línea de mandatos, el mandato toma como valor por omisión el nombre de esquema de la tabla KEYS de la base de datos.

Métodos para el mandato Load:

Antes de cargar datos, debe determinar cuál de los métodos de proceso producirá los mejores resultados.


Método load:

Tome en consideración el método load en cualquiera de las situaciones siguientes:



- Los datos fuente están limpios y la base de datos no contiene datos

Nota: Datos limpios son datos que no violan ninguna de las restricciones de las tablas en las que se están cargando.





- Los datos fuente están limpios y la base de datos no contiene los datos que se están cargando
- Los datos fuente están limpios, una o varias de las tablas de destino no contienen claves primarias y la base de datos no contiene los datos que se están cargando
- La base de datos es una base de datos DB2 local
- La base de datos es una base de datos Oracle local o remota
- Mientras está teniendo lugar la carga no están accediendo a la base de datos otros usuario u otras aplicaciones

 Con el método de load, se cargan datos en la base de datos. Si los datos ya existen, el mandato falla como resultado de un error de clave duplicada y se visualiza un mensaje de error de duplicado.

Existen las restricciones siguientes al utilizar el método load:

-  El método load no puede insertar ni actualizar datos en campos de datos de bits ni en campos DBCLOB. Solamente se insertan registros nuevos en la base de datos. Los registros existentes darán errores.
-  Con el método load, sólo se insertan registros nuevos en la base de datos; los registros existentes no se actualizan. El método load solamente se puede utilizar en bases de datos DB2 locales y no remotas.

Método import:

    Con el método import para DB2 los datos también se cargan en la base de datos. Si los datos ya existen, no se suprimen sino que se actualizan con valores nuevos. Tenga en cuenta este método en cualquiera de las situaciones siguientes:

- El sistema de gestión de bases de datos es DB2
- No sabe si los datos están limpios o no
- Tiene que actualizar grandes conjuntos de datos homogéneos a nivel de columna
- Todas las tablas en las que se están importando datos tienen claves primarias

▶ **400** Con el método import, los datos también se cargan en la base de datos. Si los datos ya existen, no se suprimen sino que se actualizan con valores nuevos. Tenga en cuenta este método en cualquiera de las situaciones siguientes:

- No sabe si los datos están limpios o no
- Los datos ya existen en la base de datos
- Todas las tablas en las que se están importando datos tienen claves primarias

Existen las restricciones siguientes al utilizar el método de importación:

- Para utilizar el método de importación, el sistema de gestión de bases de datos debe ser DB2.
- ▶ **400** El método import no puede insertar ni actualizar datos en campos de datos de bits ni en campos DBCLOB y sólo se puede utilizar en bases de datos locales.
- Con el método import, Loader sólo inserta o actualiza tablas que tienen definidas claves primarias; el método import no puede insertar o actualizar datos en tablas que no tienen una clave primaria. Si el registro de entrada sólo tiene valores para columnas que son primarias, se rechaza el registro.

método import de SQL:

Con el método import de SQL, se utilizan sentencias de JDBC o SQL para actualizar o insertar datos en la base de datos. Se insertan datos si aún no existen y los datos existentes se actualizan. Tenga en cuenta este método en cualquiera de las situaciones siguientes:

- Está actualizando datos existentes y necesita actualizaciones a nivel de columna
- Algunos de los datos no están limpios
- La base de datos no es local

Nota: Si está utilizando la sincronización de espacios de búsqueda del Asesor de productos, deberá utilizar el método import de SQL para cargar datos.

Método de supresión:

El método de supresión se utiliza para suprimir datos que están en el documento XML de entrada de la base de datos. El elemento debe contener los valores para la clave primaria o el índice exclusivo para la tabla. Si los datos que se están suprimiendo tienen datos en otra tabla que son dependientes de ellos con la "supresión en cascada" habilitada, los datos dependientes también se suprimen.

El método createonly para sistemas AIX, Linux, Solaris y Windows:

Para mejorar el rendimiento durante la creación de la instancia, utilice el método createonly. Utilice el método createonly para crear archivos de datos de carga masiva (MLD) sin cargar los datos en la base de datos. Después puede utilizar su programa de utilidad de carga de base de datos nativo para cargar los archivos MLD que ha creado en una base de datos WebSphere Commerce ejecutando el mandato Load con el método loadonly.

El método loadonly para sistemas AIX, Linux, Solaris y Windows:

Utilice el método loadonly para cargar los archivos MLD que se han creado utilizando el método createonly. Utilice el método loadonly **solamente** para la creación de instancias. Si lo utiliza en cualquier otra ocasión, el resultado puede no ser el deseado.

Comparación de los métodos:

- **Comparación de los métodos de carga e importación de SQL**

El método import de SQL comprueba la coherencia de los datos, incluidas las referencias externas, y le permite actualizar los datos existentes. El método de carga no realiza dichas acciones.

- **Comparación de los métodos de importación y de importación de SQL**

Los métodos de importación y de importación de SQL realizan funciones similares. El método import es normalmente más rápido, pero requiere espacio de disco para los archivos temporales.

El método import sólo puede insertar o actualizar tablas que tienen definidas claves primarias, mientras que el método import de SQL no necesita que las tablas contengan claves primarias.

- **Comparación de métodos basándose en el producto de base de datos utilizado**

Los métodos de importación y carga utilizan programas de utilidad nativos que están optimizados para DB2, mientras que el método import de SQL utiliza llamadas JDBC (que son genéricas para muchos productos de base de datos).

Consideraciones acerca del rendimiento:

Cuando utilice Loader para cargar documentos grandes en una base de datos, tenga en cuenta los puntos siguientes:

- **Tamaño de almacenamiento dinámico de Java Virtual Machine (JVM)**

Por omisión, la cantidad máxima de memoria asignada al almacenamiento dinámico de JVM es de 64 MB. Si dicha cantidad no se incrementa, JVM puede quedarse finalmente sin memoria durante el proceso de carga. La cantidad máxima de memoria asignada al almacenamiento dinámico de Java puede modificarse utilizando la opción -mx de JVM en el mandato Java.

- **Anotaciones cronológicas de rastreo**

El anotador de rastreo puede agotar el almacenamiento dinámico de JVM al cargar un documento XML grande. La información de rastreo se utiliza principalmente para depurar una ejecución si ésta falla. Si no es necesario rastrear el proceso de carga, se deberá desactivar el rastreo. Hay un aumento significativo del rendimiento cuando el rastreo está desactivado. El rastreo se desactiva modificando el documento XML de configuración de anotación cronológica. Para obtener información sobre cómo modificar el documento XML de configuración de anotaciones cronológicas, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

- **Número de compromisos**

El número de compromisos por omisión para Loader cuando éste está funcionando en modalidad de importación de SQL es 1. Por consiguiente, las transacciones se comprometen por omisión para cada actualización o inserción en la base de datos. Para mejorar el rendimiento de Loader para documentos grandes, se deberá incrementar el número de compromisos. Después de tener en cuenta el tamaño del archivo `input.xml`, puede utilizar un número de compromisos mayor que el número de registros del archivo. Esto permite retrotraer el archivo `input.xml` completo en caso de que se produzca un error. El número de compromisos para Loader se modifica utilizando la opción -commitcount *número* para el mandato Load (donde *número* es el número de sentencias ejecutadas antes de que se comprometa la transacción).

- **Configuración de la anotación cronológica**

Se puede producir un proceso inusualmente lento al cargar datos como resultado de una de las situaciones siguientes:

- El usuario que invoca Loader no tiene permiso para grabar en el directorio o para actualizar el archivo especificado en el documento de configuración de anotación cronológica.
- El directorio especificado como ubicación del archivo en el documento de configuración de anotación cronológica no existe.
- La unidad especificada como ubicación del archivo en el documento de configuración de anotación cronológica no tiene suficiente espacio.

Cuando corrija cualquiera de estos problemas, puede que necesite cambiar la ubicación especificada del archivo modificando el documento de configuración de anotación cronológica (por omisión WCALoggerConfig.xml). Para obtener información sobre cómo modificar el documento XML de configuración de anotación cronológica, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

Si desea obtener información detallada sobre cómo configurar y personalizar los archivos utilizados para ejecutar este mandato, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

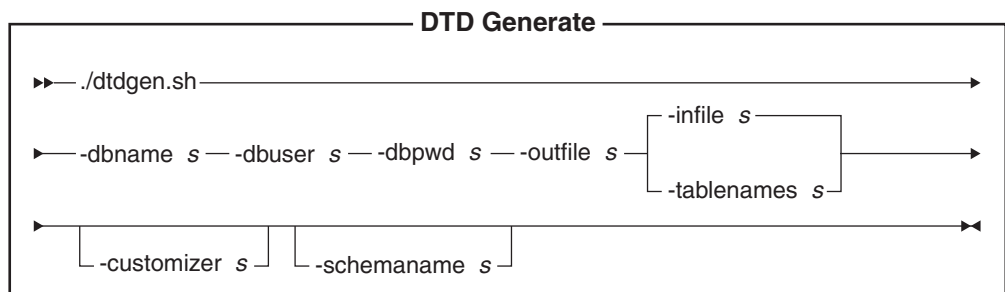
Mandatos de Loader package para transformar y extraer datos

Mandato DTD Generate

Este mandato crea una DTD para utilizarla con Loader package. Esta DTD se utilizará durante todo el proceso de carga de datos. En función de cómo invoque el mandato, DTD Generator puede generar una DTD sola o una DTD junto con un esquema XML.

DTD Generator puede crear una DTD basándose en el esquema de base de datos de WebSphere Commerce. Si utiliza las DTD proporcionadas con los archivadores de tienda de ejemplo y no modifica el esquema de base de datos, no necesitará generar una DTD utilizando DTD Generator. Las DTD que se proporcionan están situadas en el directorio *dir_instalación_WC/xml/sar*.

Se recomienda utilizar las DTD proporcionadas. Sin embargo, si personaliza un esquema de base de datos, deberá editar la DTD proporcionada para que coincida con los cambios o crear una DTD nueva.







Notas:





1. El diagrama anterior está pensado principalmente para servir de referencia para los parámetros de mandato.
2. Los nombres de archivo especificados como parámetros para este mandato pueden ir precedidos por vías de acceso relativas o absolutas.

Valores de los parámetros:

-dbname

   El nombre de la base de datos de destino
 Este es el nombre de la base de datos como se muestra en el directorio de base de datos relacional (WRKRDBDIRE).

-dbuser

   El nombre del usuario que se conecta a la base de datos.
 Generalmente, es el mismo que el nombre de usuario de la instancia.

-dbpwd

Contraseña del usuario que se conecta a la base de datos

-outfile

Nombre del archivo DTD de salida (preferiblemente con una extensión .dtd)





-infile Nombre de un archivo de entrada que contiene un nombre de tabla de base de datos en cada línea

-tablenames

Nombres de tablas separados por comas y escritos entre comillas ("")

-customizer





Nombre del archivo de propiedades de personalización que se va a utilizar. Este parámetro es opcional. El archivo de propiedades de personalización establece la forma de funcionamiento de DTD Generator. El archivo por omisión es:

   DB2ConnectionCustomizer.properties
 ISeries_GENWCSDTD_Customizer.properties

Si ha configurado la instancia para que utilice el controlador Toolbox, entonces utilice el archivo de personalización `Toolbox_GENWCSDTD_Customizer` que se proporciona para el controlador Toolbox. Asimismo debe especificar el nombre de sistema principal para el parámetro `-dbname`. El siguiente es un ejemplo de invocación del script `dtongen.sh`:

```
./dtongen.sh -dbname MI.NOMBSISPRAL.CA -dbuser instance -dbpwd micontraseña  
-outfile /path/out.dtd  
  method sqlimport -customizer Toolbox_GENWCSDTD_Customizer  
-infile /path/file.xml
```

Se puede especificar el archivo de propiedades de personalización como se muestra en el ejemplo siguiente:

  
`-customizer dir_instalación_WC/mi_directorio/nombre_archivo.properties`
`-customizer dir_instalación_WC/mi_directorio/nombre_archivo`

`-customizer dir_usuario_WC/mi_directorio/nombre_archivo.properties`
`-customizer dir_usuario_WC/mi_directorio/nombre_archivo`

Si este archivo existe en el directorio actual, se puede especificar el mismo archivo tal como se muestra en el ejemplo siguiente:

```
-customizer nombre_archivo.properties
```

Si este archivo se encuentra en el directorio especificado en la variable de entorno del sistema CLASSPATH, se puede especificar el mismo archivo, como se muestra en el ejemplo siguiente:

```
-customizer nombre_archivo
```

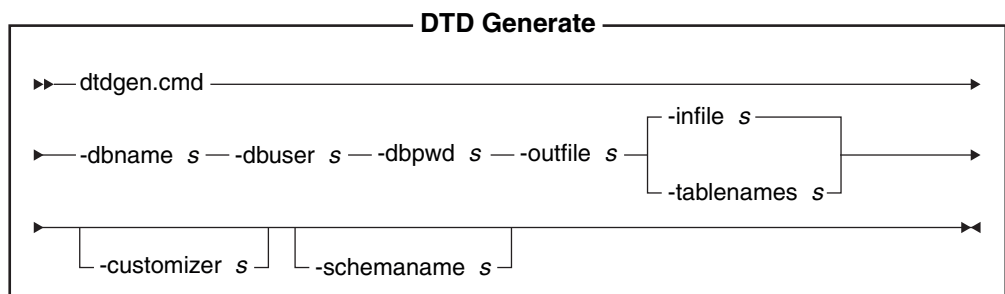
donde, *mi_directorio* es un directorio definido por el usuario y *nombre_archivo* es el nombre del archivo de propiedades que va a utilizar. Para obtener más información sobre la creación y especificación de un nuevo archivo de propiedades, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

-schemaname

Nombre del esquema de base de datos de destino. Este parámetro es opcional.

Si no se especifica este parámetro al ejecutar el mandato, éste busca una pareja nombre=valor en el archivo de propiedades de personalización que especifica el valor de SchemaName. Si esta pareja se encuentra en el archivo de propiedades, el mandato utiliza el valor especificado. Si no se especifica ni en el archivo de propiedades ni en la línea de mandatos, el mandato toma como valor por omisión el propietario del esquema de la tabla en la base de datos [400](#). Si no se especifica ni en el archivo de propiedades ni en la línea de mandatos, el mandato toma como valor por omisión el nombre del usuario de la base de datos.

Windows



Nota: Los nombres de archivo especificados como parámetros para este mandato pueden ir precedidos por vías de acceso relativas o absolutas.

Valores de los parámetros:

-dbname

Nombre de la base de datos de destino

-dbuser

Nombre del usuario que se conecta a la base de datos

-dbpwd

Contraseña del usuario que se conecta a la base de datos

-outfile

Nombre del archivo DTD de salida.

-infile Nombre de un archivo de entrada que contiene un nombre de tabla de base de datos en cada línea

-tablenames

Los nombres de tablas separados por comas.

-customizer

Nombre del archivo de propiedades de personalización que se va a utilizar. Este parámetro es opcional. El archivo de propiedades de personalización establece la forma de funcionamiento de DTD Generator.

DB2ConnectionCustomizer.properties es el archivo por omisión. Se puede especificar el archivo de propiedades de personalización como se muestra en el ejemplo siguiente:

```
-customizer dir_instalación_WC\mi_directorio\nombre_archivo.properties
```

```
-customizer dir_instalación_WC\mi_directorio\nombre_archivo
```

Si este archivo existe en el directorio actual, se puede especificar el mismo archivo tal como se muestra en el ejemplo siguiente:

```
-customizer  
nombre_archivo.properties
```

Si este archivo se encuentra en el directorio especificado en la variable de entorno del sistema CLASSPATH, se puede especificar el mismo archivo, como se muestra en el ejemplo siguiente:

```
-customizer nombre_archivo
```

donde, *mi_directorio* es un directorio definido por el usuario y *nombre_archivo* es el nombre del archivo de propiedades que va a utilizar. Para obtener más información sobre la creación y especificación de un nuevo archivo de propiedades, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

-schemaname

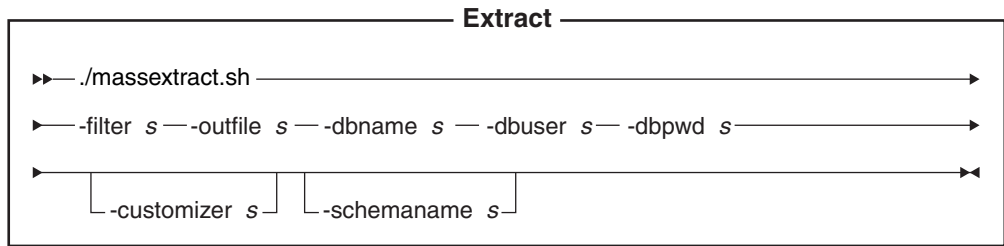
Nombre del esquema de base de datos de destino. Este parámetro es opcional. Si no se especifica este parámetro al ejecutar el mandato, éste busca una pareja nombre=valor en el archivo de propiedades de personalización que especifica el valor de SchemaName. Si esta pareja se encuentra en el archivo de propiedades, el mandato utiliza el valor especificado. Si no se especifica ni en el archivo de propiedades ni en la línea de mandatos, el mandato toma como valor por omisión el propietario del esquema de la tabla en la base de datos.

Si desea obtener información detallada sobre cómo configurar y personalizar los archivos utilizados para ejecutar este mandato, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

Mandato Extract

Este mandato extrae un subconjunto seleccionado de datos de una base de datos en forma de archivo XML.

Para extraer datos de una base de datos utilizando Extractor, debe especificar los datos que desea extraer utilizando un archivo de filtro de extracción. El filtro de extracción que utilice dependerá del tipo de datos que desee extraer.



Notas:

1. El diagrama anterior está pensado principalmente para servir de referencia para los parámetros de mandato.
2. Los nombres de archivo especificados como parámetros para este mandato pueden ir precedidos por vías de acceso relativas o absolutas.

Valores de los parámetros:

-filter Nombre del archivo de filtro de extracción

-outfile

Nombre del archivo XML de salida donde se almacenarán los datos extraídos

-dbname

▶ AIX ▶ Linux ▶ Solaris El nombre de la base de datos desde la que se están extrayendo los datos.

▶ 400 Este es el nombre de la base de datos como se muestra en el directorio de base de datos relacional (WRKRDBDIRE).

-dbuser

▶ AIX ▶ Linux ▶ Solaris El nombre del usuario de base de datos de la base de datos desde la que se están extrayendo los datos.

▶ 400 Generalmente, es el mismo que el nombre de usuario de la instancia.

-dbpwd

Contraseña asociada con el nombre de usuario para la base de datos de la que se están extrayendo datos

-customizer

Nombre del archivo de propiedades de personalización que se va a utilizar. El archivo de propiedades de personalización establece la forma de funcionamiento de Extractor. El archivo por omisión es:

▶ AIX ▶ Linux ▶ Solaris DB2ConnectionCustomizer.properties

▶ 400 ISeries_EXTWCSDTA_Customizer.properties. Si ha configurado la instancia para que utilice el controlador Toolbox, entonces utilice el archivo de personalización Toolbox_EXTWCSDTA_Customizer que se proporciona para el controlador Toolbox. Asimismo debe especificar el nombre de sistema principal para el parámetro -dbname. El siguiente es un ejemplo de cómo invocar el script massextract.sh:

```

./massextract.sh -dbname MI.NOMBSISPRAL.CA -dbuser instance -dbpwd mictra
-filter/path/filter.xml
-outfile /path/file.xml -customizer Toolbox_EXTWCSDTA_Customizer
  
```

Se puede especificar el archivo de propiedades de personalización como se muestra en el ejemplo siguiente:

▶ AIX
▶ Linux
▶ Solaris
 DB2ConnectionCustomizer.properties
 -customizer *dir_instalación_WC/mi_directorio/nombre_archivo.properties*
 -customizer *dir_instalación_WC/mi_directorio/nombre_archivo*
▶ 400

-customizer *dir_usuario_WC/mi_directorio/nombre_archivo.properties*
 -customizer *dir_usuario_WC/mi_directorio/nombre_archivo*

Si este archivo existe en el directorio actual, se puede especificar el mismo archivo tal como se muestra en el ejemplo siguiente:

-customizer
nombre_archivo.properties

Si este archivo se encuentra en el directorio especificado en la variable de entorno del sistema CLASSPATH, se puede especificar el mismo archivo, como se muestra en el ejemplo siguiente:

-customizer *nombre_archivo*

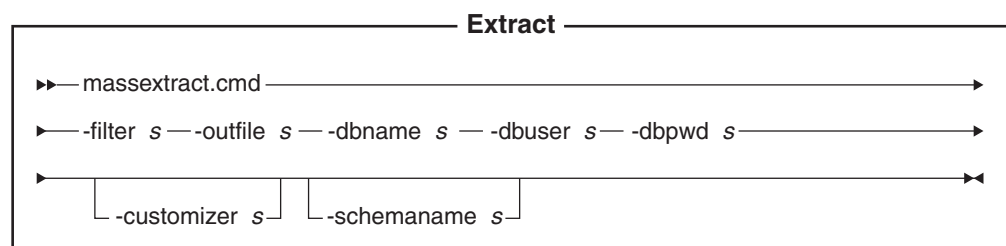
donde, *mi_directorio* es un directorio definido por el usuario y *nombre_archivo* es el nombre del archivo de propiedades que va a utilizar. Para obtener más información sobre la creación y especificación de un nuevo archivo de propiedades, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

-schemaname

Nombre del esquema de la base de datos de la que se están extrayendo datos. Este parámetro es opcional.

Si no se especifica este parámetro al ejecutar el mandato, éste busca una pareja nombre=valor en el archivo de propiedades de personalización que especifica el valor de SchemaName. Si esta pareja se encuentra en el archivo de propiedades, el mandato utiliza el valor especificado. Si no se especifica ni en el archivo de propiedades ni en la línea de mandatos, el mandato toma como valor por omisión el nombre de esquema de la tabla en la base de datos. ▶ 400 Si no se especifica el parámetro -schemaname ni en un archivo de propiedades ni en la línea de mandatos, el mandato toma por omisión el valor del parámetro -dbuser.

▶ Windows



Nota: Los nombres de archivo especificados como parámetros para este mandato pueden ir precedidos por vías de acceso relativas o absolutas.

Valores de los parámetros:

-filter Nombre del archivo de filtro de extracción

-outfile

Nombre del archivo XML de salida donde se almacenarán los datos extraídos

-dbname

Nombre de la base de datos de la que se están extrayendo datos

-dbuser

Nombre de usuario de la base de datos de la que se están extrayendo datos

-dbpwd

Contraseña asociada con el nombre de usuario para la base de datos de la que se están extrayendo datos

-customizer

Nombre del archivo de propiedades de personalización que se va a utilizar. El archivo de propiedades de personalización establece la forma de funcionamiento de Extractor. DB2ConnectionCustomizer.properties es el archivo por omisión. Se puede especificar el archivo de propiedades de personalización como se muestra en el ejemplo siguiente:

```
-customizer dir_instalación_WC\mi_directorio\nombre_archivo.properties
```

```
-customizer dir_instalación_WC\mi_directorio\nombre_archivo
```

Si este archivo existe en el directorio actual, se puede especificar el mismo archivo tal como se muestra en el ejemplo siguiente:

```
-customizer  
nombre_archivo.properties
```

Si este archivo se encuentra en el directorio especificado en la variable de entorno del sistema CLASSPATH, se puede especificar el mismo archivo, como se muestra en el ejemplo siguiente:

```
-customizer nombre_archivo
```

donde, *mi_directorio* es un directorio definido por el usuario y *nombre_archivo* es el nombre del archivo de propiedades que va a utilizar. Para obtener más información sobre la creación y especificación de un nuevo archivo de propiedades, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

-schemaname

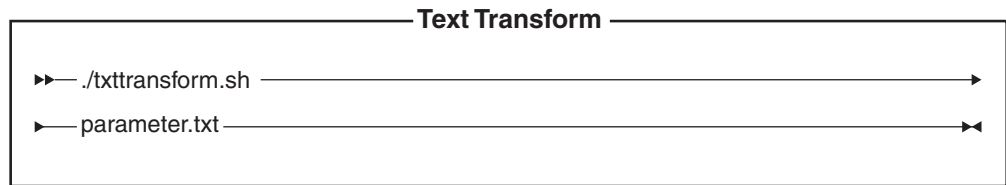
Nombre del esquema de la base de datos de la que se están extrayendo datos. Este parámetro es opcional.

Si no se especifica este parámetro al ejecutar el mandato, éste busca una pareja nombre=valor en el archivo de propiedades de personalización que especifica el valor de SchemaName. Si esta pareja se encuentra en el archivo de propiedades, el mandato utiliza el valor especificado. Si no se especifica ni en el archivo de propiedades ni en la línea de mandatos, el mandato toma como valor por omisión el nombre de esquema de la tabla en la base de datos.

Para obtener más información sobre la utilización de este mandato, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

Mandato Text Transform

Este mandato transforma los datos entre un formato variable delimitado por caracteres y un formato XML.



Nota: El diagrama anterior está pensado principalmente para servir de referencia para los parámetros de mandato.

Valores de los parámetros:

Los valores siguientes se especifican y se separan mediante comas en un archivo de parámetros (*parameter.txt*):

archivo de entrada

Nombre del archivo que se debe transformar

archivo de esquema

Nombre del archivo de esquema XML que se debe utilizar en la transformación

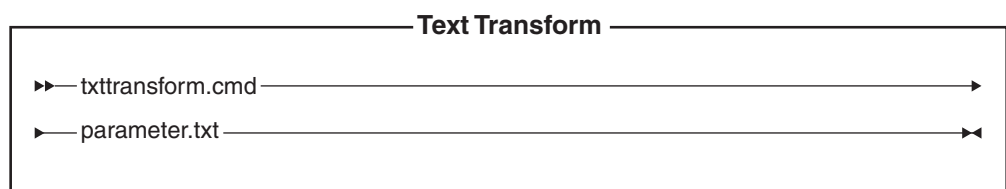
archivo de salida

Nombre del archivo de salida en el que se almacenarán los datos transformados

método de transformación

Método a utilizar al añadir los datos en el archivo de salida. Especifique **Create** si se debe crear un archivo nuevo o especifique **Append** si los datos de salida deben añadirse a un archivo de datos existente.

Este archivo también se conoce como archivo de "manifiesto" o de "mandatos". Puede contener varias líneas de cuatro parámetros cada una.



Valores de los parámetros:

Los valores siguientes se especifican y se separan mediante comas en un archivo de parámetros (*parameter.txt*):

archivo de entrada

Nombre del archivo que se debe transformar

archivo de esquema

Nombre del archivo de esquema XML que se debe utilizar en la transformación

archivo de salida

Nombre del archivo de salida en el que se almacenarán los datos transformados

método de transformación

Método a utilizar al añadir los datos en el archivo de salida. Especifique **Create** si se debe crear un archivo nuevo o especifique **Append** si los datos de salida deben añadirse a un archivo de datos existente.

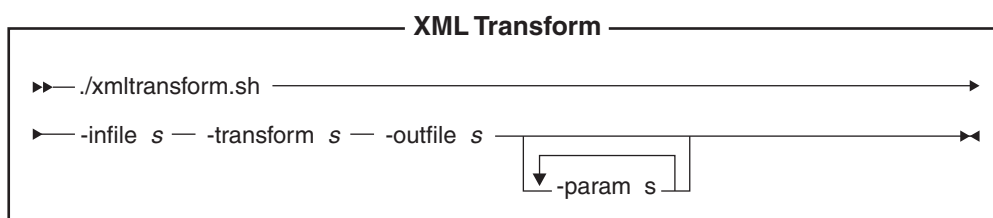
Nota: Este archivo también se conoce como archivo de "manifiesto" o de "mandatos". Puede contener varias líneas de cuatro parámetros cada una.

Para obtener más información sobre la utilización de este mandato, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

Mandato XML Transform

Este mandato convierte un archivo XML en un formato XML alternativo.

▶ AIX ▶ Linux ▶ 400 ▶ Solaris



Notas:

1. El diagrama anterior está pensado principalmente para servir de referencia para los parámetros de mandato.
2. Los nombres de archivo especificados como parámetros para este mandato pueden ir precedidos por vías de acceso relativas o absolutas.

Valores de los parámetros:

-infile Nombre del archivo que se debe transformar

-transform

Nombre del archivo de normas de correlación de XSL de transformación

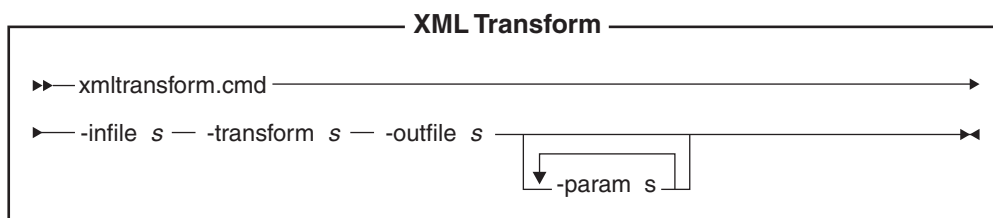
-outfile

Nombre del archivo XML de salida en el que se almacenarán los datos transformados

-param

Parámetro que se debe pasar al archivo de normas de correlación de XSL. *Este parámetro es opcional.* Este parámetro puede especificarse varias veces para pasar varias parejas nombre=valor.

▶ Windows



Nota: Los nombres de archivo especificados como parámetros para este mandato pueden ir precedidos por vías de acceso relativas o absolutas.

Valores de los parámetros:

-infile Nombre del archivo que se debe transformar

-transform

Nombre del archivo de normas de correlación de XSL de transformación

-outfile

Nombre del archivo XML de salida en el que se almacenarán los datos transformados

-param

Parámetro que se debe pasar al archivo de normas de correlación de XSL. *Este parámetro es opcional.* Este parámetro puede especificarse varias veces para pasar varias parejas nombre=valor.

Para obtener más información sobre la utilización de este mandato, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

Herramientas relacionadas con los mandatos de Loader package

Herramienta de Transformación de texto

La herramienta de Transformación de texto le ayuda a procesar una transformación de datos entre un formato variable delimitado por caracteres y un formato XML, utilizando el mandato Text Transform. Se proporcionan las vistas siguientes:

1. La Vista de edición de esquema de texto le permite crear y modificar el archivo de esquema XML que se va a utilizar en una transformación.
2. La Vista de edición de mandato de transformación le permite crear y modificar los mandatos reales utilizados para ejecutar el proceso de transformación.
3. La Vista de proceso de mandato de transformación le permite iniciar el proceso de transformación.

Para obtener más información sobre esta herramienta, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

Editor XSL

XML Transformer utiliza XSL para definir las normas para transformar un archivo XML en otro archivo XML. La función de correlación en el editor XSL le proporciona una interfaz visual con la que puede establecer la asociación de un elemento de una DTD de origen a un elemento de una DTD de destino. Con dos DTD específicas, puede desarrollar normas XSL que determinen cómo un archivo XML que se ajusta a la primera DTD (origen) se transforma en un archivo que se ajusta a la segunda DTD (destino).

Para obtener más información sobre esta herramienta, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

Carga de los datos de la tienda

Esta sección proporciona ejemplos de cómo cargar datos de la tienda en la base de datos de WebSphere Commerce Server utilizando los programas de utilidad de línea de mandatos de Loader package.

Notas:






1. Los ejemplos de esta sección se llevan a cabo en un entorno Windows. Para obtener información sobre cómo ejecutar estos mandatos en otros entornos, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.
2. Aunque los programas de utilidad de línea de mandatos de Loader package soportan bases de datos DB2, DB2 para iSeries y Oracle, en los ejemplos siguientes sólo se incluyen los mandatos y las opciones para DB2. Si está utilizando una base de datos distinta de DB2, asegúrese de modificar los archivos de propiedades de personalización tal como se describe en la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

Consulte el Capítulo 38, “Carga de grupos de elementos de base de datos de WebSphere Commerce”, en la página 405 y el apartado “Carga de grupos de elementos de base de datos” en la página 413 si desea obtener información sobre el proceso de carga para grupos de elementos de base de datos de WebSphere Commerce.


Utilización de los mandatos y scripts de Loader package

Para ejecutar los mandatos de Loader, utilice los scripts o mandatos que se proporcionan en el directorio *dir_instalación_WC/bin* de WebSphere Commerce.

Los scripts y mandatos son los siguientes:

-    
 - dtdgen.sh** Script de shell DTD Generate
 - idresgen.sh** Script de shell ID Resolve
 - massload.sh** Script de shell Load
 - massextract.sh** Script de shell Extract
 - txtransform.sh** Script de shell Text Transform
 - xmltransform.sh** Script de shell XML Transform
- 
 - dtdgen.cmd** Mandato DTD Generate
 - idresgen.cmd** Mandato ID Resolve
 - massload.cmd** Mandato Load
 - massextract.cmd** Mandato Extract
 - txtransform.cmd** Mandato Text Transform
 - xmltransform.cmd** Mandato XML Transform

Ejemplos de resolución de identificadores

Los ejemplos de resolución de identificadores descritos en esta sección utilizan los archivos de elementos de la tienda de ejemplo ToolTech de  .

Puesto que este ejemplo se basa en la carga de nuevos datos en la base de datos de WebSphere Commerce, utilizaremos el método load.

Si posteriormente necesita modificar determinados elementos en el documento XML, puede hacerlo utilizando el método update. El método update se ejecutará más rápidamente que el método load porque no se asignan identificadores nuevos con el método update. Con el método update, se realiza una consulta de base de datos para localizar el identificador y se informa de un error si no se encuentra el identificador. Consulte la sección que empieza en la página 371 para obtener más información sobre el funcionamiento de este proceso.

Si el archivo XML de entrada contiene elementos que ya existen en la base de datos así como elementos que no existen, utilice el método combinado. Con el método combinado, primero se realiza una búsqueda de la base de datos y se asigna un identificador al elemento si no se encuentra el registro. Cuando tenga dudas, utilice el método combinado. Aunque los métodos de carga y actualización proporcionan un rendimiento más rápido que el método combinado, el archivo XML resuelto producido por la utilización del método combinado tiene más probabilidades de cargarse sin errores.

Para obtener más información sobre el funcionamiento de ID Resolver, consulte el apartado “Métodos para el mandato ID Resolve” en la página 372.

Resolución de identificadores en archivos XML con alias internos

Para resolver identificadores utilizando los alias internos antes de cargar los datos en la base de datos de WebSphere Commerce Server, ejecute el mandato ID Resolve tal como se muestra en el ejemplo siguiente.

Nota: Este ejemplo presupone que WebSphere Commerce se ha instalado en una máquina Windows. Si tiene instalado WebSphere Commerce en un sistema operativo diferente, sustituya los valores por los correspondientes a su sistema operativo.

1. Cree un directorio de trabajo.

Para este ejemplo, cree el directorio *dir_instalación_WC/test*.

Nota: Si no utiliza *dir_instalación_WC/test* como directorio de trabajo, escriba el nombre y la vía de acceso del directorio de trabajo que utiliza en lugar de *dir_instalación_WC/test* en el resto de los ejemplos que se muestran en este capítulo.

2. Asegúrese de que el archivo XML de entrada así como cualquier archivo DTD al que se haga referencia estén en una ubicación donde ID Resolver pueda encontrarlos.

Para este ejemplo, realice lo siguiente:

- a. En el indicador de mandatos de Windows, entre el mandato siguiente:

```
copy dir_instalación_WC\samplstores\  
B2BDirect\B2BDirect.sar dir_instalación_WC\test
```

Esto copia el archivo B2BDirect.sar en *dir_instalación_WC\test*.

- b. En un indicador de mandatos de Windows, entre el mandato siguiente:

```
cd to dir_instalación_WC\test
```

- c. Efectúe una de las acciones siguientes:

- Si tiene instalado Java, escriba el mandato siguiente en un indicador de mandatos de Windows:

```
jar -xvf B2BDirect.sar
```

Esto extrae los archivos XML de la tienda de ejemplo Directo a la empresa, ToolTech, a la unidad:\WebSphere\CommerceServer55\test.

- Utilice cualquier producto de descompresión actualizado (por ejemplo, WinZip o PKZIP) para extraer el contenido completo de *dir_instalación_WC\samplstores\B2BDirect\B2BDirect.sar* en *dir_instalación_WC\test*.

Esto extrae los archivos XML de la tienda de ejemplo ToolTech de

```
> Business en dir_instalación_WC\test\WEB-
INF\stores\BusinessDirect\data\ToolTech\data.
```

- d. En el indicador de mandatos de Windows, entre el mandato siguiente:

```
copy dir_instalación_WC\xml\sar\store.dtd
dir_instalación_WC\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data
```

Esto copia el archivo store.dtd en *dir_instalación_WC\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data*.

- e. En el indicador de mandatos de Windows, entre el mandato siguiente:

```
copy dir_instalación_WC\xml\sar\DBLoadMacros.dtd
dir_instalación_WC\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data
```

Esto copia el archivo DBLoadMacros.dtd en *dir_instalación_WC\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data*.

- f. En el indicador de mandatos de Windows, entre el mandato siguiente:

```
copy dir_instalación_WC\xml\sar\fulfillment.dtd
dir_instalación_WC\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data
```

Esto copia el archivo fulfillment.dtd en *dir_instalación_WC\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data*.

3. Asegúrese de que el esquema de WebSphere Commerce se carga en la base de datos junto con los datos de rutina de carga necesarios, creando una instancia de base de datos adecuada de WebSphere Commerce Server.

Nota: Para obtener más información sobre cómo crear una instancia, consulte la publicación *WebSphere Commerce, Guía de instalación* correspondiente a su sistema operativo.

La instancia de base de datos de WebSphere Commerce Server que utiliza este ejemplo se denomina *mall*. Las claves primarias y externas se obtendrán de las tablas KEYS y SUBKEYS de esta base de datos; por consiguiente, ID Resolver no podrá resolver los identificadores si la base de datos no se carga correctamente.

4. Para resolver los identificadores del archivo fulfillment.xml, haga lo siguiente:

- a. Edite el archivo fulfillment.xml para que incluya lo siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fulfillment-asset SYSTEM "fulfillment.dtd">
<fulfillment-asset>
</fulfillment-asset>
```

El archivo fulfillment.xml debe ser similar al siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE fulfillment-asset SYSTEM "fulfillment.dtd">
<fulfillment-asset>
<!--defaultshipoffset can be overridden in the STORITMFFC table.-->
<!--Now in ToolTech STORITMFFC.shippingoffset is set to 86400
seconds which is one day-->
<ffmcenter
ffmcenter_id="@ffmcenter_id_1"
member_id="@MEMBER_ID;"
name="ToolTech Home"
defaultshipoffset="0"
markfordelete="0"
/>
</fulfillment-asset>
```

- b. Edite el archivo DBLoadMacros.dtd para que incluya lo siguiente si no se ha incluido:

```
<!ENTITY MEMBER_ID "-2001">
```

- c. Entre el mandato siguiente para ejecutar ID Resolver en el archivo fulfillment.xml (donde está definida la tabla FFMCENTER):

```
idresgen -dbname mall -dbuser db2admin -dbpwd db2admin
-infile dir_instalación_WC\test\WEB-INF\stores\BusinessDirect\data
\ToolTech\data\fulfillment.xml
-outfile dir_instalación_WC\test\WEB-INF\stores\BusinessDirect\data
\ToolTech\data\fulfillment1.xml
-method load
```

donde

- *mall* debe cambiarse por el nombre de la base de datos de destino si no está utilizando mall
- El primer *db2admin* debe cambiarse por el nombre del usuario que se conecta a la base de datos si no está utilizando db2admin
- El segundo *db2admin* debe cambiarse por la contraseña del usuario que se conecta a la base de datos si no está utilizando db2admin

El elemento resuelto del archivo de salida fulfillment1.xml tiene este aspecto:

```
<fulfillment-asset>
<ffmcenter
  FFMCENTER_ID="10001"
  MEMBER_ID="-2001"
  NAME="ToolTech Home"
  DEFAULTSHIPOFFSET="0"
  MARKFORDELETE="0"
/>
</fulfillment-asset>
```

Nota: Esto es un ejemplo. El archivo de salida puede contener valores diferentes.

5. Para resolver los identificadores del archivo store.xml, haga lo siguiente:

- a. Edite el archivo store.xml para que incluya lo siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE store-asset SYSTEM "store.dtd">
<store-asset>
</store-asset>
```

- b. Obtenga la clave FFMCENTER_ID del archivo de salida resultante (fulfillment1.xml) y sustituya dicha clave para todas las ocurrencias de @ffmcenter_id_1 en la copia de trabajo de store.xml en

```
dir_instalación_WC\test\WEB-
INF\stores\BusinessDirect\data\ToolTech\data
```

- c. Edite el archivo DBLoadMacros.dtd para que incluya lo siguiente si no se ha incluido:

```
<!ENTITY MEMBER_ID "-2001">
<!ENTITY STORE_IDENTIFIER "ToolTech">
<!ENTITY STORE_DIR "ToolTech">
```

- d. Entre el mandato siguiente:

```
idresgen -dbname mall -dbuser db2admin -dbpwd db2admin
-infile dir_instalación_WC\test\WEB-INF\stores\
BusinessDirect\data\ToolTech\data\store.xml-outfile
dir_instalación_WC\test\WEB-INF\stores\
BusinessDirect\data\ToolTech\data\store1.xml -method load
```

donde

- *mall* debe cambiarse por el nombre de la base de datos de destino si no está utilizando mall
- el primer *db2admin* debe cambiarse por el nombre del usuario que se conecta a la base de datos si no está utilizando db2admin
- el segundo *db2admin* debe cambiarse por la contraseña del usuario que se conecta a la base de datos si no está utilizando db2admin

Los elementos totalmente resueltos del archivo de salida store1.xml tienen este aspecto:

```
<store-asset>
<storeent
  STOREENT_ID="10151"
  MEMBER_ID="-2001"
  TYPE="S"
  IDENTIFIER="ToolTech"
  SETCCURR="USD"
/>
<store
  STORE_ID="10151"
  DIRECTORY="ToolTech"
  FFMCTR_ID="10001"
  LANGUAGE_ID="-1"
  STOREGRP_ID="-1"
  ALLOCATIONGOODFOR="43200"
  BOPMPADFACTOR="0"
  DEFAULTBOFFSET="259200"
  FFMCTR_SELECTIONFLAGS="0"
  MAXBOFFSET="7776000"
  REJECTEDORDEXPIRY="259200"
  RTNFFMCTR_ID="10001"
  PRICEREFFLAGS="0"
  STORETYPE="B2B"
/>
<vendor
  VENDOR_ID="10001"
  STOREENT_ID="10151"
  VENDORNAME="Tooltech Vendor"
  MARKFORDELETE="0"
/>
<dispentrel
  AUCTIONSTATE="0"
  CATENTRY_ID="0"
  CATENTTYPE_ID="ProductBean"
  DEVICEFMT_ID="-1"
  DISPENTREL_ID="10001"
  MBRGRP_ID="0"
  PAGENAME="CatalogProductDisplay.jsp"
  STOREENT_ID="10151"
  RANK="0"
/>
<dispentrel
  AUCTIONSTATE="0"
  CATENTRY_ID="0"
  CATENTTYPE_ID="ItemBean"
  DEVICEFMT_ID="-1"
  DISPENTREL_ID="10002"
  MBRGRP_ID="0"
  PAGENAME="CatalogItemDisplay.jsp"
  STOREENT_ID="10151"
  RANK="0"
/>
<dispcgprel
  CATGROUP_ID="0"
```

```

        DEVICEFMT_ID="-1"
        DISPCGPREL_ID="10001"
        MBRGRP_ID="0"
        PAGENAME="CatalogCategories.jsp"
        STOREENT_ID="10151"
        RANK="0"
    />
<invadjcode
    ADJUSTCODE="PCNT"
    INVADJCODE_ID="10001"
    MARKFORDELETE="0"
    STOREENT_ID="10151"
/>
<invadjcode
    ADJUSTCODE="SPLG"
    INVADJCODE_ID="10002"
    MARKFORDELETE="0"
    STOREENT_ID="10151"
/>
<invadjcode
    ADJUSTCODE="DISC"
    INVADJCODE_ID="10003"
    MARKFORDELETE="0"
    STOREENT_ID="10151"
/>
<rtnreason
    REASONTYPE="C"
    RTNREASON_ID="10001"
    STOREENT_ID="10151"
    MARKFORDELETE="0"
    CODE="WPR"
/>
<rtnreason
    REASONTYPE="B"
    RTNREASON_ID="10002"
    STOREENT_ID="10151"
    MARKFORDELETE="0"
    CODE="DEF"
/>
<rtnreason
    REASONTYPE="M"
    RTNREASON_ID="10003"
    STOREENT_ID="10151"
    MARKFORDELETE="0"
    CODE="ERR"
/>
<rtnreason
    REASONTYPE="M"
    RTNREASON_ID="10004"
    STOREENT_ID="10151"
    MARKFORDELETE="0"
    CODE="WPS"
/>
</store-asset>

```

Nota: Esto es un ejemplo. El archivo de salida puede contener valores diferentes.

6. En el archivo store.xml, encontrará el elemento siguiente:

```

<storeent
    STOREENT_ID="@storeent_id_1"
    MEMBER_ID="@seller_b2b_mbr_id"
    TYPE="S"
    IDENTIFIER="&STORE_IDENTIFER"
    SETCURR="USD"
/>

```

Este elemento de `store.xml` se correlaciona con la tabla `storeent` de la base de datos y sus atributos `STOREENT_ID`, `MEMBER_ID`, `TYPE`, `IDENTIFIER` y `SETCURR` se correlacionan con columnas de esa tabla. La especificación `@storeent_id_1` es un alias interno para el valor del atributo `STOREENT_ID` y `&MEMBER_ID`; es un parámetro de entidad. El valor de la entidad `&MEMBER_ID`; tiene que sustituirse para poderse cargar utilizando Loader. El valor de `&MEMBER_ID`; se define en el archivo de macros `DBLoadMacros.dtd` y el valor por el que se sustituye proviene de ese archivo. Cuando ID Resolver encuentra `@storeent_id_1`, examina la antememoria de tablas primarias para ver si existe `storeent`. `Storeent` existe, puesto que es una tabla primaria. ID Resolver busca el contador para dicha tabla, lo incrementa y sustituye el alias interno por el resultado. Todas las demás entradas de este tipo del archivo `store.xml` se procesan del mismo modo.

7. Asegúrese de que la vía de acceso incluya el directorio que contiene el script o mandato ID Resolve apropiado que se lista en el apartado "Utilización de los mandatos y scripts de Loader package" en la página 394.

Para este ejemplo, entre el mandato siguiente en una línea de mandatos de Windows:

```
cd dir_instalación_WC\bin
```

donde `dir_instalación_WC\bin` debe cambiarse por el nombre del directorio que contiene el mandato ID Resolve `idresgen.cmd` si no está situado en el directorio `dir_instalación_WC\bin` de su sistema.

8. En el indicador de mandatos de Windows, entre el mandato siguiente:

```
idresgen -dbname mall -dbuser wcs -dbpwd wcs1 -infile
dir_instalación_WC\test\WEB-INF\stores\BusinessDirect\data\ToolTech\
data\store.xml -outfile
dir_instalación_WC\test\WEB-INF\stores\BusinessDirect\data\ToolTech\
data\store1.xml-method load
```

donde

- `mall` debe cambiarse por el nombre de la base de datos de destino si no está utilizando `mall`
- el primer `db2admin` debe cambiarse por el nombre del usuario que se conecta a la base de datos si no está utilizando `db2admin`
- el segundo `db2admin` debe cambiarse por la contraseña del usuario que se conecta a la base de datos si no está utilizando `db2admin`

El primer fragmento XML de salida de `store1.xml` tiene este aspecto:

```
<storeent
  STOREENT_ID="10001"
  MEMBER_ID="-2001"
  TYPE="S"
  IDENTIFIER="ToolTech"
  SETCURR="USD"
/>
```

Nota: Esto es un ejemplo. El archivo de salida puede contener valores diferentes.

El segundo fragmento XML de `store1.xml` tiene este aspecto:

```
<store
  STORE_ID="10001"
  DIRECTORY="ToolTech"
  FFMCENTER_ID=""
  LANGUAGE_ID="-1"
  STOREGRP_ID="-1"
```

```

ALLOCATIONGOODFOR="43200"
BOPMPADFACTOR="0"
DEFAULTBOFFSET="2592000"
FFMCSELECTIONFLAGS="0"
MAXBOFFSET="7776000"
REJECTEDORDEXPIRY="259200"
RTNFFMCTR_ID=""
PRICEREFFLAGS="0"
STORETYPE="B2B"
/>

```

Nota: Esto es un ejemplo. El archivo de salida puede contener valores diferentes.

Puede resolver los identificadores utilizando una de las opciones siguientes:

- Opción 1:
 - a. Fusione los archivos `fulfillment.xml` y `store.xml` añadiendo el contenido que sea exclusivo de `fulfillment.xml` (incluida la referencia a `fulfillment.dtd`) en `store.xml`, asegurándose de que el elemento `ffmcenter` que se muestra más abajo precede al elemento `store`.


```

<ffmcenter
  FFMCENTER_ID="@ffmcenter_id_1"
  MEMBER_ID="&MEMBER_ID;"
  NAME="ToolTech Home"
  DEFAULTBOFFSET="0"
  MARKFORDELETE="0"
/>

```
 - b. Ejecute ID Resolver en el archivo fusionado.
- Opción 2: Cargue el grupo de datos de elementos de tienda utilizando el proceso que se describe en el apartado “Carga de grupos de elementos de base de datos” en la página 413.

Especificación de un archivo de propiedades con ID Resolver

Puede modificar el modo en que ID Resolver resuelve los identificadores utilizando el parámetro `-propfile`. El archivo de propiedades por omisión es `IdResolveKeys.properties`. Para modificar y utilizar `IdResolveKeys.properties`, copie este archivo en un directorio definido por el usuario, efectúe los cambios necesarios y luego especifique el nuevo archivo cuando invoque el mandato ID Resolve. El archivo por omisión `IdResolveKeys.properties` está situado en `dir_instalación_WC/properties`:

La especificación del archivo de propiedades tiene prioridad respecto al uso de alias internos.

El siguiente es un fragmento XML de ejemplo del archivo `store.xml`:

```

<store
  STORE_ID="@storeent_id_1"
  DIRECTORY="ToolTech"
  FFMCENTER_ID="@ffmcenter_id_1"
  LANGUAGE_ID="&es_ES;"
  STOREGRP_ID="-1"
  ALLOCATIONGOODFOR="43200"
  BOPMPADFACTOR="0"
  DEFAULTBOFFSET="2592000"
  FFMSELECTIONFLAGS="0"
  MAXBOFFSET="7776000"
  REJECTEDORDEXPIRY="259200"

```

```

RTNFFMCTR_ID="@ffmcenter_id_1"
PRICEREFFLAGS="0"
STORETYPE="B2B"
/>

```

Si ejecuta ID Resolver con `-profile` especificado como `dir_instalación_WC\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data\myPropFile` y el archivo especificado, `myPropFile.properties`, contiene las entradas siguientes:

```

NAMEDELIMITER=@
SELECTDELIMITER=:
FFMCENTER=@FFMCENTER_ID@MEMBER_ID:10051 -2001

```

ID Resolver consulta en la base de datos la tabla FFMCENTER con una cláusula `where` de 10051 y -2001 cuando se procesa el elemento de tienda. Entonces se utiliza el índice que se devuelve para este valor para resolver el identificador para FFMCENTER_ID.

Para obtener más información sobre la utilización de este mandato, consulte el apartado “Mandato ID Resolve” en la página 365.

Ejemplo de carga de datos

Una vez que haya resuelto los identificadores del archivo XML, si son necesarios, estará listo para cargar los datos en la base de datos de WebSphere Commerce Server.

Nota: Si ha resuelto correctamente los identificadores en sus datos XML, el archivo XML fuente no debería contener ninguno de estos elementos:

- palabras precedidas por el símbolo arroba (@)
- palabras precedidas por el símbolo y (&)
- identificadores con comillas vacías (")

La presencia de alguno de ellos indica que el archivo XML no está preparado para cargarse.

El ejemplo de carga de datos descrito en esta sección utiliza el archivo `fulfillment1.xml` que se ha resuelto en el apartado “Ejemplos de resolución de identificadores” en la página 394.

Para cargar datos en la base de datos de WebSphere Commerce Server, ejecute el mandato `Load` tal como se muestra en el ejemplo siguiente:

1. Cree un directorio de trabajo.

Para este ejemplo, utilice el directorio `dir_instalación_WC\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data\` que ha creado en el apartado “Ejemplos de resolución de identificadores” en la página 394.

2. Asegúrese de que el archivo XML de entrada está en una ubicación donde `Loader` puede encontrarlo.

Para este ejemplo, asegúrese de que el archivo de salida `fulfillment1.xml` que ha creado en el apartado “Ejemplos de resolución de identificadores” en la página 394 esté en `dir_instalación_WC\test\WEB-INF\stores\BusinessDirect\data\ToolTech\data\`.

3. Haga una copia de seguridad de su base de datos de WebSphere Commerce Server de forma que pueda restaurar la base de datos si se produce un error irreparable.

Nota: Consulte la documentación sobre copia de seguridad y recuperación que se proporciona con el producto para obtener información acerca de cómo hacer una copia de seguridad para la base de datos.

4. Asegúrese de que la vía de acceso incluye el directorio que contiene el script o el mandato Load apropiado tal como se lista en el apartado “Utilización de los mandatos y scripts de Loader package” en la página 394.

Para este ejemplo, entre el mandato siguiente en una línea de mandatos de Windows:

```
cd dir_instalación_WC\bin
```

donde *dir_instalación_WC\bin* debe cambiarse por el nombre del directorio que contiene el mandato Load *massload.cmd* si no está situado en *dir_instalación_WC\bin* en su sistema.

5. Ejecute el mandato Load en el archivo XML resuelto para cargar los datos en la base de datos de destino.

Para este ejemplo, entre el mandato siguiente en una línea de mandatos de Windows:

```
massload -dbname mall -dbuser db2admin -dbpwd db2admin -infile  
dir_instalación_WC\test\WEB-INF\stores\BusinessDirect\data\ToolTech  
\data\fulfillment1.xml -method sqlimport -commitcount 50
```

donde

- *mall* debe cambiarse por el nombre de la base de datos de destino si no está utilizando *mall*
- el primer *db2admin* debe cambiarse por el nombre del usuario que se conecta a la base de datos si no está utilizando *db2admin*
- el segundo *db2admin* debe cambiarse por la contraseña del usuario que se conecta a la base de datos si no está utilizando *db2admin*


Aunque haya menos de 50 elementos a cargar, este ejemplo especifica un valor de 50 para *-commitcount*. Es por motivos de rendimiento. Por omisión, el número de compromisos es 1. Si se utiliza este valor por omisión, se graba en la base de datos una operación de compromiso para cada registro. Si establece el número en 50 en el ejemplo anterior, se asegurará de que la E/S de la base de datos sólo se produzca una vez si la carga es satisfactoria y de que no se grabe nada en la base de datos si se producen errores. Sin embargo, si tiene que cargar una gran cantidad de datos, se recomienda que no establezca el número de compromisos en un valor tan grande como el del número de elementos por las razones siguientes:

- Un valor de número de compromisos alto produce un alto consumo de memoria.
- Cuando el valor del número de compromisos es más pequeño que el número de elementos, al menos se han grabado algunos datos en la base de datos. En función del valor de *-maxerror*, un valor más pequeño para *-commitcount* asegura que algunos datos se graben en la base de datos antes de que se exceda el número máximo de errores y que termine la herramienta. El valor por omisión para *-maxerror* es 1.

El valor por omisión de la opción *-noprimaries* es error para que la herramienta informe de los errores y termine cuando falten claves primarias.

Dado que estos ejemplos no cargan los elementos de tienda en el orden que utiliza la Consola de administración y que se describe en el apartado “Secuencia de carga de elementos de base de datos” en la página 406, es posible que el archivo *store1.xml* creado en el apartado “Ejemplos de resolución de identificadores” en la

página 394 viole las restricciones de integridad de algunas tablas. Si ha intentado cargar `store1.xml` sin modificar utilizando el método `load`, una violación de restricción hará que la base de datos entre en estado de pendiente. Por consiguiente, para una mayor sencillez, este ejemplo de utilización del mandato `Load` se basa en la versión resuelta del archivo `fulfillment.xml`, cuya única clave externa es la del `MEMBER_ID` definido en la tienda de ejemplo. Este ejemplo carga el archivo `fulfillment1.xml` resuelto que se ha generado en el apartado “Ejemplos de resolución de identificadores” en la página 394 y utiliza el método `SQL`. Cuando no esté seguro de si el contenido del archivo XML está limpio, utilice el método `import` de `SQL` tal como se muestra en este ejemplo, con los parámetros `-commitcount` y `-maxerror` establecidos en los valores adecuados, de forma que se informe de cualquier violación de restricción de base de datos sin alterar la base de datos ni poner en peligro la integridad de la misma.

Cuando ejecuta este mandato, se crea por omisión un archivo de texto de rastreo (`trace.txt`) en el subdirectorio de ejecución (`dir_instalación_WC\bin` en el ejemplo anterior). En , se crea por omisión el archivo `trace.txt` en `QIBM/Userdata/CommerceServer55/instances/nombre_instancia/logs`.

Si el archivo de configuración de anotaciones cronológicas `WCLoggerConfig.xml` se ha modificado para colocar `trace.txt` en una ubicación diferente, vaya a dicha ubicación y analice el archivo. Para obtener más información acerca de cómo personalizar `WCLoggerConfig.xml`, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

El archivo `trace.txt` contiene una lista de las acciones que realiza el mandato y de sus resultados. Si utiliza el método `SQL import` con el mandato como se muestra en el ejemplo anterior, al final de `trace.txt` encontrará una entrada que le indicará el número de registros comprometidos.

Para obtener más información sobre la utilización de este mandato, consulte el apartado “Mandato `Load`” en la página 374.

Capítulo 38. Carga de grupos de elementos de base de datos de WebSphere Commerce

Si no desea crear todos los elementos de base de datos y empaquetarlos en un archivo archivador de tienda antes de la publicación, puede cargar los grupos de elementos de base de datos utilizando Loader package de WebSphere Commerce.

La primera parte de este capítulo explica los grupos de elementos de base de datos de WebSphere Commerce y cómo se determinan las agrupaciones. La segunda parte describe el proceso de carga de estos grupos de elementos de base de datos, en la base de datos de WebSphere Commerce. Antes de leer esta sección, debe revisar cuidadosamente la información del Capítulo 37, “Visión general de la carga de datos de tienda”, en la página 361, que ayuda a entender lo que debe saber para cargar grupos de elementos de base de datos con Loader package.

Grupos de elementos de base de datos

Los elementos de base de datos se dividen en grupos para simplificar los procesos de creación y carga. Estos *grupos de elementos de base de datos* constan de un conjunto de tablas relacionadas de forma lógica. El orden en el que se organiza un grupo de elementos de base de datos es importante para la carga, porque es necesario que los datos existan antes de cargar la relación entre ellos.

Para cargar el conjunto entero de elementos de base de datos para una tienda, debe seguir las instrucciones del apartado “Secuencia de carga de elementos de base de datos” en la página 406. Para cargar un grupo individual de elementos de base de datos, debe asegurarse de que este grupo sea lógicamente completo. Por ejemplo, al publicar un archivador de tienda, puede elegir omitir los elementos de base de datos de catálogo, que puede publicar posteriormente. En ese caso, los elementos de base de datos dependientes del catálogo (inventario, listas de precio y algunos datos de envío e impuestos) también quedan sin publicarse. Para publicar los datos omitidos, asegúrese de que los elementos de base de datos de catálogo sean lógicamente completos: es decir, se deben proporcionar los elementos base, las entradas de catálogo, los atributos, etc. También debe publicar los elementos de base de datos dependientes que deben ser lógicamente completos por sí mismos. En otras palabras, cada código de artículo debe tener definidos el inventario, el precio, el envío y los impuestos apropiados. En este caso, los datos de catálogo relacionados que están lógicamente completos se denominan colectivamente *grupo de elementos de base de datos de catálogo*.

Los elementos de base de datos de WebSphere Commerce descritos en los capítulos anteriores de esta guía pueden disponerse en grupos. Un grupo es un conjunto de datos completo lógicamente, que se puede cargar de forma individual. Cada grupo de elementos de base de datos consta de tablas de base de datos de WebSphere Commerce y tiene dependencias externas como se describe en el Apéndice C, “Grupos de elementos de base de datos”, en la página 467. La lista de tablas se basa en las tiendas de ejemplo de WebSphere Commerce, aunque la lista es aplicable a cualquier tienda genérica. Recuerde que la lista de tablas para cada grupo de elementos de base de datos no es exhaustiva, sino que se proporciona como directriz general. Puede que necesite incluir o excluir algunas tablas en función de las necesidades específicas de su tienda.

Secuencia de carga de elementos de base de datos

Existe un orden determinado a seguir para cargar los grupos de elementos de base de datos de forma satisfactoria. Se considera que cada grupo es estructuralmente completo e independiente de los otros grupos de elementos de base de datos. Sin embargo, dentro de un grupo de elementos de base de datos hay relaciones de clave externa. Estas relaciones (con datos de otros grupos) se denominan *dependencias externas* de un grupo de elementos de base de datos.

Las dependencias externas de un grupo de elementos de base de datos deben satisfacerse antes de cargar dicho grupo en la base de datos de WebSphere Commerce. Los grupos definidos como dependencia externa de un grupo concreto de elementos de base de datos, deben cargarse en primer lugar. Puede encontrar la lista de dependencias externas y tablas relacionadas en el apartado "Dependencias de grupos de elementos de base de datos" en la página 467.

Nota: Una tienda de WebSphere Commerce tiene un propietario de tienda. Puede utilizar la organización por omisión como el propietario por omisión.

► Business Para cargar este grupo, debe crear una nueva organización en lugar de utilizar la organización por omisión.

Cargue los grupos de elementos de base de datos en el siguiente orden:

1. Sólo grupos de elementos de base de datos dependientes de datos de rutina de carga.
 - a. En primer lugar, cargue los elementos de base de datos de la **organización**.
2. Grupo de elementos de base de datos dependiente del propietario del centro de despacho.
 - a. Elementos de base de datos de **despacho de pedidos**. A excepción del grupo de elementos de base de datos de la organización, hay otros grupos de elementos de base de datos que tienen una dependencia externa directa o indirecta con los datos definidos en este grupo.
3. Grupos de elementos de base de datos dependientes de la organización de propietarios de la tienda.
 - a. Los elementos de base de datos de **control de acceso** son dependientes de la organización propietaria de la tienda (ORGENTITY_ID). Ninguno de los demás grupos de elementos de base de datos tiene una dependencia en los datos definidos en este grupo, lo que significa que los elementos de base de datos de control de acceso pueden cargarse en cualquier momento. Sin embargo, el propietario del control de acceso debe ser el mismo que el propietario de la tienda.
 - b. Los elementos de base de datos de **tienda** son dependientes de la organización propietaria de la tienda (ORGENTITY_ID).

La tienda puede hacer referencia a un centro de despacho de pedidos. La organización propietaria de la tienda también puede ser la organización propietaria del centro de despacho de pedidos.
4. Grupos de elementos de base de datos dependientes de los elementos de base de datos de la tienda. Los grupos siguientes pueden cargarse en cualquier orden:
 - a. Elementos de base de datos de **campañas**.
 - b. Elementos de base de datos de **mandatos**.
 - c. Elementos de base de datos de **monedas**.
 - d. Elementos de base de datos de **políticas**.

- e. Elementos de base de datos de **envíos**.
 - f. Elementos de base de datos de **impuestos**.
5. Otros grupos de elementos de base de datos.
- a. Los elementos de base de datos de **catálogo** son dependientes de los grupos de elementos de base de datos de envío e impuestos.
 - b. Los elementos de base de datos de la **tienda por omisión** tienen dependencias externas con el grupo de elementos de base de datos de envío. Si no hay elementos de base de datos de envío, no es necesario rellenar con datos este grupo.
 - c. Los elementos de base de datos de **contrato** son dependientes de los elementos de la organización. Los elementos de base de datos de contrato no se cargan directamente. Consulte el apartado “Publicación de elementos de contrato” en la página 420 para obtener más información. Debe cargar los elementos de contrato después de cargar los demás grupos de elementos de base de datos.

En el Apéndice C, “Grupos de elementos de base de datos”, en la página 467 puede consultar el contenido de los grupos de elementos de base de datos, tal como están formados por las tiendas de ejemplo de WebSphere Commerce.

Carga de una tienda

Para ayudarle a cargar elementos de base de datos, en el sitio Web de WebSphere Commerce encontrará paquetes de ejemplo. Para cargar datos XML para una tienda entera en la base de datos de WebSphere Commerce, realice lo siguiente:

1. Revise la información siguiente:
 - a. El Apéndice B, “Creación de los datos”, en la página 465.
 - b. El Apéndice C, “Grupos de elementos de base de datos”, en la página 467, ya que debe conocer qué tablas de base de datos y archivos de elementos de base de datos de WebSphere Commerce están afectados.
 - c. El Capítulo 37, “Visión general de la carga de datos de tienda”, en la página 361, que proporciona información básica sobre Loader package.
2. Planifique el proceso de carga para un conjunto completo de elementos de base de datos de la tienda. Tanto si desea cargar un solo grupo de elementos de base de datos, según las instrucciones del apartado “Carga de grupos de elementos de base de datos” en la página 413, como si desea cargar una tienda entera, el proceso básico sigue siendo el mismo. En los pasos siguientes, utilizará o creará los archivos siguientes para el proceso de carga:
 - a. uno o más archivos de elementos de base de datos para cada grupo. Si carga la tienda completa, necesitará todos los archivos de elementos de base de datos creados. Por ejemplo, necesitará un archivo *elemento_base_datos.xml* (como *campaign.xml*, *catalog.xml* o *currency.xml*) y archivos *elemento_base_datos.xml* específicos del entorno nacional, aparte, para los entornos nacionales a los que la tienda dé soporte. Los ejemplos de este tipo de archivos se encuentran en los archivadores de tienda de ejemplo de WebSphere Commerce. Los archivadores de tienda de ejemplo se organizan según el modelo de negocio del directorio *dir_instalación_WC/sampletores*. Tenga en cuenta que no todos los grupos de elementos de base de datos requieren información específica del entorno nacional.
 - b. un nuevo archivo XML que agrupa todos los archivos de elementos de base de datos XML para su tienda, que contiene las referencias a entidades XML y que también contiene el elemento raíz para la tienda entera. Este

archivo se denomina archivo XML principal del grupo de elementos de base de datos. Encontrará este archivo en el paquete de ejemplo, denominado store-data-assets.xml.

- c. un archivo DTD nuevo que define todos los tipos de datos necesarios para los archivos XML de un grupo de elementos de base de datos, que se denomina *archivo DTD principal del grupo de elementos de base de datos*. Encontrará este archivo en el paquete de ejemplo, denominado store-data-assets.dtd.
 - d. un segundo archivo DTD que define las dependencias externas. Quizá tenga que incluir este archivo en el *archivo DTD principal del grupo de elementos de base de datos*. Encontrará este archivo en el paquete de ejemplo, denominado ForeignKeys.dtd.
 - e. un tercer archivo DTD que contiene la definición de todas las tablas de WebSphere Commerce. El archivo wcs.dtd ya existe en WebSphere Commerce y está situado en el directorio *dir_instalación_WC/schema/xml*. Quizá tenga que incluir este archivo en el *archivo DTD principal del grupo de elementos de base de datos*. Si no ha personalizado el esquema de WebSphere Commerce, a continuación, puede utilizar este archivo sin modificarlo.
3. Cree archivos XML de elementos de base de datos como se ha indicado en los capítulos anteriores de esta guía. Si ha completado las tareas de los capítulos de los elementos, estos archivos XML ya existen. Los archivos de elementos de base de datos no deben contener ninguna declaración de DTD ni directivas de página al principio del archivo porque esto puede producir conflictos al concatenar los archivos. El único archivo que debe tener un elemento raíz es el archivo XML principal de elementos de base de datos.

Nota: Si tiene archivos de elementos de base de datos para más de un idioma, cada archivo debe empezar por `<?xml encoding = codificación específica del entorno nacional>`. Por ejemplo, los archivos de elementos de base de datos ingleses deben especificar `<?xml encoding = "UTF-8"?>`, pero los archivos franceses deben especificar `<?xml encoding = "ISO-8859-1"?>`.

4. Cree el archivo XML principal del grupo de elementos de base de datos para todo el conjunto de datos de la tienda. Este archivo contiene entidades de referencia para incluir diversos archivos XML de elementos de base de datos para la tienda. Mediante la utilización de entidades de referencia externa puede concatenar los archivos XML para simplificar el mandato ID Resolve y el proceso de carga. Además, los alias internos utilizados dentro de cada archivo XML pueden ser externos con respecto a otro archivo XML de elementos de base de datos dentro de un grupo o en otros grupos, al cargar más de un grupo a la vez. Un analizador XML sustituirá la referencia externa por el contenido del archivo al que hace referencia la entidad de referencia externa.

Utilizando el ejemplo siguiente como guía para cargar el conjunto entero de datos de tienda, puede crear el archivo de grupo de elementos de base de datos basándose en este extracto:

```
<?xml version="1.0"?>
<!DOCTYPE import SYSTEM "store-data-assets.dtd">
<import>
<!Fulfillment data group -->
&fulfillment.xml;

<!-- Store data group -->
&store.xml;
&en_US_store.xml;
&fr_FR_store.xml;
```

```

<!-- Tax data group -->
&tax.xml;
&en_US_tax.xml;
&fr_FR_tax.xml;
&taxfulfill.xml;

<!-- Shipping data group -->
&shipping.xml;
&en_US_shipping.xml;
&fr_FR_shipping.xml;
&shipfulfill.xml;

<!-- Catalog data group -->
&catalog.xml;
&en_US_catalog.xml;
&fr_FR_catalog.xml;
&storecatalog.xml;
&storefulfill.xml;
&offering.xml;
&store-catalog-tax.xml;
&store-catalog-shipping.xml;

<!-- Currency data group -->
&currency.xml;
&en_US_currency.xml;
&fr_FR_currency.xml;

<!-- Campaign data group -->
&campaign.xml;
&en_US_campaign.xml;
&fr_FR_campaign.xml;

<!-- Business policy data group -->
&businesspolicy.xml;
&en_US_businesspolicy.xml;
&fr_FR_businesspolicy.xml;

<!-- Access control data group -->
&accesscontrol.xml;
&en_US_accesscontrol.xml;
&fr_FR_accesscontrol.xml;

<!-- Other data groups -->
&command.xml;
&store-default.xml;
</import>
donde

```

- `import` es el elemento raíz del documento XML. El elemento raíz ya se ha definido en el archivo `wcs.dtd`, proporcionado con WebSphere Commerce, e incluye las definiciones de todas las tablas de base de datos WebSphere Commerce. Sin embargo, si ha personalizado el esquema de WebSphere Commerce, puede que necesite utilizar un elemento raíz diferente. Puede generar un nuevo archivo DTD que refleje el esquema personalizado o puede actualizar el archivo `wcs.dtd` existente.
- `store-data-assets.dtd` hace referencia al nombre del archivo DTD principal de grupo de elementos de base de datos que creará en el paso siguiente. El texto comentado separa los distintos grupos de elementos de base de datos para la tienda.
- `&elemento_base_datos.xml`; es una referencia de entidad XML al archivo de fragmentos XML de elemento de base de datos. La vía de acceso y ubicación de la base de datos se definen en el archivo DTD de grupo de elementos de base de datos. El nombre del archivo

Elemento_base_datos.xml. El archivo se modificará para que coincida con el elemento de base de datos que ya se ha creado para cada grupo.

- *&entorno_nacional_elemento_base_datos.xml*; es necesario para cada idioma que soporte la tienda. Si la tienda es monolingüe, entonces solamente hará referencia a un archivo. Si la tienda soporta más de un idioma, entonces necesitará una referencia para cada idioma. El extracto anterior supone que la tienda soporta los idiomas inglés y francés.
5. Cree un archivo DTD principal de grupo de elementos de base de datos que defina las entidades de referencia anteriores y los demás archivos DTD que son necesarios para los elementos de base de datos.

Utilizando el ejemplo siguiente como guía para todo el conjunto de elementos de bases de datos de la tienda, puede crear el archivo DTD principal de grupo de elementos de base de datos:

```
<!ENTITY % wcs.dtd SYSTEM "vía
completa para archivo wcs.dtd de WebSphere Commerce.">
%wcs.dtd;

<!ENTITY % ForeignKeys.dtd SYSTEM "ForeignKeys.dtd">
%ForeignKeys.dtd;
<!ENTITY fulfillment.xml SYSTEM "data/fulfillment.xml">
<!ENTITY en_US_fulfillment.xml SYSTEM "data/en_US/fulfillment.xml">
<!ENTITY fr_FR_fulfillment.xml SYSTEM "data/fr_FR/fulfillment.xml">

<!ENTITY store.xml SYSTEM "data/store.xml">
<!ENTITY en_US_store.xml SYSTEM "data/en_US/store.xml">
<!ENTITY fr_FR_store.xml SYSTEM "data/fr_FR/store.xml">

<!ENTITY tax.xml SYSTEM "data/tax.xml">
<!ENTITY en_US_tax.xml SYSTEM "data/en_US/tax.xml">
<!ENTITY fr_FR_tax.xml SYSTEM "data/fr_FR/tax.xml">
<!ENTITY taxfulfill.xml SYSTEM "data/taxfulfill.xml">

<!ENTITY shipping.xml SYSTEM "data/shipping.xml">
<!ENTITY en_US_shipping.xml SYSTEM "data/en_US/shipping.xml">
<!ENTITY fr_FR_shipping.xml SYSTEM "data/fr_FR/shipping.xml">
<!ENTITY shipfulfill.xml SYSTEM "data/shipfulfill.xml">

<!ENTITY catalog.xml SYSTEM "data/catalog.xml">
<!ENTITY en_US_catalog.xml SYSTEM "data/en_US/catalog.xml">
<!ENTITY fr_FR_catalog.xml SYSTEM "data/fr_FR/catalog.xml">
<!ENTITY store-catalog.xml SYSTEM "data/store-catalog.xml">
<!ENTITY storefulfill.xml SYSTEM "data/storefulfill.xml">
<!ENTITY offering.xml SYSTEM "data/offering.xml">
<!ENTITY store-catalog-tax.xml SYSTEM "data/store-catalog-tax.xml">
<!ENTITY store-catalog-shipping.xml SYSTEM "data/store-catalog-shipping.xml">

<!ENTITY currency.xml SYSTEM "data/currency.xml">
<!ENTITY en_US_currency.xml SYSTEM "data/en_US/currency.xml">
<!ENTITY fr_FR_currency.xml SYSTEM "data/fr_FR/currency.xml">

<!ENTITY campaign.xml SYSTEM "data/campaign.xml">
<!ENTITY en_US_campaign.xml SYSTEM "data/en_US/campaign.xml">
<!ENTITY fr_FR_campaign.xml SYSTEM "data/fr_FR/campaign.xml">

<!ENTITY businesspolicy.xml SYSTEM "data/businesspolicy.xml">
<!ENTITY en_US_businesspolicy.xml SYSTEM "data/en_US/businesspolicy.xml">
<!ENTITY fr_FR_businesspolicy.xml SYSTEM "data/fr_FR/businesspolicy.xml">

<!ENTITY accesscontrol.xml SYSTEM "data/accesscontrol.xml">
<!ENTITY en_US_accesscontrol.xml SYSTEM "data/en_US/accesscontrol.xml">
<!ENTITY fr_FR_accesscontrol.xml SYSTEM "data/fr_FR/accesscontrol.xml">
```



```
<!ENTITY command.xml SYSTEM "data/command.xml">
<!ENTITY store-defaults.xml SYSTEM "data/store-defaults.xml">
donde
```

- `wcs.dtd` hace referencia al archivo DTD que contiene datos definidos fuera de su grupo de elementos de base de datos. Este archivo, que se proporciona con WebSphere Commerce también define el elemento raíz utilizado en el archivo XML de grupo de elementos de base de datos.
- `ForeignKeys.dtd` hace referencia al archivo DTD que define elementos distintos del elemento raíz. Este archivo contiene todas las declaraciones y definiciones de referencias de entidad XML para las dependencias externas fuera del grupo de elementos de base de datos. Por ello, los archivos XML contienen referencias a valores de clave externa que no se crean como parte del grupo de elementos de base de datos y que deben cargarse en la base de datos antes que este grupo.

Nota: Asegúrese de que la vía de acceso esté identificada correctamente. En este ejemplo, el archivo está en el mismo directorio que el archivo DTD principal de grupo de elementos de base de datos.

- `store.xml`, `en_US_store.xml` y `fr_FR_store.xml` son las entidades de referencia externa utilizadas en el archivo XML principal de grupo de elementos de base de datos, suponiendo que la tienda soporte los idiomas inglés y francés. Para utilizar la referencia, siga el convenio de referencia de entidad: *&nombre_alias*;
 - `elemento_base_datos.xml` hace referencia al nombre de los archivos XML de los que se cargan los elementos de base de datos. Este nombre cambiará para coincidir con los archivos de elementos de base de datos ya creados para cada grupo. Los ejemplos de este tipo de archivos se encuentran en los archivadores de tienda de ejemplo de WebSphere Commerce. Los archivadores de tienda de ejemplo se organizan según el modelo de negocio del directorio `dir_instalación_WC/samplestores`.
 - Los archivos `entorno_nacional_elemento_base_datos.xml` son necesarios para cada idioma que soporte la tienda, y se encuentran bajo los directorios anteriores. Si la tienda es monolingüe, entonces solamente hará referencia a un archivo. Si la tienda soporta más de un idioma, entonces necesitará un archivo específico del entorno nacional para cada idioma. El extracto anterior supone que la tienda soporta los idiomas inglés y francés.
6. Cada grupo de elementos de base de datos necesita información definida fuera de su dominio o su conjunto de datos, puesto que cada grupo puede tener dependencias externas. Puede proporcionar estos datos en un archivo DTD. Por ejemplo, el grupo de elementos de base de datos de tienda tiene las dependencias externas siguientes:

```
bootstrap.LANGUAGE.LANGUAGE_ID, bootstrap.MEMBER.MEMBER_ID,
bootstrap.SETCURRE.SETCURRE_ID, fulfillment.FFMCENTER.FFMCENTER_ID
```

Al cargar un grupo de elementos de base de datos o el conjunto entero de elementos de tienda, se deberán definir las dependencias externas de la base de datos de WebSphere Commerce. Para utilizar estos datos, siga la referencia de entidad XML correspondiente. Por ejemplo, para utilizar los datos definidos por la entidad `ffmcenter_id`, deberá escribir `&ffmcenter_id`; en el archivo XML. Utilizando como guía el ejemplo siguiente de elementos de base de datos de tienda, puede crear su archivo DTD basándose en este extracto, denominado `ForeignKeys.dtd`:



```

<!ENTITY es_ES "-1">
<!ENTITY fr_FR "-2">
<!ENTITY de_DE "-3">
<!ENTITY it_IT "-4">
<!ENTITY es_ES "-5">
<!ENTITY pt_BR "-6">
<!ENTITY zh_CN "-7">
<!ENTITY zh_TW "-8">
<!ENTITY ko_KR "-9">
<!ENTITY ja_JP "-10">
<!ENTITY MEMBER_ID "-2000">
<!ENTITY ffmcenter_id "10001">
donde







```

- MEMBER_ID es el número de referencia interno que identifica al propietario de la tienda.
- ffmcenter es el número de referencia del centro de despacho de pedidos de la tienda. Puesto que la tienda puede utilizar más de un centro de despacho de pedidos, se puede definir más de uno en el archivo NonStoreForeignKeys.dtd.
- *entorno nacional* es el número de referencia de WebSphere Commerce para cada entorno nacional (que se identifica mediante el país o región y el idioma). Los valores están ubicados en la tabla de base de datos LANGUAGE.

Nota: Si está dividiendo un archivador de tienda existente en grupos de elementos de base de datos, asegúrese de que todas las referencias, por ejemplo, al alias @ffmcenter_id se sustituyen por su referencia de entidad correspondiente: &ffmcenter_id;

- Una vez que se hayan creado todos los archivos de datos, ejecute el mandato IDResolve para el archivo XML principal de grupo de elementos de base de datos para resolver los datos, tal como se describe en el apartado “Mandato ID Resolve” en la página 365.
- Ejecute el mandato Load en el archivo de datos resueltos tal como se describe en el apartado “Mandato Load” en la página 374. Para comprobar el proceso de carga, consulte los archivos de anotaciones:
 -  idresgen.db2.log y massload.db2.log
 -  idresgen.oracle.log y massload.oracle.log

Los archivos de anotaciones se encuentran bajo:

-     *dir_instalación_WC/logs*
 -  *dir_usuario_WC/instances/nombre_instancia/logs*
-  Ejecute el mandato AccountImport como se describe en el apartado “Publicación de elementos de cuenta de negocio” en la página 420.
 - Si es aplicable, publique los contratos tal como se describe en el apartado “Publicación de elementos de contrato” en la página 420.
 - Complete las tareas del apartado “Publicación de elementos de escaparate y archivos de configuración de tienda copiándolos en WebSphere Commerce Server” en la página 423

Carga de grupos de elementos de base de datos

Para cargar datos XML para un solo grupo de elementos de base de datos, en la base de datos de WebSphere Commerce, realice lo siguiente:

1. Revise la información siguiente:
 - a. El Apéndice B, “Creación de los datos”, en la página 465
 - b. El Apéndice C, “Grupos de elementos de base de datos”, en la página 467, ya que debe conocer qué archivos de elementos y tablas de base de datos de WebSphere Commerce están afectados.
 - c. El Capítulo 37, “Visión general de la carga de datos de tienda”, en la página 361, que proporciona información básica sobre Loader package.
2. Planifique el proceso de carga y decida qué grupo de elementos de base de datos cargará. Tanto si desea cargar el conjunto entero de elementos de base de datos de tienda, según las instrucciones del apartado “Carga de una tienda” en la página 407, como si desea cargar un solo grupo de elementos de base de datos, el proceso básico sigue siendo el mismo. En los pasos siguientes, utilizará o creará los archivos siguientes para el proceso de carga:
 - a. uno o más archivos de elementos de base de datos, en función del grupo que elija. Por ejemplo, si carga los elementos de grupo de base de datos de tienda, necesitará un archivo `store.xml` y un archivo `store.xml` independiente para cada entorno nacional que soporte la tienda. Los ejemplos de este tipo de archivos se encuentran en los archivadores de tienda de ejemplo de WebSphere Commerce. Los archivadores de tienda de ejemplo se organizan según el modelo de negocio del directorio `dir_instalación_WC/samplestores`. Tenga en cuenta que no todos los grupos de elementos de base de datos requieren información específica del entorno nacional.
 - b. un nuevo archivo XML que agrupa todos los archivos de elementos de base de datos XML, que contiene las referencias a entidades XML y que también contiene el elemento raíz para los elementos de base de datos. Este archivo se denomina archivo XML principal del grupo de elementos de base de datos. Encontrará este archivo en el paquete de ejemplo, denominado `store-all-assets.xml`.
 - c. un archivo DTD nuevo que define todos los tipos de datos necesarios para los archivos XML de un grupo de elementos de base de datos, que se denomina *archivo DTD principal del grupo de elementos de base de datos*. Encontrará este archivo en el paquete de ejemplo, denominado `store-all-assets.dtd`.
 - d. un segundo archivo DTD que define las dependencias externas. Quizá tenga que incluir este archivo en el *archivo DTD principal del grupo de elementos de base de datos*. Encontrará este archivo en el paquete de ejemplo, denominado `Non database asset groupForeignKeys.dtd`.
 - e. un tercer archivo DTD que contiene la definición de todas las tablas de WebSphere Commerce. El archivo `wcs.dtd` ya existe en WebSphere Commerce y está situado en el directorio `dir_instalación_WC/schema/xml`. Quizá tenga que incluir este archivo en el *archivo DTD principal del grupo de elementos de base de datos*. Si no ha personalizado el esquema de WebSphere Commerce, a continuación, puede utilizar este archivo sin modificarlo.
3. Cree archivos XML de elementos de base de datos para el grupo que cargará, tal como se ha indicado en los capítulos anteriores de esta guía. Si ha completado las tareas de los capítulos de los elementos, estos archivos XML ya existen. Los archivos de elementos de base de datos no deben contener ninguna declaración de DTD ni directivas de página al principio del archivo,

porque esto puede producir conflictos al concatenar los archivos. Además, y para una mayor simplicidad, quizá decida no crear ningún elemento raíz. El único archivo que debe tener un elemento raíz es el archivo XML principal de elementos de base de datos.

Nota: Si tiene archivos de elementos de base de datos para más de un idioma, cada archivo debe empezar por `<?xml encoding = codificación específica del entorno nacional>`. Por ejemplo, los archivos de elementos de base de datos ingleses deben especificar `<?xml encoding = "UTF-8"?>`, pero los archivos franceses deben especificar `<?xml encoding = "ISO-8859-1"?>`. Debe asegurarse de que la codificación que especifique coincida con la codificación real del archivo.

4. Cree el archivo XML principal del grupo de elementos de base de datos para cada grupo que desee cargar. Este archivo contiene entidades de referencia para incluir diversos archivos XML en un grupo de elementos de base de datos, o más. Mediante la utilización de entidades de referencia externa puede concatenar los archivos XML para simplificar el mandato IDResolve y el proceso de carga. Además, los alias internos utilizados dentro de cada archivo XML pueden ser externos con respecto a otro archivo XML de datos dentro de un grupo o en otros grupos, al cargar más de un grupo a la vez. Un analizador XML sustituirá la referencia externa por el contenido del archivo al que hace referencia la entidad de referencia externa.

Utilizando el ejemplo siguiente como guía para cargar un solo grupo de elementos de base de datos de tienda, puede crear el archivo XML de grupo de elementos de base de datos basándose en este extracto:

```
<?xml version="1.0"?>
<!DOCTYPE import SYSTEM "store-assets.dtd">
<import>
&store.xml;
&en_US_store.xml;
&fr_FR_store.xml;
</import>
```

donde

- `import` es el elemento raíz del documento XML. El elemento raíz ya se ha definido en el archivo `wcs.dtd`, proporcionado con WebSphere Commerce, e incluye las definiciones de todas las tablas de base de datos WebSphere Commerce. Sin embargo, si ha personalizado el esquema de WebSphere Commerce, puede que necesite utilizar un elemento raíz diferente. Puede generar un nuevo archivo DTD que refleje el esquema personalizado o puede actualizar el archivo `wcs.dtd` existente.
 - `store-assets.dtd` hace referencia al nombre del archivo DTD principal de grupo de elementos de base de datos que creará en el paso siguiente.
 - `&store.xml;` es una referencia de entidad XML al archivo XML de grupo de elementos de base de datos. La vía de acceso y ubicación se definen en el archivo DTD de grupo de elementos de base de datos. Este nombre cambiará para coincidir con los archivos de elementos ya creados para cada grupo.
 - `entorno nacional_store.xml;` es necesario para cada idioma que soporte la tienda. Si la tienda es monolingüe, entonces solamente hará referencia a un archivo. Si la tienda soporta más de un idioma, entonces necesitará una referencia para cada idioma. El extracto anterior supone que la tienda soporta los idiomas inglés y francés.
5. Cree un archivo DTD principal de grupo de elementos de base de datos que defina las entidades de referencia anteriores y los demás archivos DTD que son necesarios para el grupo.

Utilizando el ejemplo siguiente como guía para cargar un solo grupo de elementos de base de datos de tienda, puede crear el archivo DTD principal de grupo de elementos de base de datos para cualquier grupo de datos:

```
<!ENTITY % wcs.dtd SYSTEM "vía
completa para archivo wcs.dtd de WebSphere Commerce.">
%wcs.dtd;

<!ENTITY % ForeignKeys.dtd SYSTEM "ForeignKeys.dtd">
%ForeignKeys.dtd;
<!ENTITY store.xml SYSTEM "store.xml">
<!ENTITY en_US_store.xml SYSTEM "en_US/store.xml">
<!ENTITY fr_FR_store.xml SYSTEM "fr_FR/store.xml">
donde
```

- `wcs.dtd` hace referencia al archivo DTD que contiene datos definidos fuera de su grupo de elementos de base de datos. Este archivo, que se proporciona con WebSphere Commerce también resuelve y define el elemento raíz utilizado en el archivo XML de grupo de elementos de base de datos.
- `ForeignKeys.dtd` hace referencia al archivo DTD que define elementos distintos del elemento raíz. Este archivo contiene todas las declaraciones y definiciones de referencias de entidad XML para las dependencias externas fuera del grupo de elementos de base de datos. Por ello, los archivos XML contienen referencias a valores de clave externa que no se crean como parte del grupo de elementos de base de datos y que deben cargarse en la base de datos antes que este grupo.

Nota: Asegúrese de que la vía de acceso esté identificada correctamente. En este ejemplo, el archivo está en el mismo directorio que el archivo DTD de grupo de elementos de base de datos.

- `store.xml`, `en_US_store.xml` y `fr_FR_store.xml` son las entidades de referencia externa utilizadas en el archivo XML de grupo de elementos de base de datos. Para utilizar la referencia, siga el convenio de referencia de entidad: *&nombre_alias;*.
 - `store.xml` hace referencia al archivo de datos del grupo del que se cargan los elementos de base de datos. Este nombre cambiará para coincidir con los archivos de elementos de base de datos ya creados para cada grupo. Tenga en cuenta que los archivos XML específicos del entorno nacional se encuentran bajo el directorio `dir_instalación_WC/samplestores`.
 - Los archivos *vía de acceso_store.xml* son necesarios para cada idioma que soporte la tienda, y se encuentran bajo los directorios anteriores. Si la tienda es monolingüe, entonces solamente hará referencia a un archivo. Si la tienda soporta más de un idioma, entonces necesitará un archivo específico del entorno nacional para cada idioma. El extracto anterior supone que la tienda soporta los idiomas inglés y francés.
6. Cada grupo de elementos de base de datos necesita información definida fuera de su dominio o su conjunto de datos, puesto que cada grupo puede tener dependencias externas. Puede proporcionar estos datos en un archivo DTD. Por ejemplo, el grupo de elementos de base de datos de tienda tiene las dependencias externas siguientes:

```
bootstrap.LANGUAGE.LANGUAGE_ID, bootstrap.MEMBER.MEMBER_ID,
bootstrap.SETCURR.SETCURR_ID, fulfillment.FFMCENTER.FFMCENTER_ID
```

Al cargar un grupo de datos o el conjunto entero de datos de la tienda, se deberán definir las dependencias externas siguientes de la base de datos de WebSphere Commerce. Para utilizar estos datos, siga la referencia de entidad XML correspondiente. Por ejemplo, para utilizar los datos definidos por la









entidad `ffmcenter_id`, deberá escribir `&ffmcenter_id;` en el archivo XML. Utilizando como guía el ejemplo siguiente de grupo de elementos de base de datos de tienda, puede crear su archivo DTD basándose en este extracto, denominado `Nondatabase asset groupForeignKeys.dtd`:

```
<!ENTITY es_ES "-1">
<!ENTITY fr_FR "-2">
<!ENTITY de_DE "-3">
<!ENTITY it_IT "-4">
<!ENTITY es_ES "-5">
<!ENTITY pt_BR "-6">
<!ENTITY zh_CN "-7">
<!ENTITY zh_TW "-8">
<!ENTITY ko_KR "-9">
<!ENTITY ja_JP "-10">
<!ENTITY MEMBER_ID "-2000">
<!ENTITY ffmcenter_id "10001">
```

donde

- `MEMBER_ID` es el número de referencia interno que identifica al propietario de la tienda.
- `ffmcenter` es el número de referencia del centro de despacho de pedidos de la tienda. Puesto que la tienda puede utilizar más de un centro de despacho de pedidos, se puede definir más de uno en el archivo `ForeignKeys.dtd`.
- *entorno nacional* es el número de referencia de WebSphere Commerce para cada entorno nacional (que se identifica mediante el país o región y el idioma). Los valores están ubicados en la tabla de base de datos `LANGUAGE`.

Notas:

- Si está dividiendo un archivador de tienda existente en grupos de elementos de base de datos, asegúrese de que todas las referencias, por ejemplo, al alias `@ffmcenter_id` se sustituyen por su referencia de entidad correspondiente: `&ffmcenter_id;`.
 - Si hace referencia a un ID de miembro que ya existe en la base de datos, puede sustituir el alias interno utilizado en los datos de tienda de ejemplo por `&MEMBER_ID;`. De lo contrario, puede incluir el XML necesario para resolver el ID de miembro utilizando `@member_id`.
- Una vez que se hayan creado todos los archivos de datos, ejecute el mandato `IDResolve` para el archivo XML de grupo de elementos de base de datos para resolver los datos, tal como se describe en el apartado “Mandato ID Resolve” en la página 365.
 - Ejecute el mandato `Load` en el archivo de datos resueltos tal como se describe en el apartado “Mandato Load” en la página 374. Para comprobar el proceso de carga, consulte los archivos de anotaciones:
 -  `idresgen.db2.log` y `massload.db2.log`
 -  `idresgen.oracle.log` y `massload.oracle.log`
- Los archivos de anotaciones se encuentran bajo:
-     `dir_instalación_WC/logs`
 -  `dir_usuario_WC/instances/nombre_instancia/logs`
-  Ejecute el mandato `AccountImport` como se describe en el apartado “Publicación de elementos de cuenta de negocio” en la página 420.
 - Si es aplicable, publique los contratos tal como se describe en el apartado “Publicación de elementos de contrato” en la página 420.

11. Complete las tareas del apartado “Publicación de elementos de escaparate y archivos de configuración de tienda copiándolos en WebSphere Commerce Server” en la página 423

Capítulo 39. Publicación de cuentas de negocio y contratos

Loader package no puede cargar algunos de los elementos de base de datos de la tienda, (cuentas de negocio y contratos). Puede publicar estos elementos de base de datos utilizando la Consola de administración o desde la línea de mandatos, como parte de la publicación de una opción de tienda completa, como se describe en el Capítulo 36, “Publicación de una tienda completa”, en la página 347, o puede publicar cuentas de negocio y contratos utilizando los mandatos correspondientes. Estos mandatos son los siguientes:

- AccountImport— Crea cuentas de negocio a partir del archivo `businessaccount.xml` del archivador de tienda.
- ContractImportApprovedVersion— Crea un contrato del archivo `contract.xml`. Si el contrato está en estado activo, el mandato crea y despliega el contrato. Aunque el archivo `contract.xml` contenga más de un contrato, sólo es necesario llamar al mandato una vez.

Nota: Para obtener más información, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

Los elementos de cuenta de negocio están incluidos en forma de archivos XML en algunos de los archivadores de tienda de ejemplo proporcionados con WebSphere Commerce. Sin embargo, se recomienda que cree los elementos de cuenta de negocio utilizando las herramientas proporcionadas, en lugar de crear archivos XML para estos elementos. Para obtener más información sobre cómo crear estos elementos utilizando las herramientas proporcionadas, consulte la Ayuda en línea a la producción de WebSphere Commerce. Las instrucciones para publicar cuentas de negocio se incluyen en las secciones siguientes, en el caso de que elija publicar los archivos XML correspondientes proporcionados con los archivadores de tienda de ejemplo o crear los suyos propios.

Nota: Si no utiliza la Consola de administración para publicar las cuentas de negocio o los contratos, para poder publicarlos debe primero publicar los elementos de tienda y de catálogo. Concretamente, necesita los identificadores de tienda y de catálogo, así como el ID de la organización propietaria de la tienda y los ID de las organizaciones compradoras asociadas al contrato, si las hay. Si los términos y condiciones de su contrato no especifican un catálogo, no es necesario publicar un catálogo antes de publicar una cuenta de negocio o un contrato. Si publica estos elementos utilizando el programa de utilidad de publicación de la Consola de administración o de la línea de mandatos, asegúrese de seleccionar la opción de catálogo o de que la tienda ya tenga un catálogo publicado. Si publica estos elementos utilizando los mandatos correspondientes, asegúrese de que ya ha cargado los elementos listados más arriba en la base de datos.

Publicación de cuentas de negocio y contratos utilizando la Consola de administración o la línea de mandatos

Las cuentas de negocio y los contratos pueden publicarse utilizando la Consola de administración o el programa de utilidad de publicación desde la línea de mandatos. Para publicar las cuentas de negocio o los contratos utilizando la Consola de administración o la línea de mandatos, los elementos deben estar

empaquetados en formato de archivador de tienda. Para obtener más información sobre cómo empaquetar los elementos de escaparate como un archivador de tienda, consulte la Parte 9, “Empaquetado de la tienda”, en la página 337.

Para obtener instrucciones paso a paso sobre cómo publicar elementos utilizando la Consola de administración o la línea de mandatos, consulte la Ayuda en línea a la producción de WebSphere Commerce.

Publicación de cuentas de negocio y contratos utilizando mandatos

Si prefiere no empaquetar los elementos como un archivador de tienda, puede publicar las cuentas de negocio y los contratos utilizando los mandatos correspondientes:

- AccountImport— Crea cuentas de negocio a partir del archivo `businessaccount.xml` del archivador de tienda.
- ContractImportApprovedVersion— Importa un contrato aprobado o activo, de un archivo XML a WebSphere Commerce Server. Antes de importar el contrato, el mandato comprueba que el mandato que se importa contenga los términos y condiciones necesarios, y que sea un contrato válido.

Publicación de elementos de cuenta de negocio

Para publicar los elementos de cuenta de negocio desde las tiendas de ejemplo, realice lo siguiente:

1. Copie `ForeignKeys.dtd` en la ubicación siguiente:
 - `dir_instalación_WC/xml/trading/dtd`
 - `400 dir_usuario_WC/instances/nombre_instancia/xml/trading/xml`

ForeignKeys.dtd contiene los valores de entidad a los que hace referencia *businessaccount.xml*.
2. Copie `businessaccount.xml` en la ubicación siguiente:
 - `dir_instalación_WC/xml/trading/xml`
 - `400 dir_usuario_WC/instances/nombre_instancia/xml/trading/xml`
3. Abra la Consola de administración. Conéctese como administrador.
4. En un navegador, escriba el siguiente URL:
 - `https://nombresistpral:8002/webapp/wcs/admin/servlet/AccountImport?fileName=businessaccount.xml&URL=`
El URL al que se redirecciona una vez se ha finalizado correctamente.

Nota: Para obtener más información sobre la sintaxis del mandato y los parámetros, consulte la Ayuda en línea a la producción y el desarrollo de WebSphere Commerce.

Publicación de elementos de contrato

Para publicar los elementos de contrato desde las tiendas de ejemplo, realice lo siguiente:

1. Copie `ForeignKeys.dtd` en la ubicación siguiente:
 - `dir_instalación_WC/xml/trading/dtd`
 - `400 dir_usuario_WC/instances/nombre_instancia/xml/trading/xml`

ForeignKeys.dtd contiene los valores de entidad a los que hace referencia *businessaccount.xml*.

2. Copie `contract.xml` en la ubicación siguiente:
 - `dir_instalación_WC/xml/trading/xml`
 - `400 dir_usuario_WC/instancias/nombre_instancia/xml/trading/xml`
3. Abra la Consola de administración. Conéctese como administrador.
4. En un navegador, escriba lo siguiente:
 - `https://nombre_sistema_principal:8002/webapp/wcs/admin/servlet/ContractImportApprovedVersion?fileName=contract.xml&xsd=false&URL=ContractDisplay`
5. Si la tienda contiene varios archivos `contract.xml` (por ejemplo, archivos de contrato específicos del entorno nacional), repita los pasos 1 a 4 para cada archivo `contract.xml`.

Capítulo 40. Publicación de los elementos de escaparate y de los archivos de configuración de tiendas

La publicación de los elementos de escaparate, los archivos HTML y JSP, los archivos de propiedades o paquetes de recursos, y las imágenes y los gráficos que crean las páginas de la tienda, forma parte del proceso de creación de una tienda operativa. Puede publicar los elementos de escaparate utilizando la Consola de administración o desde la línea de mandatos, como parte de la publicación de una opción de tienda completa, que se describe en el Capítulo 36, “Publicación de una tienda completa”, en la página 347, o puede publicar los elementos de escaparate simplemente copiando los elementos en una ubicación específica de WebSphere Commerce Server.

Si publica los archivos JSP que contienen las tiendas de ejemplo y cambia el flujo de tienda, tendrá que publicar los archivos de configuración de la tienda que forman parte de dicho archivador de tienda. Los archivos de configuración XML para la tienda están situados en el directorio siguiente del archivador de tienda:

```
WEB-INF/xml/tools/stores/StoreDirectory/devtools/flow
```

El archivo de propiedades para la herramienta Cambiar flujo está situado en el directorio siguiente del archivador de tienda:

```
StoreDirectory/devtools/flow/ui
```

Publicación de los elementos de escaparate y de los archivos de configuración de tienda utilizando la Consola de administración o la línea de mandatos

Puede publicar los elementos de escaparate y los archivos de configuración de tienda utilizando la Consola de administración o el programa de utilidad de publicación desde la línea de mandatos.

Para publicar los elementos de escaparate y los archivos de configuración de tienda utilizando la Consola de administración o la línea de mandatos, los elementos de escaparate y los archivos de configuración de tienda deben estar empaquetados en formato de archivador de tienda. Para obtener más información sobre cómo empaquetar los elementos de escaparate como un archivador de tienda, consulte la Parte 9, “Empaquetado de la tienda”, en la página 337.

Publicación de elementos de escaparate y archivos de configuración de tienda copiándolos en WebSphere Commerce Server

Si prefiere no empaquetar los elementos como un archivador de tienda, puede publicar de todos modos los elementos de escaparate copiándolos directamente en WebSphere Commerce Server. Los elementos Web (HTML, archivos JSP, imágenes y gráficos) deben copiarse en el directorio raíz de documentos de la aplicación Web. Los paquetes de recursos o los archivos de propiedades deben copiarse en la vía de acceso de propiedades de las aplicaciones. Los archivos de configuración de tienda se deben copiar en las ubicaciones adecuadas bajo los módulos Web de tiendas o de herramientas.

Nota: Cuando desempaqueta los archivadores de tienda de ejemplo en el módulo Web de tiendas (reteniendo la estructura de vía de acceso) los elementos de archivo se colocarán en la ubicación correcta. No obstante, los archivos de propiedades que utiliza la herramienta Cambiar flujo (StoreDirectory/devtools/flow/ui/*.properties) se deben copiar en la vía de acceso ToolsStoresPropertiesPath definida en el archivo de configuración de WebSphere Commerce, `nombre_instancia.xml`.

Para copiar los elementos de escaparate y los archivos de configuración de tienda en WebSphere Commerce Server, realice lo siguiente:

1. Copie los archivos JSP, HTML, archivos de inclusión, imágenes y gráficos en el directorio de la tienda (*dirtienda*) en el directorio raíz de documentos de la aplicación Web de tiendas:

- `dir_instalación_WAS/installedApps/nombre_célula/nombre_instancia_WC.ear/Stores.war/dirtienda`
- `400 dir_usuario_WAS/nombre_instancia_WAS/installedApps/nombre_célula/WC_nombre_instancia.ear/Stores.war/dirtienda`

donde *dirtienda* es el valor de la columna DIRECTORY de la tabla de base de datos STORE.

2. Copie los paquetes de recursos y los archivos de propiedades en la vía de acceso de propiedades de la aplicación:

- `dir_instalación_WAS/installedApps/nombre_célula/nombre_instancia_WC.ear/Stores.war/WEB-INF/classes/dirtienda`
- `400 dirusu_WAS/nombre_inst_WAS/InstalledApps/nombre_célula/nombre_instancia_WC.ear/Stores.war/WEB-INF/classes/dirtienda`

3. Copie los archivos de configuración de tienda en las ubicaciones definidas en el archivo de configuración de WebSphere Commerce, `nombre_instancia.xml`. Este archivo está ubicado en el directorio siguiente:

- `dir_instalación_WC/instances/instance_name/xml`
- `400 dir_usuario_WC/instances/nombre_instancia/xml`

Los archivos de configuración de la tienda se copian en las ubicaciones siguientes:

- El XML de configuración de la tienda (WEB-INF/xml/tools/stores/storedirectory/devtools/flow) se copia en ToolsStoresXMLPath. Esta vía de acceso está definida en el archivo de configuración de WebSphere Commerce, `nombre_instancia.xml`.
 - Los archivos de propiedades de configuración de la tienda se copian en ToolsStoresPropertiesPath. Esta vía de acceso está definida en el archivo de configuración de WebSphere Commerce, `nombre_instancia.xml`.
4. Inicie la tienda utilizando uno de los siguientes métodos:
 - Utilice el mandato StoreCatalogDisplay:
`StoreCatalogDisplay?storeId=idTienda&catalogId=idCat&langId=idIdioma`
donde
 - *idTienda* es el valor que se encuentra en la columna STORE_ID de la tabla de base de datos STORE,
 - *idCat* es el valor que se encuentra en la columna CATALOG_ID de la tabla de base de datos CATALOG,

- idIdioma es el valor de la columna LANGUAGE_ID de la tabla de base de datos LANGUAGE para un entorno nacional específico. Para obtener una lista de los valores de WebSphere Commerce por omisión, consulte la tabla de base de datos LANGUAGE.
- Si la tienda se basa en una tienda de ejemplo de WebSphere Commerce, configure el URL de la tienda editando el archivo `index.jsp` bajo:
 - `dir_instalación_WAS/installedApps/nombre_instancia/nombre_instancia_WC.ear/Stores.war/dir_tienda`
 - `400 dir_usuario_WAS/nombre_instancia_WAS/installedApps/nombre_célula/nombre_instancia_WC.ear/Stores.war/dirtienda`

Añada los valores correctos para los parámetros siguientes:

- `nombresistpral` es el nombre totalmente calificado de la máquina de WebSphere Commerce,
- `idTienda` es el valor que se encuentra en la columna STORE_ID de la tabla de base de datos STORE,
- `idCatálogo` es el valor que se encuentra en la columna CATALOG_ID de la tabla de base de datos CATALOG,
- `idIdioma` es el valor de la columna LANGUAGE_ID de la tabla de base de datos LANGUAGE para un entorno nacional específico. Para obtener una lista de los valores de WebSphere Commerce por omisión, consulte la tabla de base de datos LANGUAGE.

Para ver la tienda en un navegador, inicie el siguiente URL: `http://nombre_sistpral/webapp/wcs/stores/servlet/dirtienda/index.jsp`

Parte 11. Adición de características de WebSphere Commerce a la tienda

Para añadir ciertas características disponibles en WebSphere Commerce a la tienda, debe completar algunos pasos manualmente. Los capítulos de esta sección describen cómo se añaden las siguientes características a la tienda:

- Capítulo 41, “Adición del servicio de atención al cliente a la tienda”, en la página 429
- Capítulo 42, “Adición de zonas de e-marketing a la tienda”, en la página 455


Capítulo 41. Adición del servicio de atención al cliente a la tienda

La característica de atención al cliente de WebSphere Commerce proporciona soporte de servicio de atención al cliente en tiempo real mediante una interfaz de texto síncrona que utiliza el servidor Lotus Sametime. Cuando está habilitado el servicio de atención al cliente en la tienda, el cliente puede entrar en la tienda, pulsar un enlace y conectarse con un representante de servicio al cliente (CSR). A continuación, el cliente puede comunicarse con un CSR a través de Internet.

Nota: Este capítulo describe cómo activar el servicio de atención al cliente en la tienda. Sin embargo, para poder habilitar el servicio de atención al cliente en la tienda, deberá instalar primero un servidor Sametime y configurarlo para que funcione con WebSphere Commerce. Para obtener más información, consulte la publicación *WebSphere Commerce, Guía de software adicional*. Si el servidor Sametime no utiliza el mismo servidor LDAP que WebSphere Commerce, también deberá registrar los CSR en la Consola de administración para habilitarlos para utilizar el servicio de atención al cliente. Para obtener más información sobre esta tarea, así como sobre los conceptos generales de la atención al cliente y cómo emplea un CSR el servicio de atención al cliente, consulte la ayuda en línea de WebSphere Commerce.

Nota:

Puede habilitar el servicio de atención al cliente en la tienda de un modo fácil y rápido utilizando WebSphere Commerce Accelerator, si crea la tienda basándose en uno de los ejemplos siguientes:

-  Directo a B2B (ToolTech)
- Directo al consumidor (FashionFlow)

Después de publicar la tienda utilizando el programa de utilidad de publicación en la Consola de administración, abra WebSphere Commerce Accelerator, seleccione el menú **Tiendas**, a continuación, seleccione **Cambiar flujo** y habilite las características de atención al cliente. Para obtener las instrucciones paso a paso en la Ayuda en línea a la producción de WebSphere Commerce consulte el tema "Cambio de los flujos de tienda mediante WebSphere Commerce Accelerator"

Sin embargo, si no crea la tienda utilizando como base una tienda de ejemplo, tendrá que realizar algunas tareas para habilitar el servicio de atención al cliente en su tienda. El resto de este capítulo describe los conceptos y los pasos necesarios para activar el servicio de atención al cliente en una tienda que no esté basada en uno de los ejemplos.

Nota: Las tiendas de ejemplo  ToolTech y FashionFlow muestran cómo debe implementarse el servicio de atención al cliente y proporcionan el código que se puede utilizar en la tienda para habilitar el servicio de atención al cliente. Este capítulo hace referencia a estas dos tiendas de ejemplo para ilustrar cómo puede activar el servicio de atención al cliente en la tienda. Cuando lea este capítulo, asegúrese de que tiene la versión más reciente de las tiendas de ejemplo.

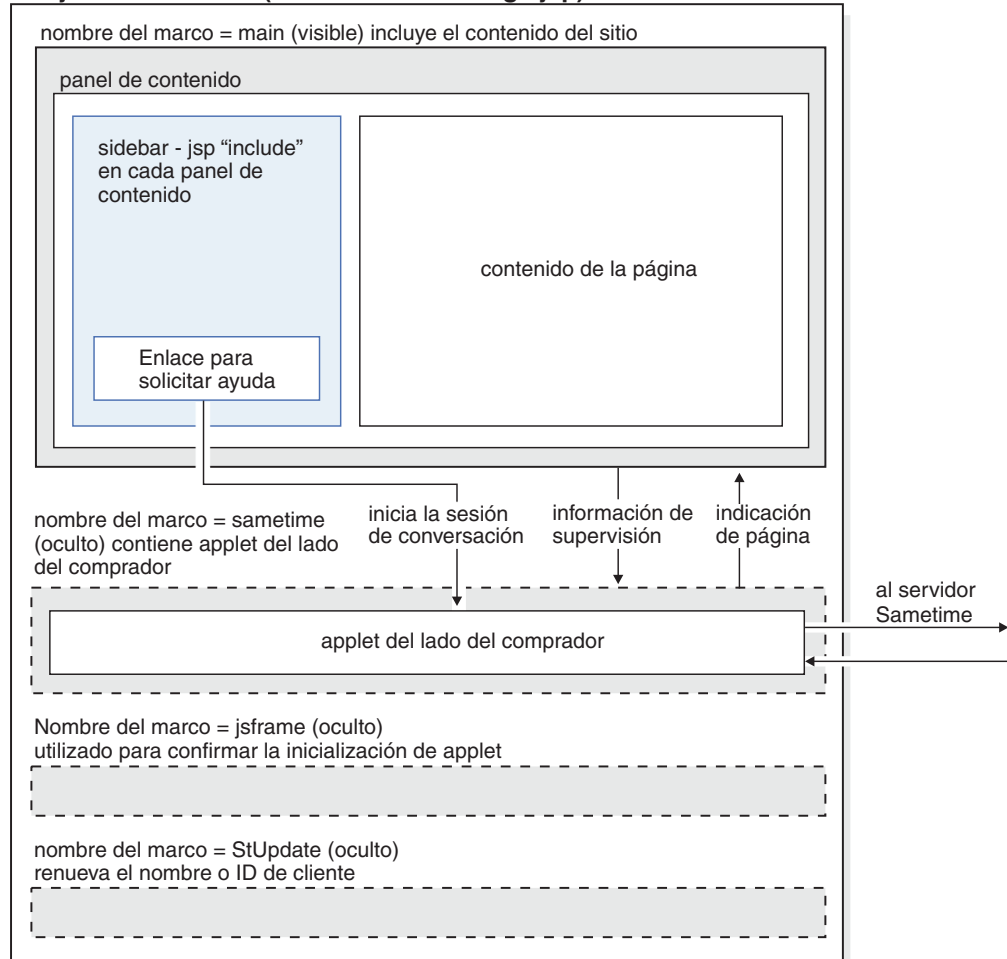
Nota: Para dar soporte a la compatibilidad con versiones anteriores y con la JVM incorporada que incluyen los navegadores Internet Explorer y Netscape versión 4.x, el código del applet se ha desarrollado con los componentes JDK1.1 y AWT. Por lo tanto, algunas de las características disponibles con JDK 1.2 o superior (incluido el soporte de idioma, el soporte bidireccional y el acceso) no estarán disponible o no estarán totalmente soportadas. Los plug-ins de Sun Java y el navegador Netscape versión 6 y 7 no están soportados.

Descripción del servicio de atención al cliente en una tienda

Cuando un cliente selecciona el enlace del servicio de atención al cliente, por ejemplo **Conversación en directo con Asistencia al cliente**, en una tienda que tiene activado el servicio de atención al cliente, se inicia un applet que contiene la ventana para la conversación. Este applet se ejecuta dentro de un conjunto de marcos ocultos que no interfiere con el diseño externo del sitio.

El diagrama siguiente muestra el diseño del conjunto de marcos.

conjunto de marcos (en storeFamesetPage.jsp)



El conjunto de marcos incluye cuatro marcos:

- **Main:** el marco que contiene el contenido de la tienda, incluidos los archivos que crean las páginas de la tienda, es decir, los archivos que crean los archivos de cabecera, pie de página, el texto de la página y los archivos sidebar. Los visitantes de la tienda pueden ver el contenido de este marco. Tenga en cuenta

que el marco principal (main) contiene las conexiones siguientes con el marco Sametime: un enlace con el servicio de atención al cliente e información de supervisión. La información de supervisión se describe detalladamente en el apartado "Supervisión de los clientes que utilizan el servicio de atención al cliente" en la página 442.

- Sametime: el marco que contiene el applet del servicio de atención al cliente. Este marco no está visible para los visitantes de la tienda. Sin embargo, cuando un cliente pulsa el enlace para iniciar el applet, el cliente verá la ventana del servicio de atención al cliente. Este marco también pasa información al marco principal (main), mediante la característica de indicación de página.
- jsframe: el marco que confirma que el applet se ha cargado correctamente. El contenido de este marco no se muestra a los clientes.
- StUpdate: El marco renueva la información del cliente, incluido el nombre o ID de cliente.

Utilización del conjunto de marcos

Al iniciar el applet de servicio al cliente en un conjunto de marcos se separa el código del applet del código de las páginas de la tienda. Como muestra el diagrama anterior, las páginas de la tienda están contenidas en el marco principal del conjunto de marcos, mientras que el código del applet está en el marco Sametime. Al separar el código del applet de las páginas de la tienda se disminuye el tráfico de la red, ya que el applet solamente se baja una vez.

Al utilizar un conjunto de marcos se puede mantener la conexión con el servidor Sametime. Si el applet formara parte de cada página y no del conjunto de marcos, se crearía una nueva sesión Sametime cada vez que el cliente accediera a una página nueva. Dado que el applet del servicio de atención al cliente se conecta al servidor Sametime de forma anónima, crear una sesión nueva cada vez que un cliente accede a una página nueva no permitiría hacer un seguimiento de las actividades de los clientes en la tienda. Al utilizar el conjunto de marcos, se mantiene la sesión Sametime original del cliente y las actividades del cliente se devuelven al servidor Sametime a medida que cambian los atributos.

Aspectos relacionados con la utilización de conjuntos de marcos

Aunque el método recomendado para implementar el servicio de atención al cliente en la tienda es utilizar un conjunto de marcos, debe tener en cuenta los aspectos siguientes relacionados con la utilización de conjuntos de marcos:

- Un solo punto de entrada: los clientes solamente pueden utilizar el servicio de atención al cliente si navegan por la tienda dentro de la infraestructura. Del mismo modo, los representantes del servicio al cliente solamente pueden supervisar el movimiento de los clientes a través del conjunto de marcos. Para garantizar que los clientes naveguen por la tienda mediante el conjunto de marcos, deben acceder al sitio mediante un solo punto de entrada. Si un cliente accede a la tienda a través de otra página, por ejemplo, a través de una página de catálogo, no estará en el conjunto de marcos.
- Marcadores: cuando los clientes utilicen el conjunto de marcos solamente podrán obtener los marcadores del URL del conjunto de marcos, no de las páginas individuales.
- Renovación: cuando un cliente esté en el conjunto de marcos y pulse renovar, se le devolverá a la página original, como se ha codificado en el conjunto de marcos, que puede ser la misma página que se está visualizando en ese momento en el navegador.
- Ajuste del tamaño de la ventana del navegador: Si un cliente ajusta el tamaño de la ventana del navegador mientras está en el conjunto de marcos, puede que el

navegador vuelva a cargar automáticamente la dirección de entrada. Si se vuelve a cargar la dirección de entrada, puede que termine la conexión con el servidor Sametime. En esta situación, los diferentes navegadores se comportan de forma distinta.

- Seguridad: Para que funcione correctamente, las páginas del conjunto de marcos de atención al cliente deben poder comunicarse entre sí. Esta comunicación se habilita mediante las llamadas a la función JavaScript. Cuando un cliente está navegando por un sitio a través de un conjunto de marcos, cada marco individual, así como el conjunto de marcos (el URL de la barra de ubicación), mantiene su propia conexión, ya sea insegura (http, por omisión el puerto 80) ya sea segura (https, por omisión el puerto 443). Si un cliente está examinando la tienda a través de una conexión insegura, todos los marcos del conjunto de marcos están en HTTP. En este caso no hay aspectos relacionados con SSL. Sin embargo, si el cliente va a una página segura (por ejemplo, la página de registro), el marco principal (main) del conjunto de marcos se conmutará a HTTPS, mientras el resto de los marcos seguirán siendo no seguros (http). En este caso, el cliente no podrá iniciar el applet del servicio de atención al cliente. El navegador no autorizará el inicio del applet, porque la función JavaScript que llama al applet (seguro, puerto 443) parece proceder de un servidor diferente del URL de la barra de ubicación del navegador (HTTP, puerto 80). Para solucionar este problema, debe utilizar siempre el mandato StoreFramesetView para entrar en la tienda, ya que activará la conexión HTTPS para el URL del conjunto de marcos.
- Para permitir las comunicaciones entre marcos, el applet Java y las funciones JavaScript también deben poder comunicarse. Dado que la base del código del applet está en el servidor Sametime y no en WebSphere Commerce Server, algunos navegadores (por ejemplo Netscape versiones 6 y 7) que utilizan el plug-in de Sun Java impiden que JavaScript se comunique con los applets Java que se cargan desde un sistema principal diferente.

Utilización del servicio de atención al cliente sin un conjunto de marcos

El servicio de atención al cliente funcionará sin un conjunto de marcos, no obstante, ninguna de las actividades de supervisión de actividades de clientes estará disponible en CSR. Esto es, la información recopilada acerca de un cliente será precisa sólo en el momento en que el cliente envíe la petición de ayuda. Si después de ese punto el cliente cambia las páginas o añade más elementos al carro de la compra, la información nueva no se actualizará para CSR en la lista de supervisión.

Para poder utilizar el servicio de atención al cliente sin un conjunto de marcos, haga lo siguiente:

1. Añada el código siguiente al archivo JSP que incluye en enlace para iniciar la página de conversación:

```
<script>
function LaunchChat()
{
<%
    String pname = request.getRequestURI();
    int indpn = pname.lastIndexOf('/');
    indpn = pname.lastIndexOf('/', indpn-1);
    if(indpn != -1) { pname = pname.substring(indpn+1);}
    String headerType = (String) request.getAttribute("liveHelpPageType");
    if (headerType==null) { headerType="";}
    if (headerType.equals("personal"))
    {
```

```

%>
currentPageURL='PERSONAL_URL';
<% } else { %>
currentPageURL=escape(location.href);
<% } %>
currentpageDesc="<%=pname%>";
chatURL="<%=com.ibm.commerce.tools.util.UIUtil.getWebappPath(request)%>
CCChatPageView?"
+ "pageURL=" + currentPageURL
+ "&pageDesc=" + currentpageDesc;
WindowName="";
chatAttr="toolbars=no,location=no,directories=no,status=yes,
menubar=no,scrollbars=no,resizable=no,width=360,height=400"
window.open(chatURL,WindowName, chatAttr);
return true;
}
</script>

```

Nota: Asegúrese de que la anchura y altura sean las mismas que se han definido en el código APPLET de CustomerCareChatSetup.jsp.

- Si la página que añade al código anterior es una página personal, asegúrese de que ha definido la página personal antes de este bloque de código con la sentencia siguiente: `request.setAttribute("liveHelpPageType", "personal");`
- Es posible que desee también añadir parejas de valores de parámetros adicionales que utilicen la serie del URL para los atributos personalizados o para otra información útil que se pueden recoger en CustomerCareChatSetup.jsp mediante el objeto de petición.
- Añada el código siguiente para iniciar la página de conversación desde un enlace o una imagen:

```

<A HREF="javascript:void(LaunchChat());">
<FONT COLOR="#ffffff" STYLE="font-size : 8pt">
<%= tooltechtext.getString("LiveHelp")%></FONT></A>

```
- (Opcional) Personalice el archivo CustomerCareChatSetup.jsp para pasar información adicional sobre clientes. Los parámetros del applet siguientes permiten personalizar CustomerCareChatSetup.jsp.

Tabla 15.

Nombre del parámetro personalizable	Descripción	Nota
CHAT_FONT_SIZE	El tamaño de font que se ha de utilizar en el área de conversación.	El valor por omisión es 12.
CHAT_FONT_COLOR	El color del font que se va a utilizar para los mensajes de entrada del área de conversación.	El valor por omisión es azul (#0000FF).
CHAT_NAME_LENGTH	La longitud de los caracteres que se invierten para visualizar los nombres de usuario en el área de conversación.	El valor por omisión es 15.

Tabla 15. (continuación)

WAIT_RANGE_1	El valor es un entero, si el número de clientes que están esperando en la tienda es menor que este valor, se muestra el mensaje de espera 1. De lo contrario se muestra el mensaje de espera correspondiente al valor WAIT_RANGE_2.	Utilice el valor -1 cuando se visualizará únicamente el mensaje 1.
WAIT_RANGE_2	El valor es un entero, si el número de clientes que están esperando en la tienda es menor que este valor pero mayor que WAIT_RANGE_1, se muestra el mensaje de espera 2. De lo contrario se muestra el mensaje de espera correspondiente al valor WAIT_RANGE_3.	Se ignora si el valor de WAIT_RANGE_1 es -1. Utilice el valor -1 para inhabilitar este rango.
WAIT_RANGE_3	El valor es un entero, si el número de clientes que están esperando en la tienda es menor que este valor pero mayor que WAIT_RANGE_2, se muestra el mensaje de espera 3. De lo contrario, se muestra el mensaje de espera 4.	Se ignora si el valor de WAIT_RANGE_2 es -1. Utilice el valor -1 para inhabilitar este rango.
contentFrame	El nombre del marco que se utiliza para las páginas de tienda de WebSphere Commerce normales.	El valor por omisión es "_blank". Cuando el CSR devuelve un URL, iniciará siempre una ventana de navegador nueva.
COUNTER_UNIT_WAIT	El valor es un entero e indica la frecuencia con que debe aumentar en 1 el contador de espera.	El valor por omisión es de 30 segundos. Asegúrese de que el valor sea el mismo que el que se ha definido en el archivo CustomerCareMonitorList.jsp de la tienda.
WIDTH	La anchura preferida del marco de conversación en píxeles.	El valor por omisión es de 360 píxeles. La longitud del mensaje de invitación afectará a la anchura final.
HEIGHT	La altura del marco de conversación en píxeles.	El valor por omisión es de 400 píxeles.
QUEUE_ID	El ID de cola de Atención al cliente	Si se proporciona un ID de cola válido la petición de ayudará se colocará directamente en la cola.
ML_ATTRIBUTES	La lista de los ID de atributos de supervisión personalizados, separados por comas.	Ejemplo: <PARAM name="ML_ATTRIBUTES" value="8001,9002">

Tabla 15. (continuación)

ML_ATTRIBUTE_xxxx	Proporcione el valor del atributo de supervisión personalizado xxxx (donde xxxx es el ID de atributo)	Ejemplo: <PARAM name="ML_ATTRIBUTE_8001" value="A value string "> Asegúrese de que el ID de atributo ya se haya definido en el parámetro ML_ATTRIBUTES o se ignorará este parámetro.
-------------------	---	---

Definición del servicio de atención al cliente

Antes de utilizar el servicio de atención al cliente en la tienda, es necesario realizar las siguientes tareas de definición:

- Definición de la lista de supervisión de la tienda
- Definición de la lista de temas de la tienda
- Definición de los URL de la tienda

En los apartados siguientes se describen estos conceptos, utilizando ejemplos de las tiendas de ejemplo para ilustrar el modo en que se ha definido la información.

Definición de la lista de supervisión de la tienda

Cada tienda contiene un archivo que define los elementos que va a supervisar el CSR. El archivo, CustomerCareMonitorList.jsp, se encuentra en el directorio siguiente:

- *dir_instalación_WAS/installedApps/nombre_instancia/nombre_instancia_WC.ear/Stores.war/dir_tienda/CustomerServiceArea/CollaborationSection*
- **400** *dir_usuario_WAS/nombre_instancia_WAS/installedApps/nombre_célula/nombre_instancia_WC.ear/Stores.war/dir_tienda/CustomerServiceArea/CollaborationSection*

El archivo CustomerCareMonitorList.jsp se carga en el applet del CSR cuando WebSphere Commerce Accelerator llama al mandato CCMonitorListView. Este mandato devuelve un archivo JSP que, a su vez, devuelve la siguiente serie XML que define la información de supervisión de la tienda.

```
<?xml version="1.0" encoding="UTF-8"?>
<WCS_CUSTOMERCARE>
  <MONITORING_LIST>
    <ATTR ID="4" LABEL="MonitoringVisitorsTableCurrentPage" ></ATTR>
    <ATTR ID="3010" LABEL="MonitoringVisitorsTableInSite" UNIT="30" ></ATTR>
    <ATTR ID="3011" LABEL="MonitoringVisitorsTableInPage" UNIT="30" ></ATTR>
    <ATTR ID="6" LABEL="MonitoringVisitorsTableCart" ></ATTR>
  </MONITORING_LIST>
</WCS_CUSTOMERCARE>
```

El elemento de atributo (ATTR), según se ha ilustrado anteriormente, define un atributo de supervisión para el cliente.

Tabla 16.

Nombre del atributo	Descripción	Nota
ID	El ID de atributo Sametime que se utiliza para realizar un seguimiento de este elemento.	Obligatorio

Tabla 16. (continuación)

Label	La clave de etiqueta de la descripción de este elemento en el archivo de propiedades. La descripción correspondiente se visualizará en el applet CSR.	Obligatorio
Unit	El valor entero que indica el número de segundos, transcurrido el cual, el contador se aumentará en 1.	Opcional. Solamente lo utilizan los atributos de contador definidos previamente.

Cuando el CSR inicia el applet CSR, el applet carga la serie XML y crea una lista de supervisión. La lista de supervisión muestra el valor de la serie con el atributo Sametime correspondiente. Si es necesario, las funciones JavaScript recuperan los valores de atributo de las páginas de tienda adecuadas.

Nota: Si la serie XML contiene un error, o si no se puede localizar el archivo `CustomerCareMonitorList.jsp`, el applet del CSR utiliza la lista de supervisión por omisión que solamente incluye el nombre del cliente.

Archivo `CustomerCareMonitorList.jsp` de ejemplo: El ejemplo de código siguiente pertenece a la tienda de ejemplo B2B, ToolTech:

```
<%
// Crear una serie XML
String strCfg=ECLivehelpConstants.EC_CC_XML_HEADER
    + LiveHelpConfiguration.getOpenTagString(ECLivehelpConstants.EC_CC_XML_ROOT)
    + LiveHelpConfiguration.getOpenTagString(ECLivehelpConstants.
EC_CC_XML_MONITORING_LIST);
%>

<% // modificar este bloque para personalizar la lista de supervisión
    strCfg=strCfg + LiveHelpConfiguration.getMonitorAttributeElementString
(ECLivehelpConstants.EC_CC_ST_ATTR_PAGE_URL, "MonitoringVisitorsTableCurrentPage")
    + LiveHelpConfiguration.getCloseTagString(ECLivehelpConstants.
EC_CC_XML_MONITORING_ATTR);
    strCfg=strCfg + LiveHelpConfiguration.getMonitorCounterAttributeElementString
(ECLivehelpConstants.EC_CC_ST_ATTR_SITE_COUNTER,
"MonitoringVisitorsTableInSite","30")
    + LiveHelpConfiguration.getCloseTagString(ECLivehelpConstants.
EC_CC_XML_MONITORING_ATTR);
    strCfg=strCfg + LiveHelpConfiguration.getMonitorCounterAttribute
ElementString
(ECLivehelpConstants.EC_CC_ST_ATTR_PAGE_COUNTER,
"MonitoringVisitorsTableInPage", "30")
    + LiveHelpConfiguration.getCloseTagString(ECLivehelpConstants.
EC_CC_XML_MONITORING_ATTR);
    strCfg=strCfg + LiveHelpConfiguration.getMonitorAttributeElementString
(ECLivehelpConstants.EC_CC_ST_ATTR_CART_ITEMS,
"MonitoringVisitorsTableCart")
    + LiveHelpConfiguration.getCloseTagString
(ECLivehelpConstants.EC_CC_XML_MONITORING_ATTR);
%>

<%
strCfg=strCfg
    + LiveHelpConfiguration.getCloseTagString
(ECLivehelpConstants.EC_CC_XML_MONITORING_LIST)
    + LiveHelpConfiguration.getCloseTagString
(ECLivehelpConstants.EC_CC_XML_ROOT);
%>
```

El ejemplo utiliza los métodos de programa de utilidad de la clase LiveHelpConfiguration para asegurarse de que el archivo XML es correcto. Para evitar problemas de análisis, el ejemplo utiliza también valores de atributo codificados. La tabla siguiente proporciona información más detallada sobre los métodos.

Tabla 17.

Método	Descripción	Notas
static String getOpenTagString(String tagName)	Devuelve una serie de códigos abierta	LiveHelpConfiguration. getOpenTagString("HELLO") returns string <HELLO>
static String getCloseTagString(String tagName)	Devuelve una serie de códigos cerrada	LiveHelpConfiguration. getCloseTagString ("HELLO") returns string </HELLO>
static String getMonitorAttribute ElementString(String sAttributeId, String sLabelKey)	Devuelve una serie de elementos ATTR para la lista de supervisión	LiveHelpConfiguration. getMonitorAttribute ElementString("1234", "myLabel") returns string <ATTR ID="1234" LABEL="myLabel" >
static String getMonitorCounter AttributeElementString (String sAttributeId, String sLabelKey, String sUnit)	Devuelve una serie de elementos Counter ATTR para la lista de supervisión	LiveHelpConfiguration .getMonitorCounter AttributeElementString ("1234", "myLabel", "60") returns string <ATTR ID="1234" LABEL="myLabel" UNIT="60" >

La tabla siguiente lista los ID de atributos Sametime definidos previamente que se pueden supervisar.

Tabla 18.

ID de atributo	Descripción	Serie de claves de etiqueta	Notas
ECLivehelpConstants. EC_CC_ST_ATTR_ PAGE_URL	La página Web por la que navega el comprador	"MonitoringVisitors TableCurrentPage"	Requiere entrada de las páginas de tienda
ECLivehelpConstants. EC_CC_ST_ATTR_ CART_ITEMS	El número de elementos que se han añadido al carro de la compra	"MonitoringVisitors TableCart"	Requiere entrada de las páginas de tienda
ECLivehelpConstants. EC_CC_ST_A TTR_SITE_COUNTER	El tiempo que ha estado el cliente en el sitio de la tienda	"Monitoring VisitorsTable InSite"	El atributo de contador, utiliza UNIT para establecer la frecuencia de actualización.

Tabla 18. (continuación)


ECLivehelpConstants. EC_CC_ST_ATTR _PAGE_COUNTER	El tiempo que ha estado el cliente en esta página.	"Monitoring VisitorsTable InPage"	El atributo de contador, utiliza UNIT para establecer la frecuencia de actualización. Para restablecer el contador, es necesario que haya alguna entrada en las páginas de la tienda.
ECLivehelpConstants. EC_CC_ST_ATTR _WAIT_COUNTER	El tiempo que ha estado esperando el cliente al CSR	"Monitoring Visitors TableInWait"	El atributo de contador, utiliza UNIT para establecer la frecuencia de actualización. Los restaura un cliente que envía una petición de ayuda o cuando un CSR vuelve a colocar a un cliente en la cola durante una conversación. También se utiliza para determinar a qué cliente dará servicio primero el CSR cuando utilice el botón "Atender siguiente".
ECLivehelpConstants. EC_CC_ST _ATTR REQ_QUEUE	Nombre de la cola en la que está la petición del cliente.	"Monitoring VisitorsTable Queue	El nombre parcial de la cola; se puede proporcionar cuando un cliente envía una petición en la página de la tienda
ECLivehelpConstants. EC_CC_ ST_ATTR _REQ_CSR_NAME	El nombre del CSR que conversa con el cliente	"Monitoring Visitors TableCSR"	Se actualizará cuando un CSR acepte una petición de una cola.

Nota: Los elementos de supervisión personalizados se pueden definir utilizando un ID de atributo mayor que 8000. Por ejemplo:

```
<ATTR ID="8004" LABEL="MonitoringVisitorsTableItem8004"></ATTR>
```

Definición de la lista de temas de la tienda

Cada tienda contiene un archivo que define los temas que puede utilizar el CSR durante la sesión de conversación con un cliente. El archivo CustomerCareStoreQuestionList.jsp se encuentra en el directorio siguiente:

- *dir_instalación_WAS*/installedApps/*nombre_instancia*/*nombre_instancia_WC.ear*/Stores.war/*dir_tienda*/CustomerServiceArea//CollaborationSection
-  400 *dir_usuario_WAS*/*nombre_instancia_WAS*/installedApps/*nombre_célula*/*nombre_instancia_WC.ear*/Stores.war/*dir_tienda*/CustomerServiceArea/CollaborationSection

Este archivo JSP devuelve la siguiente serie XML que define los temas de conversación de la tienda.

```
<?xml version="1.0" encoding="UTF-8"?>
<WCS_CUSTOMERCARE>
  <QUESTION_LIST>
    <GROUP NAME="TopicGroupName" >
      < QUESTION TITLE="TopicName" TEXT="Topic+Text" ></QUESTION>
    </GROUP>
  </QUESTION_LIST>
</WCS_CUSTOMERCARE>
```

El elemento de grupo (GROUP), según se ha ilustrado anteriormente, define un grupo de temas que puede tener varios subtemas. Tiene los atributos siguientes:

Tabla 19.

Nombre del atributo	Descripción	Nota
Name	Si se duplica un nombre de grupo, se utiliza la definición de grupo más reciente.	Obligatorio

El elemento de pregunta (QUESTION) define un solo tema. Tiene los atributos siguientes:

Tabla 20.

Nombre del atributo	Descripción	Nota
Title	Si se duplica un título, se utiliza la definición de título más reciente.	Obligatorio
Text	El contenido del tema.	Obligatorio

Archivo CustomerCareStoreQuestionList.jsp de ejemplo: El código siguiente procede de la tienda de ejemplo Directo a B2B, ToolTech, CustomerCareStoreQuestionList.jsp:

```
<% // Crear una serie XML
String strCfg=ECLivehelpConstants.EC_CC_XML_HEADER
  + LiveHelpConfiguration.getOpenTagString(ECLivehelpConstants.EC_CC_XML_ROOT)
  + LiveHelpConfiguration.getOpenTagString(ECLivehelpConstants.
EC_CC_XML_QUESTION_LIST);
%>
<% //desmarcar este bloque para añadir grupo de temas/temas
  // iniciar el bloque de grupo de temas, repetir para más grupos de temas
strCfg=strCfg + LiveHelpConfiguration.getTopicGroupElementString
("TopicGroupName");
  // iniciar el bloque de temas, repetir para todos los temas del
mismo grupo
  strCfg=strCfg + LiveHelpConfiguration.getTopicElementString("TopicName","
Topic Text")
  + LiveHelpConfiguration.getCloseTagString(ECLivehelpConstants.
EC_CC_XML_QUESTION_QUESTION);
  // fin del bloque de temas
strCfg=strCfg + LiveHelpConfiguration.getCloseTagString
(ECLivehelpConstants.EC_CC_XML_QUESTION_GROUP);
  // fin del bloque de grupo de temas
%>
<%
strCfg=strCfg + LiveHelpConfiguration.getCloseTagString
(ECLivehelpConstants.EC_CC_XML_QUESTION_LIST)
```

```
+ LiveHelpConfiguration.getCloseTagString
(ECLivehelpConstants.EC_CC_XML_ROOT);
%>
```

El ejemplo utiliza los métodos de programa de utilidad de la clase LiveHelpConfiguration para asegurarse de que el archivo XML es correcto. La tabla siguiente proporciona información más detallada sobre los métodos.

Notas:

1. Para evitar problemas de análisis, el ejemplo utiliza también valores de atributo codificados. También se utiliza la serie Unicode para evitar que se dañen los caracteres.
2. Para dar soporte a varios idiomas, puede utilizar un archivo de propiedades por idioma y recuperar el nombre o tema traducido según el ID de idioma seleccionado en la sesión del CSR.

Tabla 21.

Método	Descripción	Notas
static String getOpenTagString (String tagName)	Devuelve una serie de códigos abierta	LiveHelpConfiguration. getOpenTagString ("HELLO") returns string <HELLO>
static String getCloseTagString(String tagName)	Devuelve una serie de códigos cerrada	LiveHelpConfiguration. getCloseTagString ("HELLO") returns string </HELLO>
static String getTopicGroupElement String (String sGroupName)	Devuelve una serie de elementos GROUP ATTR para la lista de temas	LiveHelpConfiguration. getTopicGroupElementString ("myGroup") returns string <GROUP NAME = 'myGroup'>
static String getTopic ElementString (String sTitle, String sText)	Devuelve una serie de elementos QUESTION ATTR para la lista de temas	LiveHelpConfiguration. getTopicElementString ("myTitle", "myText") returns <QUESTION TITLE ="myTitle" TEXT ="myText">

Definición de la lista de URL de la tienda

Cada tienda contiene un archivo que define los URL que puede enviar el CSR al navegador de un cliente durante la sesión de conversación con un cliente. El archivo, CustomerCareStoreURLList.jsp, se encuentra en el directorio siguiente:

- *dir_instalación_WAS/installedApps/nombre_instancia/nombre_instancia_WC.ear/Stores.war/dir_tienda/CustomerServiceArea/CollaborationSection*
- **400** *dir_usuario_WAS/nombre_instancia_WAS/installedApps/nombre_célula/nombre_instancia_WC.ear/Stores.war/dir_tienda/CustomerServiceArea/CollaborationSection*

Este archivo JSP devuelve la siguiente serie XML que contiene la información de URL de la tienda.

```
<?xml version="1.0" encoding="UTF-8"?>
<WCS_CUSTOMERCARE>
  <URL_LIST>
    <GROUP NAME="URLGroupName" >
```

```

        <PAGE NAME="IBM" URL="http%3A%2F%2Fwww.ibm.com" ></PAGE>
    </GROUP>
</URL_LIST>
</WCS_CUSTOMERCARE>

```

El elemento de grupo (GROUP), según se ha ilustrado anteriormente, define un grupo de URL que puede tener varios URL. Tiene los atributos siguientes:

- Name: El nombre del grupo de URL. El nombre es un atributo obligatorio. Si se duplica un nombre de grupo, se utiliza la definición de grupo más reciente.

El elemento de página (PAGE) define una sola dirección de URL. Tiene los atributos siguientes:

- Name: El nombre de la página de URL. El nombre es un atributo obligatorio. Si se duplica un nombre de página, se utiliza la definición de página más reciente.
- URL: La dirección de URL de la página. El URL es un atributo obligatorio.

Archivo CustomerCareStoreURLList.jsp de ejemplo: El código siguiente procede de la tienda de ejemplo Directo a B2B, ToolTech, CustomerCareStoreURLList.jsp:

```

<% // Crear una serie XML
String strCfg=ECLivehelpConstants.EC_CC_XML_HEADER
    + LiveHelpConfiguration.getOpenTagString(ECLivehelpConstants.
EC_CC_XML_ROOT)
    + LiveHelpConfiguration.getOpenTagString(ECLivehelpConstants.
EC_CC_XML_URL_LIST);
%>

<% //desmarcar el bloque siguiente para añadir grupo de URL/páginas
// iniciar el bloque de grupo de URL, repetir para más grupos de URL
strCfg=strCfg + LiveHelpConfiguration.getURLGroupElementString("URLGroupName");
// inicio del bloque de páginas de URL,
// repetir para todas las páginas del mismo grupo
strCfg=strCfg + LiveHelpConfiguration.getURLPageElementString
("IBM","http://www.ibm.com")
    + LiveHelpConfiguration.getCloseTagString
(ECLivehelpConstants.EC_CC_XML_URL_PAGE);
// fin del bloque de páginas de URL
strCfg=strCfg + LiveHelpConfiguration.getCloseTagString
(ECLivehelpConstants.EC_CC_XML_URL_GROUP);
// fin del bloque de grupo de URL
%>
<%
strCfg=strCfg + LiveHelpConfiguration.getCloseTagString
(ECLivehelpConstants.EC_CC_XML_URL_LIST)
    + LiveHelpConfiguration.getCloseTagString(ECLivehelpConstants.EC_CC_XML_ROOT);
%>

```

El ejemplo utiliza los métodos de programa de utilidad de la clase LiveHelpConfiguration para asegurarse de que el archivo XML es correcto. La tabla siguiente proporciona información más detallada sobre los métodos.

Nota: Para evitar problemas de análisis, el ejemplo utiliza también valores de atributo codificados. También se utiliza la serie Unicode para evitar que se dañen los caracteres.

Tabla 22.

Método	Descripción	Notas
static String getOpenTagString(String tagName)	Devuelve una serie de códigos abierta	LiveHelpConfiguration .getOpenTagString ("HELLO") returns string <HELLO>

Tabla 22. (continuación)

static String getCloseTagString(String tagName)	Devuelve una serie de códigos cerrada	LiveHelpConfiguration. getCloseTagString ("HELLO") returns string </HELLO>
static String getTopicGroup ElementString (String sGroupName)	Devuelve una serie de elementos GROUP para la lista de URL	LiveHelpConfiguration. getTopicGroupElementString ("myGroup") returns string <GROUP NAME = 'myGroup'>
static String getURLPage ElementString (String sName, String sURL)	Devuelve una serie de elementos PAGE para la lista de URL	LiveHelpConfiguration. getURLPageElement String ("myName", "myURL") returns <QUESTION TITLE ="myName" TEXT ="myURL">

Supervisión de los clientes que utilizan el servicio de atención al cliente

Cuando un cliente selecciona un enlace en el marco principal, se devuelve una página nueva y se establece la cadena de sucesos consecutivos siguientes.

1. Esta página nueva incluye el archivo `CustomerCareHeaderSetup.jsp` en su cabecera.
2. `CustomerCareHeaderSetup.jsp` llama a la función JavaScript siguiente en la página del conjunto de marcos: `parent.setPageParams()`.
3. `parent.setPageParams()` actualiza las variables para que guarde la información de página de tienda actual y luego llama a `UpdateStInfo()`.
4. `UpdateStInfo()` vuelve a cargar el marco `StUpdate` que luego llama al mandato de vista `CCShopperInfoUpdatePageView`.
5. Cuando `CCShopperInfoUpdatePageView` se ejecuta, carga el archivo `CustomerCareInformationSetup.jsp` y recopila información del cliente incluidos el nombre del cliente, el ID y los elementos del carro de la compra del cliente.
6. A continuación, `CCShopperInfoUpdatePageView` llama a `setCustomerName()`, `setShoppingCartItems()` para que actualice la información de este cliente y luego llama a `changeSTAttributes()` para que actualice todos los atributos de cliente del applet.
7. `changeSTAttributes()` llama a varios métodos de applet para que actualicen los valores de atributo que CSR supervisa.
8. A continuación, `changeSTAttributes()` llama al método `changeWCSAttrs()` del applet para que envíe todos los valores de los atributos actualizados al sistema `Sametime`, el cual pasa a notificar al applet del CSR que se han modificado los valores de los atributos.

El servicio de atención al cliente le permite supervisar qué clientes se comunican con el representante del servicio al cliente de la tienda de uno de los modos siguientes:

- Obteniendo el nombre o el ID del cliente
- Determinando en qué página está navegando el cliente
- Realizando un seguimiento de los artículos del carro de la compra
- Realizando un seguimiento de los elementos de supervisión personalizados

Para obtener esta información, se añade el código personalizado a las páginas de la tienda. Los apartados siguientes describen cómo se implementan estas características de supervisión en las tiendas de ejemplo.

Obtención del nombre o el ID del cliente

Cuando se inicia el applet del servicio de atención al cliente y el representante del servicio de atención al cliente (CSR) está conectado, éste puede identificar quién está utilizando el applet según el nombre o el ID del comprador. Las tiendas de ejemplo incluyen un código especializado que funciona con el applet del servicio de atención al cliente para determinar el nombre o el ID del comprador. Este código determina si se trata de un cliente invitado, un cliente invitado que tiene artículos en un carro de la compra o un cliente registrado, a continuación, asigna un nombre o ID al cliente y lo pasa al applet del servicio de atención al cliente. A continuación, estos nombres se muestran al CSR. Por ejemplo, si el cliente es un comprador invitado que todavía no ha colocado nada en el carro de la compra, se asigna al cliente un ID generado con el ID de comprador -1002. Si el cliente es un comprador invitado que tiene artículos en el carro de la compra, se mostrará el ID del comprador y si el cliente está registrado, se visualizarán su nombre y sus apellidos.

Las tiendas de ejemplo obtienen el nombre o ID del cliente incluyendo CustomerCareInformationSetup.jsp en el archivo de cabecera de las páginas de tienda. El código siguiente del archivo CustomerCareInformationSetup.jsp obtiene el nombre y el ID del cliente.

```
<jsp:useBean id="userRDB" class="com.ibm.commerce.user.beans.
UserRegistrationDataBean" scope="page">
<% DataBeanManager.activate(userRDB, request); %>
</jsp:useBean>
<%
String customer_name="";
customer_name=userRDB.getUserId();
if (userRDB.findUser()){
    if (userRDB.getLastName() !=null && userRDB.getLastName().
length() > 0){
        if (locale.toString().equals("ja_JP")||locale.toString().
equals("ko_KR")||locale.toString().equals("zh_CN")||locale.toString().
equals("zh_TW"))
            {customer_name = userRDB.getLastName() + " " +
userRDB.getFirstName();}
        else
            {customer_name = userRDB.getFirstName() + " " +
userRDB.getLastName();}
    }
}
if (customer_name.equals("-1002")) {
    customer_name="";
}
else {
    // es necesario seleccionar artículos de pedido
    ....
}
        customer_name=customer_name.trim();
%>
```

Nota: Cada vez que un cliente navega por una página nueva de la tienda, se renueva el nombre o el ID del cliente.

El código siguiente del archivo CustomerCareInformationSetup.jsp actualiza el nombre y el ID del cliente.

```

<script language="javascript">
    ....
function changeSTAttributes()
{
    if (typeof top.setCustomerName == 'function') {
        top.setCustomerName(<%=userRDB.getUserId()%>, '<%=customer_name%>');
        top.setShoppingCartItems(<%=shoppingCartItems%>);
        top.changeSTAttributes();
    }
}
</script>

```

En la página Desconexión de la tienda de ejemplo se incluye más código del cliente, que establece el nombre del cliente en un ID generado y restablece el número de artículos del carro de la compra a cero. La página de desconexión es UserLogoffRouter.jsp. El código del cliente es el siguiente:

```

<HTML>
<HEAD>
<SCRIPT language="javascript">
if (typeof parent.setCustomerName == 'function')
    parent.setCustomerName (parent.WCSGUESTID, '')
    if (typeof parent.setShoppingCartItems == 'function')
        parent.setShoppingCartItems(0);
</SCRIPT>
</HEAD>
</HTML>

```

Determinación de la página por la que navega el cliente

El servicio de atención al cliente también permite a los CSR determinar por qué página de la tienda están navegando actualmente los clientes. Las tiendas de ejemplo determinan las páginas en las que están los clientes, incluyendo el archivo CustomerCareHeaderSetup.jsp en el archivo de cabecera (HeaderDisplay.jsp). El código siguiente del archivo CustomerCareHeaderSetup.jsp obtiene la información de URL de la página y actualiza el applet del comprador:

```

<script language="javascript">
var PageName="";
var PersonalPage=false;
<%
    String pname = request.getRequestURI();
    int indpn = pname.lastIndexOf('/');
    indpn = pname.lastIndexOf('/', indpn-1);

    if(indpn != -1)
        pname = pname.substring(indpn+1);

String headerType = (String) request.getAttribute("liveHelpPageType");
if (headerType==null) headerType="";

// Determinar si se trata de una página personal o no
if (headerType.equals("personal"))
{
    %>
    if (typeof parent.setPageParams == 'function') {
        PersonalPage=true;
        parent.setPageParams('PERSONAL_URL', '<%=pname%>');
    }
    <%
}
else
{
    %>
if (typeof parent.setPageParams == 'function')
    parent.setPageParams(location.href, '%=pname%>');
    <%
}
}

```

```

}
%>
Pagename="<%=pname%>";
</script>

```

No debe permitir que un CSR vea las páginas de cliente que contienen información personal ya que es posible que el contenido que visualiza el CSR no sea el mismo que el que visualiza un cliente. Por ejemplo, es posible que un CSR no tenga acceso a una página de campaña, a una página que incluye un precio determinado por contrato o a una página que incluye el ID de usuario, por ejemplo, la página del listín. Estas páginas se deben marcar como personales para evitar confundir al CSR durante una sesión de conversación.

Para marcar las páginas como personales, es decir, como no disponibles para el CSR, las tiendas de ejemplo incluyen el código siguiente en la página de cabecera, justo antes de incluir el archivo CustomerCareHeaderSetup.jsp.

```

<flow:ifEnabled feature="customerCare">
<%
request.setAttribute("liveHelpPageType", "personal");
%>
</flow:ifEnabled>

```

Nota: Aunque un CSR no puede ver el contenido de una página marcada como personal utilizando el botón Ver página de cliente, el CSR puede ver el URL de dicha página.

Seguimiento del número de artículos del carro de la compra

El servicio de atención al cliente permite a los CSR realizar un seguimiento de la cantidad de artículos que un cliente tiene en el carro de la compra en cualquier momento. El código siguiente del archivo CustomerCareInformationSetup.jsp obtiene el número de elementos del carro de la compra:

```

<%
JSPHelper jhelper = new JSPHelper(request);
String storeId = jhelper.getParameter("storeId");
int shoppingCartItems = 0;
%>
<
jsp:useBean id="userRDB" class="com.ibm.commerce.user.beans.
UserRegistrationDataBean" scope="page">
<% DataBeanManager.activate(userRDB, request); %></jsp:useBean>

<%
.....
// es necesario seleccionar artículos de pedido
OrderListDataBean orderListBean = new OrderListDataBean();
orderListBean.setStoreId(new Integer(storeId));
orderListBean.setOrderStatus("P");
orderListBean.setUserId(cmdcontext.getUserId());
DataBeanManager.activate(orderListBean, request);
Vector pendingOrders = orderListBean.getOrders();
for (int k=0; k< pendingOrders.size(); k++) {
    OrderAccessBean next_order = (OrderAccessBean) pendingOrders.
elementAt(k);
    OrderDataBean orderBean = new OrderDataBean();
    orderBean.setOrderId(next_order.getOrderId());
    DataBeanManager.activate(orderBean, request);
    //Obtener artículos del pedido
    OrderItemDataBean [] orderItems = orderBean.getOrderItemDataBeans();
    for (int i = 0; ((orderItems != null) &&
(i < orderItems.length)); i++)
    {
        OrderItemDataBean orderItem = orderItems[i];

```

```

        shoppingCartItems += orderItem.getQuantityInEJBType().intValue();
    }
}
....
%>

```

La función JavaScript siguiente actualizará la información del carro en el applet Sametime:

```

<script language="javascript">
    ...
    function changeSTAttributes()
    {
        if (typeof top.setCustomerName == 'function') {
            top.setCustomerName(<%=userRDB.getUserId()%>,
                '<%=customer_name%>');
        }
        top.setShoppingCartItems(<%=shoppingCartItems%>);
        top.changeSTAttributes();
    }
}
</script>

```

Nota: Un CSR puede ver el contenido del carro de la compra utilizando el botón **Ver carro de la compra**. Para obtener más información, consulte la ayuda en línea de WebSphere Commerce.

Las tiendas de ejemplo determinan el número de artículos del carro de la compra añadiendo el código siguiente a las páginas anteriores:

- En primer lugar, se define una variable int
`int liveHelpShoppingCartItems= 0;`
- A continuación, se utiliza la siguiente línea de código para agregar la cantidad a `liveHelpShoppingCartItems` siempre que se añada un `orderitem` al carro de la compra:
`liveHelpShoppingCartItems+= orderItem.getQuantityInEJBType().intValue();`
- A continuación, se añade el código siguiente al final de la página para establecer el nombre del cliente en el ID del comprador invitado y para obtener el número de artículos del carro de la compra del cliente.

```

<script language="javascript">
    if (typeof parent.setShoppingCartItems == 'function')
        parent.setShoppingCartItems(<%=liveHelpShoppingCartItems%>);
</script>

```

El código siguiente se utiliza en la página de carro de la compra vacío y en la página de confirmación de pedido para restablecer el número de artículos del carro a cero:

```

<script language="javascript">
    if (typeof parent.setShoppingCartItems == 'function')
        parent.setShoppingCartItems(0);
</script>

```

Seguimiento de los elementos de supervisión personalizados

Para realizar un seguimiento de los elementos de supervisión personalizados, efectúe lo siguiente:

1. Defina un ID de atributo de supervisión (los ID de atributo inferiores a 8000 están reservados). Para obtener más información, consulte el apartado “Definición de la lista de supervisión de la tienda” en la página 435.

2. Defina una variable JavaScript en el archivo Sametime.js para guardar el ID de atributo y el valor y permitir el acceso desde todos los marcos. Por ejemplo: `var myTrackAttributeId=8001; var myTrackAttributeValue="anything";`
3. Añada el código JavaScript para asignar o actualizar el valor de esta variable en las páginas de tienda. Por ejemplo `top.myTrackAttributeValue="new Value";`
4. Llame al método `setWcsAttribute()` del applet para actualizar el atributo en la función `changeSTAttributes()` del archivo Sametime.js:

```
function changeSTAttributes()
{
    if(CustomerAppletIsUp) {
        sametime.document.applets["InteractivePresenceApplet"].
setWcsAttribute(myTrackAttributeId,myTrackAttributeValue);
        sametime.document.applets["InteractivePresenceApplet"].
changeWCSAttrs();
    }
    else
        setTimeout("changeSTAttributes()",3000);
// esperar 3 segundos e intentarlo de nuevo
}
```

Nota: Inserte el código antes del método `changeWCSAttrs()`.

Envío de peticiones directamente a una cola de atención al cliente

Por omisión, cuando un cliente solicita una conversación con el CSR, la petición del cliente se coloca en la cola por omisión (`queueId=0`). A continuación, es responsabilidad del CSR dirigir la petición a una cola más específica. No obstante, también puede personalizar la página de tienda para enviar directamente las peticiones a las colas de atención al cliente definidas. Para personalizar las páginas de la tienda de modo que envíen peticiones a una cola definida, haga lo siguiente:

1. Cree las colas de atención al cliente para la tienda utilizando WebSphere Commerce Accelerator. Para obtener más información, consulte la Ayuda en línea a la producción de WebSphere Commerce.
2. Asigne estas colas a los CSR. Para obtener más información, consulte la Ayuda en línea a la producción de WebSphere Commerce.
3. Guarde un registro de los ID de cola de las colas que ha creado. Por ejemplo, el ID de cola es 10020.
4. Utilice JavaScript para actualizar la variable `reqQueue` en los JSP de la tienda. Por ejemplo, `top.reqQueue="10020";`.

La información de cola no se actualizará automáticamente después de que un CSR inicie el applet del CSR. Si se añade una cola al sistema, o si se modifica una cola existente, los cambios no entrarán en vigor hasta que el CSR inicie otra vez el applet de CSR.

Si un applet de CSR no reconoce el ID de cola, este ID de cola se volverá a colocar en la cola por omisión del CSR. Este comportamiento no afectará el valor de atributo de cola asociado al cliente.

Personalización de la atención al cliente

Tiene las opciones siguientes para personalizar la atención al cliente:

- Personalización de los parámetros del applet
- Personalización de los mensajes de la tienda

Personalización de los parámetros del applet

La tabla siguiente lista los parámetros del applet personalizables, que le permiten personalizar el servicio de atención al cliente modificando el archivo CustomerCareFrameSetup.jsp.

Tabla 23.

Nombre del parámetro personalizable	Descripción	Nota
CHAT_FONT_SIZE	El tamaño de font que se ha de utilizar en CharArea	El valor por omisión es 12.
CHAT_FONT_COLOR	El color del font que se va a utilizar para los mensajes de entrada del área de conversación.	El valor por omisión es azul (#0000FF).
CHAT_NAME_LENGTH	La longitud de los caracteres que se invierten para visualizar los nombres de usuario en el área de conversación.	El valor por omisión es 15.
WAIT_RANGE_1	El valor es un entero, si el número de clientes que están esperando en la tienda es menor que este valor, se muestra el mensaje de espera 1. De lo contrario se muestra el mensaje de espera correspondiente al valor WAIT_RANGE_2.	Utilice el valor -1 cuando se visualizará únicamente el mensaje 1.
WAIT_RANGE_2	El valor es un entero, si el número de clientes que están esperando en la tienda es menor que este valor pero mayor que WAIT_RANGE_1, se muestra el mensaje de espera 2. De lo contrario se muestra el mensaje de espera correspondiente al valor WAIT_RANGE_3.	Se ignora si el valor de WAIT_RANGE_1 es -1. Utilice el valor -1 para inhabilitar este rango.
WAIT_RANGE_3	El valor es un entero, si el número de clientes que están esperando en la tienda es menor que este valor pero mayor que WAIT_RANGE_2, se muestra el mensaje de espera 4.	Se ignora si el valor de WAIT_RANGE_2 es -1. Utilice el valor -1 para inhabilitar este rango.
contentFrame	El nombre del marco que se utiliza para las páginas de tienda de WebSphere Commerce regulares.	El valor por omisión es "main"
COUNTER_UNIT_PAGE	El valor entero que indica la frecuencia con que el contador aumentará en 1.	El valor por omisión es de 30 segundos. Asegúrese de que el valor sea el mismo que el que se ha definido en el archivo CustomerCare MonitorList.jsp de la tienda.

Tabla 23. (continuación)

COUNTER_UNIT_SITE	El valor entero que indica la frecuencia con que el contador del sitio aumentará en 1.	El valor por omisión es de 30 segundos. Asegúrese de que el valor sea el mismo que el que se ha definido en el archivo CustomerCareMonitorList.jsp de la tienda.
COUNTER_UNIT_WAIT	El valor entero que indica la frecuencia con que el contador de espera aumentará en 1.	El valor por omisión es de 30 segundos. Asegúrese de que el valor sea el mismo que el que se ha definido en el archivo CustomerCareMonitorList.jsp de la tienda.
WIDTH	La anchura preferida del marco de conversación en píxeles.	El valor por omisión es de 360 píxeles. La longitud del mensaje de invitación afectará a la anchura final.
HEIGHT	La altura del marco de conversación en píxeles.	El valor por omisión es de 400 píxeles.

Personalización de los mensajes de la tienda

Los mensajes que se muestran a un cliente cuando éste se conecta inicialmente a un CSR, por ejemplo "Hola, ¿en qué puedo ayudarle?" o "Nuestro horario de oficina es de 9 a 21", se almacenan en el archivo de propiedades del servidor Sametime. Los archivos de propiedades se dividen en dos tipos de archivos: Customer.properties y Agent.properties. El archivo Customer.properties contiene mensajes que se muestran al cliente, mientras que el archivo Agent.properties contiene información que se muestra al CSR. Estos dos archivos también tienen archivos correspondientes específicos del entorno nacional, por ejemplo Customer_de_DE.properties y Agent_de_DE.properties, para cada entorno nacional instalado en la instancia de WebSphere Commerce.

Para cambiar los mensajes de estos archivos, realice lo siguiente:

1. Localice los archivos de propiedades en el servidor Sametime. Por omisión, estos archivos de propiedades están ubicados en el directorio siguiente:
 - *dir_instalación_atención_cliente*\properties
2. Realice los cambios necesarios. La tabla siguiente lista las claves de mensajes para cada mensaje.

Tabla 24.

Clave del mensaje	Descripción	Notas
WelcomeMessage	El primer mensaje que se va a visualizar en el applet del cliente después de que el CSR acepta una petición de mensaje.	
GoodbyeMessage	El mensaje se visualiza en el applet del cliente justo después de que el CSR o el cliente finalicen la sesión de conversación.	

Tabla 24. (continuación)

PushPageMessage	El mensaje que ha de visualizar en el applet del cliente después de que el CSR envíe un URL al navegador del cliente.	
CallCSRMessage	El mensaje que se ha de visualizar cuando un cliente envía una petición de ayuda y el applet se conecta al servidor Sametime.	
NoConnectionMessage	El mensaje que se ha de visualizar cuando el applet del cliente no puede conectarse con el servidor Sametime o si ha perdido la conexión con el servidor Sametime.	
StoreCloseMessage	El mensaje que se ha de visualizar cuando los CSR no están disponibles para dar servicio al cliente.	Siempre se visualiza con el mensaje StoreWorkingHour.
StoreWorkingHour	El mensaje que se ha de visualizar cuando los CSR no están disponibles para dar servicio al cliente. Describe las horas de funcionamiento de la tienda.	Siempre se visualiza con el mensaje StoreCloseMessage.
WaitingMessage	El mensaje que se ha de visualizar cuando el cliente envía una petición y el número total de clientes que esperan es menor que WAIT_RANGE_1.	
WaitingMessage1	El mensaje que se ha de visualizar cuando un cliente envía una petición y el número total de clientes que esperan es menor que WAIT_RANGE_2 pero mayor que WAIT_RANGE_1.	
WaitingMessage2	El mensaje que se ha de visualizar cuando un cliente envía una petición y el número total de clientes que esperan es menor que WAIT_RANGE_3 pero mayor que WAIT_RANGE_2.	
WaitingMessage3	El mensaje que se ha de visualizar cuando un cliente envía una petición y el número total de clientes que esperan es mayor que WAIT_RANGE_3.	

Tabla 24. (continuación)

CIWarningLabelLineOne	La primera línea del mensaje de invitación que se visualiza cuando un CSR inicia una petición de conversación con un cliente.	La longitud de visualización del mensaje afectará a la anchura del marco de conversación. El carácter separador de línea no debe utilizarse.
CIWarningLabelLineTwo	La segunda línea del mensaje de invitación que se visualiza cuando un CSR inicia una petición de conversación con un cliente.	La longitud de visualización del mensaje afectará a la anchura del marco de conversación. El carácter separador de línea no debe utilizarse.

Nota: Para iniciar una línea nueva, debe insertarse el carácter separador de línea "\n". De no ser así, el mensaje puede superar los límites del área de visualización.

Nota: Es posible que necesite modificar las propiedades de los diferentes entornos nacionales para asegurarse de que el mensaje esté correctamente traducido.

3. Cierre el archivo y guárdelo.

Adición del servicio de atención al cliente a la tienda

Para añadir el servicio de atención al cliente a una tienda que no esté basada en un ejemplo, efectúe lo siguiente:

Parte 1: Instalación de requisitos previos

Para que el servicio de atención al cliente funcione en la tienda, deberá realizar lo siguiente:

- Instalar un servidor Sametime. Para obtener más información, consulte la publicación *WebSphere Commerce, Guía de software adicional*.
- Instalar el paquete de integración de Sametime de WebSphere Commerce. Para obtener más información, consulte la publicación *WebSphere Commerce, Guía de software adicional*.
- Detener la instancia de WebSphere Commerce, habilitar Sametime en el Gestor de configuración y, a continuación, reiniciar la instancia. Para obtener más información, consulte la publicación *WebSphere Commerce, Guía de software adicional*.
- Si el servidor Sametime no utiliza el mismo servidor LDAP que WebSphere Commerce, cree un CSR y regístrelo para atención al cliente mediante la Consola de administración. Para obtener más información, consulte la Ayuda en línea a la producción de WebSphere Commerce.

Parte 2: Copia de los archivos de integración del servicio de atención al cliente de la tienda de ejemplo

Las tiendas de ejemplo FashionFlow y ToolTech incluyen los archivos siguientes que se utilizan para integrar el servicio de atención al cliente a la tienda:

- `Sametime.js`: Contiene las funciones JavaScript que se incluyen para todos los marcos. Se llama a las funciones de este archivo con un prefijo padre de las páginas del marco principal, por ejemplo, `parent.setCustomerName`.

- `CustomerCareBlankSetup.jsp`: Un archivo JSP vacío. Lo llama el mandato `CCShopperBlankPageView` como contenedor para los marcos.
- `CustomerCareFrameSetup.jsp`: Contiene las funciones JavaScript e intercala el applet de Atención al cliente para el escaparate. Lo llama el mandato `CCShopperFramePageView` y carga el applet del cliente en el conjunto de marcos.
- `CustomerCareAppletReadySetup.jsp`: Indica que el applet se ha cargado correctamente. Lo llama el mandato `CCShopperReadyPageView` e indica que el applet del cliente está preparado para las funciones de Javascript.
- `CustomerCareHeaderSetup.jsp`: Un archivo de cabecera que pasa un parámetro al applet indicando si la página que incluye esta cabecera es una página personal o no. Debe incluirse en cada página para actualizar el URL de la página del cliente.
- `CustomerCareInformationSetup.jsp`: Actualiza la información de nombre y el ID del cliente. Lo llama el mandato `CCShopperInfoUpdatePageView`.
- `CustomerCareMonitorList.jsp`: El archivo de configuración de la lista de supervisión. Lo llama el CSR para cargar la lista de supervisión.
- `CustomerCareStoreQuestionList.jsp`: El archivo de configuración de la lista de temas. Lo llama el CSR para cargar la lista de temas de la tienda.
- `CustomerCareStoreURLList.jsp`: El archivo de configuración de la lista de URL a nivel de tienda. Lo llama el CSR para cargar la lista de URL de la tienda.
- `CustomerCareChatSetup.jsp`: El archivo de configuración para utilizar la atención al cliente sin un conjunto de marcos. Lo llama el mandato `CCChatPageView` e inicia el applet del cliente en una configuración sin conjunto de marcos.
- `EnvironmentSetup.jsp`: Lo utilizan todos los archivos JSP de atención al cliente para la configuración a nivel de tienda (por ejemplo, para paquetes de recursos).
- `index.jsp`: La página de entrada que redirige el navegador del cliente a la página del conjunto de marcos o las páginas regulares del catálogo según los valores de flujo de tienda de atención al cliente.
- `StoreFramesetPage.jsp`: Lo llama el mandato `StoreFramesetView` para crear un conjunto de marcos para atención al cliente.

Para copiar los archivos de integración Sametime de la tienda de ejemplo a su tienda, efectúe lo siguiente:

1. Localice el archivador de tienda para la tienda Directo al consumidor (tienda FashionFlow) o Directo a B2B (ToolTech). Los archivos del archivador de tienda están situados en el directorio siguiente:
 - `dir_instalación_WC/samplestores`
2. Abra la carpeta `ConsumerDirect` o `B2BDirect`, luego seleccione `consumerdirect.sar` o `B2Bdirect.sar`.
3. Abra el archivador de tienda utilizando WinZip o una herramienta similar.
4. Seleccione los archivos listados anteriormente:
5. Extraiga los archivos en el directorio que contiene los elementos web de la tienda. Para mantener la misma estructura de directorio que las tiendas de ejemplo, es posible que desee crear los subdirectorios para los archivos siguientes:
 - `.../CustomerServiceArea/CollaborationSection`
 - `CustomerCareAppletReadySetup.jsp`
 - `CustomerCareBlankSetup.jsp`
 - `CustomerCareBlankSetup.jsp`

- CustomerCareFrameSetup.jsp
- CustomerCareInformationSetup.jsp
- CustomerCareMonitorList.jsp
- CustomerCareStoreQuestionList.jsp
- CustomerCareStoreURLList.jsp
- CustomerCareChatSetup.jsp
- /include
 - CustomerCareHeaderSetup.jsp
 - EnvironmentSetup.jsp
- ..\
 - index.jsp
 - Sametime.js
 - StoreFramesetPage.jsp

Parte 3: Adición de código para determinar qué página está examinando el cliente

Para determinar por qué página está navegando el cliente, efectúe lo siguiente:

1. Incluya el archivo CustomerCareHeaderSetup.jsp en el archivo de cabecera de la tienda, por ejemplo:

```
<%@ include file="include\CustomerCareHeaderSetup.jsp" %>
```

2. Añada el código siguiente a cualquier página que deba marcarse como personal y, de este modo, no estará disponible para que acceda el CSR. Asegúrese de que ha añadido el código siguiente antes de incluir el archivo CustomerCareHeaderSetup.jsp.

```
<flow:ifEnabled feature="customerCare">
<%
request.setAttribute("liveHelpPageType", "personal");
%>
</flow:ifEnabled>=incfile%>"flush="true"/>
```

Parte 4: Adición de un enlace al servicio de atención al cliente

Para que los clientes puedan acceder al servicio de atención al cliente, efectúe lo siguiente:

1. Determine dónde desea colocar el enlace con el servicio de atención al cliente. Por ejemplo, es posible que desee colocar el enlace en una barra de navegación, de modo que los clientes puedan verlo siempre o en determinadas páginas de la tienda.
2. Copie el código siguiente en las páginas que contendrán el enlace:

Nota: Es posible que tenga que sustituir infashiontext por el nombre de objeto que se utiliza para dicho recurso de tienda.

```
<a href="javascript:if((parent.sametime != null))
top.interact();"><%=infashiontext.getString("LiveHelp")%> </a>
```

Parte 5: Creación de una página de entrada que redirigirá la página del conjunto de marcos de atención al cliente

Dado que para que la mayor parte de las características de atención al cliente funcione correctamente se necesita el conjunto de marcos, el cliente debe llamar al

mandato `StoreFramesetView` para activar el conjunto de marcos. Para obtener un ejemplo, consulte el archivo `index.jsp` de la tienda de ejemplo Directo al consumidor o Directo a B2B.

Capítulo 42. Adición de zonas de e-marketing a la tienda

Las zonas de e-marketing reservan espacio en las páginas de tienda en las que se visualiza el contenido de marketing personalizado para iniciativas de campaña. Cuando un cliente solicita una página, las zonas de e-marketing presentes en la página se comunican con el servidor de normas para procesar el código basado en normas asociado con la zona. Cada zona de e-marketing tiene una o más iniciativas de campaña asociadas a ella. Para obtener más información sobre campañas e iniciativas de campaña, consulte el Capítulo 20, "Elementos de campaña", en la página 223 y la ayuda en línea de WebSphere Commerce.

Para que las iniciativas de campaña se visualicen correctamente en las páginas de la tienda, debe añadirse una zona de e-marketing al archivo JSP y, a continuación, registrarla en la base de datos utilizando WebSphere Commerce Accelerator. Este capítulo describe cómo añadir zonas de e-marketing a los archivos JSP de la tienda. Para obtener más información sobre cómo registrar la zona de e-marketing en la base de datos utilizando WebSphere Commerce Accelerator, consulte la ayuda en línea de WebSphere Commerce.

Zona de e-marketing

A continuación se muestra un ejemplo de una zona de e-marketing.

```
<!-- =====
/**-----
/** El ejemplo siguiente se proporciona "TAL CUAL".
/**
/** IBM lo ofrece como un simple ejemplo y no se ha probado exhaustivamente
/** bajo todas las condiciones. Por tanto, IBM no puede garantizar su
/** fiabilidad, operatividad o funcionalidad.
/**
/** Este ejemplo puede incluir los nombres de individuos, empresas, marcas
/** y productos para ilustrar los conceptos de la forma más completa posible.
/** son ficticios y cualquier similitud con nombres
/** y direcciones utilizados por personas reales
/** o empresas es pura coincidencia.
/**-----
/**
===== -->
<%
/**
* INICIO - el código siguiente debe existir sólo una vez por página, inicializa
* el contexto de mandatos y el bean de datos de tienda.
*/

// crear el bean de tienda para obtener el directorio de tienda
String collateralPath = "/webapp/wcs/stores/";
com.ibm.commerce.command.CommandContext emsCommandContext =
    (com.ibm.commerce.command.CommandContext)
    request.getAttribute(com.ibm.commerce.server.ECConstants.EC_COMMANDCONTEXT);
com.ibm.commerce.common.beans.StoreDataBean storeDataBean =
    new com.ibm.commerce.common.beans.StoreDataBean();
storeDataBean.setStoreId(emsCommandContext.getStoreId().toString());
com.ibm.commerce.beans.DataBeanManager.activate(storeDataBean, request);
if (storeDataBean.getDirectory() != null) {
    collateralPath += storeDataBean.getDirectory() + "/";
}

/**
* INICIO - el código siguiente debe existir sólo una vez por página, inicializa
* el contexto de mandatos y el bean de datos de tienda.
*/
%>

<!-- =====
// El formulario HTML siguiente somete de petición de la zona de e-Marketing al mandato ClickInfo
```

```

// que captura las estadísticas de campaña y redirige a la ubicación
// especificada por el parámetro URL.
===== -->
<form name="storeEmsForm" method="POST" action="/webapp/wcs/stores/servlet/ClickInfo">
  <input type="hidden" name="evtype">
  <input type="hidden" name="mpe_id">
  <input type="hidden" name="intv_id">
  <input type="hidden" name="URL">
</form>

<%
  /**
   * INICIO - el código siguiente se puede utilizar para descargar varias zonas de e-marketing
   *          en la página. Personalice el nombre de la instancia de EMarketingSpot adecuada y
   *          la zona de e-Marketing antes de su uso. Duplique este código si necesita más de 1
   *          zona, pero no utilice el mismo nombre de zona.
   */

  // crear la zona de e-marketing
  com.ibm.commerce.marketing.beans.EMarketingSpot eMarketingSpot =
    new com.ibm.commerce.marketing.beans.EMarketingSpot();

  // IMPORTANTE - establezca aquí el nombre correcto
  String emsName = request.getParameter("emsName");
  if (emsName == null) {
    emsName = "defaultEMSName";
  }
  eMarketingSpot.setName(emsName);

  // Establezca el ID de entrada de catálogo que se está visualizando en la página. Esto es
  // necesario si la iniciativa de venta ascendente/venta cruzada está basada en el contenido
  // de la página actual.
  String sourceCatentryId = request.getParameter("sourceCatentryId");
  if (sourceCatentryId != null) {
    eMarketingSpot.setSourceCatalogEntryId(sourceCatentryId); // use this method to set single ID
    //eMarketingSpot.setMultipleSourceCatalogEntryId(sourceCatentryId2);
    // utilice este método para definir varios ID
  }

  // el número máximo de producto/categorías/anuncios que se visualizan mediante esta
  // zona de e-Marketing Spot se puede establecer aquí
  eMarketingSpot.setMaximumNumberOfCatalogEntries(20);
  eMarketingSpot.setMaximumNumberOfCategories(20);
  eMarketingSpot.setMaximumNumberOfCollateral(20);
  eMarketingSpot.setMaximumNumberOfAssociateCatalogEntries(20);

  // crear instancia de bean
  com.ibm.commerce.beans.DataBeanManager.activate(eMarketingSpot, request);
%>

<%
  // El bloque siguiente se utiliza para visualizar los productos de venta ascendente/venta
  // cruzada asociados a esta zona de e-marketing. Se hará referencia a la página de
  // visualización de productos que muestra el producto seleccionado en la campaña enviando
  // el formulario HTML adjunto anterior.

  if (eMarketingSpot.getAssociateCatalogEntries() != null
    && eMarketingSpot.getAssociateCatalogEntries().length < 0) {
%>
<TABLE>
  <%
    for (int i=0; i<eMarketingSpot.getAssociateCatalogEntries().length; i++) {
      String associateCatalogEntryThumbNail = null;
      String associateCatalogEntryShortDescription = null;
      try {
        associateCatalogEntryThumbNail =
          eMarketingSpot.getAssociateCatalogEntries()[i].getDescription(emsCommandContext.
            getLanguageId()).getThumbNail();
        associateCatalogEntryShortDescription =
          eMarketingSpot.getAssociateCatalogEntries()[i].getDescription(emsCommandContext.
            getLanguageId()).getShortDescription();
      }
      catch (Exception e) {
        // no se ha definido ninguna descripción para el idioma actual
      }
%>
<TR>
  <TD>
    <A HREF="/webapp/wcs/stores/servlet/ClickInfo?evtype=CpgnClick&mpe_id=<%=
      eMarketingSpot.getId() %>&intv_id=<%=

```

```

eMarketingSpot.getAssociateCatalogEntries()[i].getInitiativeId() %>
&URL=/webapp/wcs/stores/servlet/ProductDisplay&<%=
com.ibm.commerce.server.ECConstants.EC_STORE_ID %>=<%=
emsCommandContext.getStoreId().toString() %>&<%=
com.ibm.commerce.server.ECConstants.EC_PRODUCT_ID %>=<%=
eMarketingSpot.getAssociateCatalogEntries()[i].getCatalogEntryID() %>&<%=
com.ibm.commerce.server.ECConstants.EC_LANGUAGE_ID %>=<%=
emsCommandContext.getLanguageId().toString() %>"
<IMG SRC="<%= collateralPath + associateCatalogEntryThumbnail %>"
ALT="<%= associateCatalogEntryShortDescription %>" BORDER=0 WIDTH=60>
</A>
</TD>
<TD><%= associateCatalogEntryShortDescription %></TD>
</TR>
<% } %>
</TABLE>
<% } %>

<%
// El bloque siguiente se utiliza para visualizar los anuncios asociados con esta
// zona de e-marketing. Se puede hacer referencia al enlace de URL definido con un anuncio
// enviando el formulario HTML adjunto anterior.

if (eMarketingSpot.getCollateral() != null && eMarketingSpot.getCollateral().length > 0) {
%>
<TABLE>
<% for (int i=0; i<eMarketingSpot.getCollateral().length; i++) { %>
<TR>
<% if (eMarketingSpot.getCollateral()[i].getTypeName().equals("Image")) { %>
<TD>
<A HREF="javascript:document.storeEmsForm.evtype.value='CpgnClick';
document.storeEmsForm.mpe_id.value='<%= eMarketingSpot.getId()
%>';document.storeEmsForm.intv_id.value='<%=
eMarketingSpot.getCollateral()[i].getInitiativeId()
%>';document.storeEmsForm.URL.value='<%=
eMarketingSpot.getCollateral()[i].getUrlLink()
%>';document.storeEmsForm.submit();">
<IMG SRC="<%= collateralPath + eMarketingSpot.getCollateral()[i].getLocation() %>">
</A>
</TD>
<TD>
<%= eMarketingSpot.getCollateral()[i].getMarketingText() %>
</TD>
<% } else if (eMarketingSpot.getCollateral()[i].getTypeName().equals("Flash")) { %>
<TD>
<EMBED src="<%= collateralPath + eMarketingSpot.getCollateral()[i].getLocation()
%>" quality=high bgcolor=#FFFFFF WIDTH=120 HEIGHT=90
TYPE="application/x-shockwave-flash"></EMBED>
</TD>
<TD>
<A HREF="javascript:document.storeEmsForm.evtype.value='CpgnClick';
document.storeEmsForm.mpe_id.value='<%= eMarketingSpot.getId()
%>';document.storeEmsForm.intv_id.value='<%=
eMarketingSpot.getCollateral()[i].getInitiativeId()
%>';document.storeEmsForm.URL.value='<%=
eMarketingSpot.getCollateral()[i].getUrlLink()
%>';document.storeEmsForm.submit();">
<%= eMarketingSpot.getCollateral()[i].getMarketingText() %>
</A>
</TD>
<% } %>
</TR>
<% } %>
</TABLE>
<% } %>

<%
// El bloque siguiente se utiliza para visualizar las categorías asociadas a esta zona de
// e-marketing. Se hará referencia a la página de visualización de categorías que muestra
// la categoría seleccionada en la campaña enviando el formulario HTML adjunto anterior.

if (eMarketingSpot.getCategories() != null && eMarketingSpot.getCategories().length > 0) {
%>
<TABLE>
<%
for (int i=0; i<eMarketingSpot.getCategories().length; i++) {
String catalogGroupName = null;
String catalogGroupLongDescription = null;
try {
catalogGroupName = eMarketingSpot.getCategories()[i].getDescription

```

```

        (emsCommandContext.getLanguageId()).getName();
        catalogGroupLongDescription = eMarketingSpot.getCategories()[i].getDescription(
            emsCommandContext.getLanguageId()).getLongDescription();
    }
    catch (Exception e) {
        // no se ha definido ninguna descripción para el idioma actual
    }
}
%>
<TR>
<TD>
<A HREF="/webapp/wcs/stores/servlet/ClickInfo?evtype=CpgnClick&mpe_id=<%=
eMarketingSpot.getId() %>&intv_id=<%=
eMarketingSpot.getCategories()[i].getInitiativeId()
%>&URL=/webapp/wcs/stores/servlet/CategoryDisplay&<%=
com.ibm.commerce.server.ECConstants.EC_STORE_ID %>=<%=
emsCommandContext.getStoreId().toString() %>&<%=
com.ibm.commerce.server.ECConstants.EC_CATEGORY_ID %>=<%=
eMarketingSpot.getCategories()[i].getCategoryId() %>&<%=
com.ibm.commerce.server.ECConstants.EC_CATALOG_ID %>=<%=
eMarketingSpot.getCategories()[i].getCatalogId() %>&<%=
com.ibm.commerce.server.ECConstants.EC_LANGUAGE_ID %>=<%=
emsCommandContext.getLanguageId().toString() %>">
<%= catalogGroupName %>
</A>
</TD>
<TD><%= catalogGroupLongDescription %></TD>
</TR>
<% } %>
</TABLE>
<% } %>

<%
// El bloque siguiente se utiliza para visualizar los productos asociados a esta zona de
// e-Marketing. Se hará referencia a la página de visualización de productos que muestra
// el producto seleccionado en la campaña enviando el formulario HTML adjunto anterior.

if (eMarketingSpot.getCatalogEntries() != null &&eMarketingSpot.getCatalogEntries().length > 0) {
%>
<TABLE>
<%
for (int i=0; i<eMarketingSpot.getCatalogEntries().length; i++) {
    String catalogEntryThumbNail = null;
    String catalogEntryShortDescription = null;
    try {
        catalogEntryThumbNail =
            eMarketingSpot.getCatalogEntries()[i].getDescription(emsCommandContext.
                getLanguageId()).getThumbNail();
        catalogEntryShortDescription =
            eMarketingSpot.getCatalogEntries()[i].getDescription(emsCommandContext.
                getLanguageId()).getShortDescription();
    }
    catch (Exception e) {
        // no se ha definido ninguna descripción para el idioma actual
    }
}
%>
<TR>
<TD>
<A HREF="/webapp/wcs/stores/servlet/ClickInfo?evtype=CpgnClick&mpe_id=<%=
eMarketingSpot.getId() %>&intv_id=<%=
eMarketingSpot.getCatalogEntries()[i].getInitiativeId()
%>&URL=/webapp/wcs/stores/servlet/ProductDisplay&<%=
com.ibm.commerce.server.ECConstants.EC_STORE_ID %>=<%=
emsCommandContext.getStoreId().toString() %>&<%=
com.ibm.commerce.server.ECConstants.EC_PRODUCT_ID %>=<%=
eMarketingSpot.getCatalogEntries()[i].getCatalogEntryID()
%>&<%= com.ibm.commerce.server.ECConstants.EC_LANGUAGE_ID %>=<%=
emsCommandContext.getLanguageId().toString() %>">
<IMG SRC="<%= collateralPath + catalogEntryThumbNail %>"
    ALT="<%= catalogEntryShortDescription %>" BORDER=0 WIDTH=60>
</A>
</TD>
<TD><%= catalogEntryShortDescription %></TD>
</TR>
<% } %>
</TABLE>
<% } %>

<%
/**
* FIN - el código siguiente se utiliza para desactivar varias zonas de e-Marketing en la página.

```


* Antes de utilizarlo personalice el nombre de la zona de e-Marketing.
* Duplique este código si necesita más de 1 zona, pero no utilice el mismo nombre de zona.
*/
>

La zona de e-marketing anterior soporta cuatro tipos de iniciativas de campaña:

- Recomendación de productos
- Recomendación de categorías
- Anuncios publicitarios informativos
- Asociación de comercialización

Nota: Para obtener información más detallada sobre cada una de estas iniciativas, consulte el Capítulo 20, “Elementos de campaña”, en la página 223.

Bean e-MarketingSpot

Las zonas de e-marketing utilizan el bean e-MarketingSpot para devolver el resultado de las iniciativas de campaña que están planificadas actualmente en la zona. El uso de distintas propiedades del bean le permite personalizar la zona de e-marketing y la iniciativa de campaña correspondiente. Para obtener más información sobre el bean e-MarketingSpot y sus propiedades, consulte la ayuda en línea de WebSphere Commerce.

Adición de una zona de e-marketing a las páginas de la tienda

Para añadir una zona de e-marketing a las páginas de la tienda, haga lo siguiente:

1. Determine en qué archivos JSP se visualizará la zona. La zona se puede añadir a varios archivos JSP.
2. Determine en qué parte del archivo JSP colocará la zona.
3. Copie la zona de e-Marketing de ejemplo del apartado “Zona de e-marketing” en la página 455 en un archivo JPS nuevo dentro de la aplicación Web de la tienda.
4. Personalice la zona de e-marketing de ejemplo para que se adapte al diseño de su(s) archivo(s) JSP.
5. En el código de la zona de e-marketing, ponga un nombre a la zona de e-marketing.

Nota: Los nombres de las zonas de e-marketing deben ser descriptivos, de forma que incluyan su ubicación, por ejemplo AnuncioPáginaPresentación o RecomendaciónPáginaPasarPorCaja. Esto ayuda a reducir la confusión acerca de dónde aparecerá y del contenido que deberá tener. Si es necesario, se pueden añadir números al nombre para establecer diferencias entre dos Zonas de e-marketing que aparecen en la misma página. Los nombres de zona de e-marketing deben ser identificadores válidos de Java. Debe utilizar este mismo nombre al registrar la zona de e-marketing en la base de datos utilizando WebSphere Commerce Accelerator.

6. Añada la zona de e-Marketing en el archivo JSP incluyendo dinámicamente la zona mediante el código `<jsp:include>`.
7. Si necesita más de una zona de e-Marketing por archivo JSP, repita los pasos 2 a 6.
8. Registre la zona de e-marketing en la base de datos utilizando WebSphere Commerce Accelerator. Para obtener instrucciones más detalladas, consulte la ayuda en línea de WebSphere Commerce.

Nota:

- a. Si piensa añadir el ID de tienda, el ID de catálogo o el ID de idioma al URL utilizando el siguiente convenio, "langId=<%= languageId %>", tenga en cuenta que el archivo JSP en el que se incluye la zona de e-marketing debe poner a disposición el ID adecuado. Los ID también pueden recuperarse a través del contexto de mandatos, por ejemplo, `getContext().getLanguageId()`.
- b. El parámetro de URL, `catalogDisplay` debe empezar con "&" en lugar de "?" porque el código no hace referencia al mandato directamente.
- c. Utilizando la dinámica Java incluya el código para añadir la zona de e-Marketing en el archivo JSP de la tienda, se puede habilitar la nueva característica `Dynacache`, de modo que el contenido del archivo JSP, excluida la zona de e-Marketing, se almacene en la antememoria. La zona de e-Marketing se renovará en cada visita y, de este modo, visualizará el contenido dinámico correcto. El siguiente es un ejemplo de cómo se puede utilizar la dinámica incluye para añadir una zona de e-Marketing a un archivo JSP denominado `ESpot.jsp`:

```
<jsp:include page="ESpot.jsp" flush="true">
<jsp:param name="emsName" value="StoreHomePage" />
</jsp:include>
```

- d. Si se está planificando una iniciativa de asociación de comercialización en la zona de e-Marketing y el fuente de la asociación está basado en el contenido de la página, se puede establecer utilizando los métodos `setSourceCatalogEntryId(String source)` y `setMultipleCatalogEntryId(String source)` que se proporcionan en `com.ibm.commerce.marketing.beans.EMarketingSpot`. Por ejemplo, en la página de Visualización de productos, si el producto que aparece en la página se va a utilizar como fuente de la asociación, se invocará el método siguiente:

```
eSpot.setSourceCatalogEntryId(productId)
```

donde *eSpot* es una instancia de la clase `com.ibm.commerce.marketing.beans.EMarketingSpot` y *productId* es el identificador del producto fuente.

Parte 12. Apéndices

Apéndice A. Descripción de UML

UML (Unified Modeling Language) es un lenguaje gráfico estándar para representar diferentes elementos de diseño de software. Los ejemplos siguientes son algunos de los elementos más comunes de UML. Para obtener más detalles sobre las especificaciones formales, consulte <http://www.rational.com> y <http://www.omg.org>.

Los diagramas UML constan de los elementos siguientes:

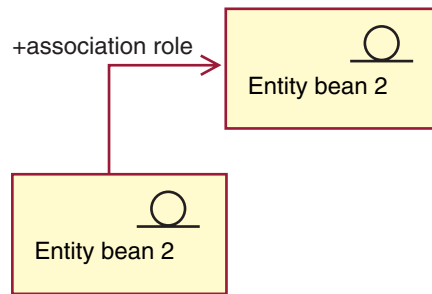
- **Recuadros:** Los recuadros representan clases de objetos. Los nombres de clase aparecen en la parte superior del recuadro. Los atributos, si se muestran, aparecen debajo del nombre de clase. El nombre de clase y los atributos están separados por una línea.
- **Líneas:** Las líneas representan las posibles relaciones entre objetos de dos clases. Los objetos de la clase de un extremo de la línea pueden "asociarse" con objetos de la otra clase.
- **Rombos sólidos:** Los rombos sólidos en el extremo de una línea indican contención por valor. Los objetos de la clase del otro extremo de la línea forman parte de un objeto, y sólo uno, de la clase a la que toca el rombo.
- **Rombos abiertos:** Los rombos abiertos en el extremo de una línea indican contención por referencia. Los objetos en el extremo de la línea donde hay el rombo pueden considerarse como objetos de agrupación de la clase del otro extremo de la línea.
- **Números de cardinalidad:** Aparecen en el extremo de las líneas de relación para indicar una restricción de cardinalidad. La tabla siguiente resume las restricciones de cardinales:

Número cardinal	Tipo de relación
1	uno y sólo uno
0..1	cero o uno
0..n	cero o más
1..n	uno o más

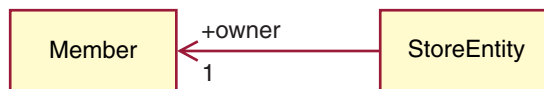
Si no se muestra una restricción de cardinales, se supone que la cardinalidad es 0..n, a menos que aparezca un rombo sólido en el extremo de una línea de relación. En tal caso, deberá ser 1.

- **Signos más:** Los signos más que aparecen en el extremo de las líneas de relación indican que el objeto de la clase del extremo de la línea desempeña un rol en la relación. El texto que va a continuación del signo más indica el rol del objeto en la relación.
- **Flechas:** Las flechas en el extremo de una línea de relación indican que la dirección de la relación entre dos objetos sigue la dirección de la flecha. La ausencia de flechas en una línea de relación indica que la dirección de la relación entre los objetos es normalmente en ambas direcciones.

Los diagramas siguientes ilustran los conceptos mencionados anteriormente:



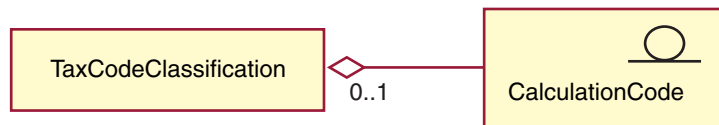
Este diagrama muestra dos beans de entidad con el símbolo estereotipo que indica un bean al que se denomina Enterprise Java Bean. Existe una asociación unidireccional del primer bean al segundo bean de entidad. El signo más va seguido de texto que describe qué rol desempeña el bean de Entidad 2 en la asociación.



En este diagrama, una Entidad de tienda tiene un solo y único propietario, que es un Miembro. Un Miembro puede tener cero o más Entidades de tienda. El signo más indica que el miembro desempeña un rol en la relación. En este caso, el Miembro es el propietario de la Entidad de tienda. La flecha indica que normalmente se puede encontrar el propietario de una entidad de tienda solicitando el propietario a la entidad de tienda y no solicitando a un miembro todas las entidades de tienda de las que es propietario.



En este diagrama, un Artículo de pedido siempre forma parte de un solo pedido. Un pedido tiene cero o más Artículos de pedido.



Este diagrama indica que un Código de cálculo se agrupa mediante cero o una Clasificación de códigos de impuestos y una Clasificación de códigos de impuestos agrupa cero o más Códigos de cálculo.

Apéndice B. Creación de los datos

Antes de crear los datos de la tienda con formato de archivo XML, efectúe lo siguiente:

- Revise la información del Capítulo 37, “Visión general de la carga de datos de tienda”, en la página 361.
- Determine el orden de la información que está creando. La información de cada uno de los capítulos de datos de la tienda le recomienda el orden en que debe crear los datos, pero cuando cree los archivos XML recuerde que la información de una tabla padre debe ser anterior a la información de una tabla hijo. Para obtener más información sobre el orden en que se cargan los elementos, consulte el Capítulo 38, “Carga de grupos de elementos de base de datos de WebSphere Commerce”, en la página 405.

Creación de datos para tiendas de ejemplo

Los datos de los archivadores de tienda de ejemplo tienen el formato de archivos XML válidos para Loader package. Los archivos XML del archivador de tienda están destinados a ser portables y no deben contener claves primarias generadas que sean específicas de una instancia determinada de la base de datos. En lugar de ello, utilizan alias internos que ID Resolver resuelve en el momento de la publicación. La utilización de estos convenios permite copiar y publicar los archivadores de tienda de ejemplo varias veces.



No es necesario que siga estos convenios cuando cree datos de tienda para su tienda con formato de archivos XML, a no ser que piense crear un archivador de tienda que se utilizará para generar varias tiendas o que desee crear un archivador de tienda portable, es decir un archivador de tienda que pueda publicarse en otra instancia de WebSphere Commerce.

Por lo tanto, los archivadores de tienda de ejemplo utilizan los convenios siguientes:

- @ como en `ffmcenter_id="@ffmcenter_id_1"`. El uso del símbolo @ se conoce como resolución de alias interno. ID Resolver es un programa de utilidad de Loader package que genera identificadores para elementos XML que los necesitan. Una de las técnicas que ID Resolver utiliza es la resolución de alias internos. Al utilizar la resolución de alias interno, la clave primaria (identificador) se sustituye por un alias en el documento XML. Este alias se utiliza luego en cualquier otra parte del archivo XML para hacer referencia a ese elemento. De este modo, no es necesario conocer los índices exclusivos necesarios para poder crear el archivo XML. Durante la publicación en la Consola de administración o utilizando Loader package, ID Resolver sustituye el símbolo @ por un valor exclusivo. Vea los ejemplos siguientes de un archivo XML:

– ID Resolver previo

```
<catalog
catalog_id="@catalog_id_1"
member_id="&MEMBER_ID;"
identifier=FashionFlow"
description="Catálogo de FashionFlow"
tpclevel="0"/>
```

– ID Resolver posterior

```
<catalog  
  catalog_id="10001"  
  member_id="-2000"  
  identifier="FashionFlow"  
  description="Catálogo de FashionFlow"  
  tpcllevel="0"/>
```

donde 10001 es el ID exclusivo asignado por ID Resolver y -2000 es el ID de miembro seleccionado por el usuario en la Consola de administración. El archivo XML resultante se carga utilizando Loader package. Si se ejecutan los archivos mediante ID Resolver, se asegura que se puedan crear muchas tiendas a partir de un conjunto individual de archivos XML.

Apéndice C. Grupos de elementos de base de datos

Todos los elementos de base de datos de WebSphere Commerce se dividen en grupos para crearse y cargarse. Estos grupos son un conjunto de tablas relacionadas de forma lógica. El orden en el que se organizan estos grupos de elementos de base de datos es importante para la carga de datos, puesto que los objetos deben existir antes de cargar la relación entre los objetos.

Al cargar elementos de base de datos en formato XML para la tienda, puede elegir cargar sólo grupos seleccionados. Estos grupos constan de los elementos de base de datos creados en los capítulos anteriores, por ejemplo los elementos de catálogo o de despacho de pedidos. Antes de cargar grupos de datos como se indica en el apartado “Carga de grupos de elementos de base de datos” en la página 413, realice lo siguiente:

- Determine qué grupo de elementos de base de datos está cargando. Cada grupo contiene dependencias que deben satisfacerse para que los elementos se puedan cargar. Revise la información del apartado “Dependencias de grupos de elementos de base de datos”.
- Asegúrese de haber creado o actualizado los archivos XML para el grupo de elementos de base de datos seleccionado. La información de cada uno de los capítulos de elementos de la tienda le recomienda el orden en el que debe crear los elementos de base de datos, pero cuando cree o actualice archivos XML, recuerde que la información de una tabla padre debe preceder a la información de una tabla hijo.

Dependencias de grupos de elementos de base de datos

Cada grupo de elementos de base de datos obtiene su información de las tablas de base de datos de WebSphere Commerce. Los elementos de base de datos tienen dependencias dentro de su propio grupo. Es decir, un grupo de elementos de base de datos no puede obtener datos en otros archivos XML de un grupo de datos distinto, y cada grupo es independiente menos las claves externas. Sin embargo, si el grupo de elementos de base de datos tiene que hacer referencia a los datos externos definidos en otro grupo, deberá proporcionar estos datos manualmente. Esto significa que los datos de un grupo tienen una *dependencia externa* de los datos definidos fuera de su dominio, es decir, en otro grupo de elementos de base de datos. Las dependencias externas se producen cuando un grupo de elementos de base de datos tiene una relación de clave externa con la clave primaria de una tabla en otro grupo. Para cargar un grupo de elementos de base de datos, deben satisfacerse sus dependencias externas. Para utilizar un ejemplo del diagrama siguiente, una de las dependencias externas del grupo de elementos de base de datos de la tienda es fulfillment.FFMCENTER.FFMCENTER_ID; esta dependencia indica que, para poder cargar el grupo de elementos de base de datos de la tienda, los elementos de base de datos de despacho de pedidos deben ya existir en la base de datos de WebSphere Commerce.

Consulte el diagrama siguiente antes de iniciar el proceso de carga. Cada grupo de elementos de base de datos es dependiente de otras tablas de base de datos, de las que se cargan los datos.

Algunos puntos que se deben recordar:

- Algunas dependencias externas quizá no se satisfagan con un solo grupo. Los elementos generales de base de datos o los elementos a nivel de sitio, utilizados por todas las tiendas, se rellenan automáticamente al crear la instancia en la *rutina de carga* y se pueden acceder de inmediato. Las tablas contenidas en grupos de elementos de base de datos tienen referencias de clave externa a este tipo de datos. Los datos de la rutina de carga están divididos en datos comunes y datos específicos del entorno nacional. Si tiene una tienda multilingüe, debe elegir los datos de rutina de carga comunes y los específicos del entorno nacional. Por ejemplo, necesita los datos de rutina de carga de idioma y de miembros. El proceso de creación de la instancia rellena la tabla LANGUAGE con los idiomas de WebSphere Commerce soportados para su tienda y crea una organización raíz (MEMBER.MEMBER_ID=-2001) y una organización por omisión (MEMBER.MEMBER_ID=-2000). Debe utilizar la organización raíz cuando sea necesario, pero debe crear una organización propietaria de la tienda en lugar de utilizar la organización por omisión. Para obtener más información sobre las organizaciones y su jerarquía, consulte la ayuda en línea de WebSphere Commerce.
- Los archivos que se listan bajo la columna **Dependencias externas** utilizan la siguiente estructura de denominación: *grupo elementos basedatos.tabla basedatos.columna basedatos*. Utilizando el archivo store.STOREENT.STOREENT_ID como ejemplo, los datos se obtienen del grupo de elementos de base de datos store, tabla STOREENT y columna STOREENT_ID. Los nombres de archivo que empiezan por *bootstrap* indican que los datos se han llenado durante la creación de la instancia de WebSphere Commerce.
- Los archivos listados bajo la columna **Dependencias externas** contienen referencias de claves externas a las **Tablas relacionadas**. Primero deben llenarse de datos estas tablas.
- Las tablas se han dividido, sólo a efectos de presentación, para indicar las tablas específicas de entorno nacional que contienen información multilingüe como, por ejemplo, las descripciones de productos.
- Las tablas del diagrama representan los elementos de base de datos de las tiendas de ejemplo de WebSphere Commerce. Las tablas pueden variar de acuerdo con el tamaño, la función y las necesidades de la tienda. Dependiendo de los requisitos de su tienda, asegúrese de incluir todas las tablas de base de datos que contienen los elementos de la tienda, incluso si esa tabla en concreto no está listada a continuación.

Elementos de base de datos de control de acceso		
Dependencias externas	Tablas relacionadas de los archivos XML de elementos de base de datos	Tablas relacionadas específicas de entorno nacional de los archivos XML de elementos de base de datos
bootstrap.LANGUAGE.LANGUAGE_ID (organizaciones raíz y propietaria de la tienda), bootstrap.MEMBER.MEMBER_ID (organizaciones raíz y propietaria de la tienda)	accesscontrol.xml ACACTACTGP, ACACTGRP, ACACTION, ACPOLICY, ACRESCGRY, ACRESGPRES, ACRESGRP	accesscontrol.xml ACACGPDESC, ACACTDESC, ACPOLDESC, ACRSCGDES, ACRESGPDES
Elementos de base de datos de política de negocio		

Dependencias externas	Tablas relacionadas de los archivos XML de elementos de base de datos	Tablas relacionadas específicas de entorno nacional de los archivos XML de elementos de base de datos
bootstrap.LANGUAGE.LANGUAGE_ID, bootstrap.MEMBER.MEMBER_ID, store.STOREENT.STOREENT_ID (organización propietaria de la tienda)	businesspolicy.xml POLICY, POLICYCMD	businesspolicy.xml POLICYDESC
Elementos de base de datos de campaña		
Dependencias externas	Tablas relacionadas de los archivos XML de elementos de base de datos	Tablas relacionadas específicas de entorno nacional de los archivos XML de elementos de base de datos
store.STOREENT.STOREENT_ID	campaign.xml CAMPAIGN, COLLATERAL, EMSPOT	campaign.xml COLLDESC
Elementos de base de datos de catálogo		
Dependencias externas	Tablas relacionadas de los archivos XML de elementos de base de datos	Tablas relacionadas específicas de entorno nacional de los archivos XML de elementos de base de datos

bootstrap.LANGUAGE.LANGUAGE_ID, bootstrap.MEMBER.MEMBER_ID (organización propietaria de la tienda), store.STOREENT.STOREENT_ID, shipping.CALCODE.CALCODE_ID, tax.CALCODE.CALCODE_ID	catalog.xml BASEITEM, CATALOG, CATENTREL, CATENTRY, CATGROUP, CATGRPREL, CATTOGRP, ITEMSPC, ITEMVERSN, RA, RADETAIL, STOREITEM, STORITMFFC, VERSIONSPC offering.xml CATGRPTPC, MGPTRDPSCN, OFFER, OFFERPRICE, TRADEPOSCN storefulfill.xml INVENTORY store-catalog.xml DISPCGPREL, DISPENTREL, STORECAT, STORECENT, STORECGRP store-catalog- shipping.xml CATENTCALCD, CATENTSHIP store-catalog- tax.xml CATENTCALD	catalog.xml ATTRIBUTE, ATTRVALUE, BASEITMDSC, CATALOGDSC, CATENTDESC, CATGRPDESC, PKGATTR, PKGATTRVAL,
Elementos de base de datos de mandato		
Dependencias externas	Tablas relacionadas de los archivos XML de elementos de base de datos	Tablas relacionadas específicas de entorno nacional de los archivos XML de elementos de base de datos
store.STOREENT.STOREENT_ID	command.xml CMDREG, VIEWREG	N/D
Business Elementos de base de datos de contratos		
Dependencias externas	Tablas relacionadas de los archivos XML de elementos de base de datos	
store.STOREENT.STOREENT_ID	Las tablas de base de datos de contrato no se cargan directamente y siguen un proceso diferente de los demás grupos de datos de WebSphere Commerce. Consulte el apartado "Publicación de elementos de contrato" en la página 420 para obtener más información.	

Elementos de base de datos de moneda		
Dependencias externas	Tablas relacionadas de los archivos XML de elementos de base de datos	Tablas relacionadas específicas de entorno nacional de los archivos XML de elementos de base de datos
store.STOREENT.STOREENT_ID	currency.xml CURCVLIST	currency.xml CURCONVERT, CURLIST
Elementos de base de datos de despacho de pedidos		
Dependencias externas	Tablas relacionadas de los archivos XML de elementos de base de datos	Tablas relacionadas específicas de entorno nacional de los archivos XML de elementos de base de datos
bootstrap.LANGUAGE.LANGUAGE_ID, bootstrap.MEMBER.MEMBER_ID (organización propietaria de la tienda)	fulfillment.xml FFMCENTER, STADDRESS	fulfillment.xml FFMCENTDS
Elementos de base de datos de organización		
Dependencias externas	Tablas relacionadas de los archivos XML de elementos de base de datos	Tablas relacionadas específicas de entorno nacional de los archivos XML de elementos de base de datos
bootstrap.LANGUAGE.LANGUAGE_ID (organizaciones raíz y propietaria de la tienda), bootstrap.MEMBER.MEMBER_ID (organizaciones raíz y propietaria de la tienda)	organization.xml ADDRBOOK, ADDRESS, MBRREL, MEMBER, ORGENCY	N/D
Elementos de base de datos de envío		
Dependencias externas	Tablas relacionadas de los archivos XML de elementos de base de datos	Tablas relacionadas específicas de entorno nacional de los archivos XML de elementos de base de datos
bootstrap.LANGUAGE.LANGUAGE_ID, bootstrap.MEMBER.MEMBER_ID (organización propietaria de la tienda), fulfillment.FFMCENTER.FFMCENTER_ID, store.STOREENT.STOREENT_ID	shipping.xml CALCODE, CALRULE, CRULESCALE, JURST, JURSTGPREL, JURSTGROUP, SHIPMODE, STENCALUSG shipping.xml SHPJCRULE, SHPARRANGE	shipping.xml CALCODEDSC, CALRANGE, CALLOOKUP, CALSCALE, SHPMODEDSC
Elementos de base de datos de tienda		
Dependencias externas	Tablas relacionadas de los archivos XML de elementos de base de datos	Tablas relacionadas específicas de entorno nacional de los archivos XML de elementos de base de datos

bootstrap.LANGUAGE.LANGUAGE_ID, bootstrap.MEMBER.MEMBER_ID(organización propietaria de la tienda), bootstrap.SETCURR.SETCURR_ID, fulfillment.FFMCENTER.FFMCENTER_ID	store.xml INVADJCODE, RTNREASON, STORE, STOREENT, STORELANG, VENDOR	store.xml FFMCENTDS, INVADJDESC, RTNRSNDESC, STADDRESS, STOREENTDS, STORLANGDS, VENDORDESC
Elementos de base de datos por omisión de tienda		
Dependencias externas	Tablas relacionadas de los archivos XML de elementos de base de datos	Tablas relacionadas específicas de entorno nacional de los archivos XML de elementos de base de datos
shipping.SHIPMODE.SHIPMODE_ID (si es aplicable, primero debe cargarse este archivo), contract.CONTRACT.CONTRACT_ID, store.STOREENT.STOREENT_ID	store-default.xml STOREDEF	N/D
Elementos de base de datos de impuestos		
Dependencias externas	Tablas relacionadas de los archivos XML de elementos de base de datos	Tablas relacionadas específicas de entorno nacional de los archivos XML de elementos de base de datos
bootstrap.LANGUAGE.LANGUAGE_ID, bootstrap.MEMBER.MEMBER_ID(organización propietaria de la tienda), store.STOREENT.STOREENT_ID, fulfillment.FFMCENTER.FFMCENTER_ID, store.STOREENT.STOREENT_ID	tax.xml CALCODE, CALRANGE, CALRLOOKUP, CALRULE, CALSCALE, CRULESCALE, JURST, JURSTGROUP, JURSTGPREL, STENCALUSG, TAXCGRY, TAXJCRULE taxfulfill.xml TAXJCRULE	tax.xml CALCODEDSC, CALSCALEDS, TAXCGRYDS

Apéndice D. Avisos

Esta información se ha desarrollado para productos y servicios ofrecidos en EE.UU.

Es posible que IBM no proporcione los productos, servicios o características a los que hace referencia este documento en otros países. Póngase en contacto con su representante de IBM local para obtener información acerca de los productos y servicios disponibles actualmente en su área. Cualquier referencia a un producto, programa o servicio de IBM no pretende afirmar ni implica que sólo pueda utilizarse ese producto, programa o servicio de IBM. En su lugar puede utilizarse cualquier producto, programa o servicio funcionalmente equivalente que no vulnere ninguno de los derechos de propiedad intelectual de IBM. No obstante, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio que no sea de IBM.

IBM puede tener patentes o solicitudes de patente pendientes que cubran temas tratados en este documento. La posesión de este documento no le otorga ninguna licencia sobre dichas patentes. Puede enviar consultas sobre licencias, por escrito, a:

*IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
EE.UU.*

Para realizar consultas sobre la licencia en relación a información de doble byte (DBCS), póngase en contacto con el departamento de propiedad intelectual de IBM en su país o envíe sus consultas, por escrito, a:

*IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokio 106, Japón*

El párrafo siguiente no es aplicable al Reino Unido ni a cualquier otro país en el que tales disposiciones contradigan la normativa local:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL", SIN GARANTÍAS DE NINGUNA CLASE, NI EXPLÍCITAS NI IMPLÍCITAS, INCLUIDAS, PERO SIN LIMITARSE A, LAS GARANTÍAS IMPLÍCITAS DE NO INFRACCIÓN, COMERCIALIZACIÓN O IDONEIDAD PARA UNA FINALIDAD DETERMINADA. Algunas legislaciones no contemplan la exclusión de garantías, ni implícitas ni explícitas, en determinadas transacciones, por lo que puede haber usuarios a los que no les afecte dicha norma.

Esta información puede contener imprecisiones técnicas o errores tipográficos. La información aquí contenida está sometida a cambios periódicos; tales cambios se irán incorporando en nuevas ediciones de la publicación. IBM se reserva el derecho de realizar cambios y/o mejoras, cuando lo considere oportuno y sin previo aviso, en los productos y/o programas descritos en esta publicación.

Todas las referencias hechas en este documento a sitios Web que no son de IBM se proporcionan únicamente para su información y no representan en modo alguno una recomendación de dichos sitios Web. El contenido de esos sitios Web no forma parte del contenido de este producto de IBM, por lo que la utilización de dichos sitios es responsabilidad del usuario.

IBM puede utilizar o distribuir la información que se le envíe del modo que estime conveniente sin incurrir por ello en ninguna obligación para con el remitente.

Los propietarios de licencias de este programa que deseen obtener información sobre el mismo con el fin de permitir: (i) el intercambio de información entre programas creados independientemente y otros programas (incluido éste) y (ii) el uso mutuo de la información que se ha intercambiado, deberán ponerse en contacto con:

*IBM Canada Ltd.
Office of the Lab Director
Av. Diagonal 571, Edif. L'illa
Markham, Ontario L6G 1C7
Canadá*

Dicha información puede estar disponible sujeta a los términos y condiciones apropiados, incluyendo, en algunos casos, el pago de una cantidad.

IBM proporciona el programa bajo licencia descrito en esta información, y todo el material bajo licencia disponible para el mismo, bajo los términos del Contrato de cliente IBM, el Acuerdo Internacional de Programas bajo Licencia IBM, o de cualquier acuerdo equivalente entre IBM y el cliente.

La información sobre productos que no son de IBM se ha obtenido de los distribuidores de dichos productos, de los anuncios publicados o de otras fuentes disponibles públicamente. IBM no ha probado esos productos y no puede confirmar la precisión del rendimiento, la compatibilidad ni ninguna otra afirmación relacionada con productos que no son de IBM. Las preguntas sobre las prestaciones de productos no de IBM deben dirigirse a los distribuidores de dichos productos.

Esta información contiene ejemplos de datos e informes que se utilizan en operaciones comerciales cotidianas. Para ilustrar los ejemplos de la forma más completa posible, éstos incluyen nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier similitud con nombres y direcciones utilizados por una empresa real es puramente casual.

LICENCIA DE COPYRIGHT:

Esta información contiene programas de aplicación de ejemplo en lenguaje fuente que ilustran las técnicas de programación en diversas plataformas operativas. Puede copiar, modificar y distribuir libremente estos programas de ejemplo, sin pagar por ello a IBM, con la finalidad de desarrollar, utilizar, comercializar o distribuir programas de aplicación que se ajusten a la interfaz de programas de aplicación para la plataforma operativa para la cual están escritos los programas de ejemplo. Estos ejemplos no han sido probados en profundidad bajo todas las condiciones. En consecuencia, IBM no puede garantizar ni afirmar la fiabilidad, operatividad o funcionalidad de estos programas. Puede copiar, modificar y distribuir libremente estos programas de ejemplo, sin pagar por ello a IBM, con la

finalidad de desarrollar, utilizar, comercializar o distribuir programas de aplicación que se ajusten a las interfaces de programas de aplicación de IBM.

Cada copia, parcial o completa, de estos programas de ejemplo, o cualquier trabajo obtenido a partir de los mismos, debe incluir el siguiente aviso de copyright:

©Copyright International Business Machines Corporation 2003. Parte de este código se ha obtenido de los programas de ejemplo de IBM Corp. ©Copyright IBM Corp. 2000, 2003. Reservados todos los derechos.

Si examina esta información en formato de copia software, es posible que algunas fotografías o ilustraciones a color no aparezcan.

Las imágenes de tarjetas de crédito, las marcas registradas y los nombres de comercios de este producto sólo deben ser utilizados por comerciantes autorizados por el propietario de la marca de tarjeta de crédito para aceptar el pago a través de dicha tarjeta de crédito.

Marcas registradas

Los términos siguientes son marcas registradas de International Business Machines Corporation en los Estados Unidos y/o en otros países:

400	IBM	
AIX	Lotus	WebSphere
DB2	OS/400	VisualAge
DB2 Universal Database	QuickPlace	
Domino	Sametime	

El logotipo de IBM es una marca registrada de International Business Machines Corporation en Estados Unidos y en otros países.

Java y todas las marcas registradas basadas en Java son marcas registradas de Sun Microsystems, Inc. en Estados Unidos y/o en otros países.

Microsoft, Windows y Windows NT son marcas registradas de Microsoft Corporation en Estados Unidos y/o en otros países.

Vital Processing Services es una marca registrada de Vital Corporation.

VisaNet es una marca registrada de Visa International Services Organization.

Otros nombres de compañías, productos y servicios pueden ser marcas registradas o marcas de servicio de otras empresas.

IBM