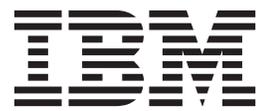


IBM Business Process Manager  
Versión 7.5.0

*Visión general de IBM Business  
Process Manager*





---

## Manuales en PDF y Centro de información

Estos manuales en PDF se proporcionan a efectos prácticos para su impresión o su lectura cuando esté fuera de línea. Para ver la información más reciente, vea el Centro de información en línea.

En conjunto, los manuales en PDF contienen el mismo contenido que el Centro de información. Algunos enlaces a manuales PDF pueden estar adaptados para su utilización en Information Centers y es posible que no funcionen adecuadamente.

La documentación en PDF está disponible en un plazo de un trimestre después de un release importante del Centro de información como, por ejemplo, las versiones 7.0 ó 7.5.

La documentación en PDF se actualiza con menos frecuencia que el Centro de información, pero con más frecuencia que los Redbooks. En general, los manuales en PDF se actualizan cuando se han acumulado cambios suficientes para el manual.



# Contenido

Manuales en PDF y Centro de información . . . . .	iii
---	-----

## Capítulo 1. Iniciación a IBM Business Process Manager V7.5 . . . . . 1

Visión general del producto . . . . .	1
Configuraciones de IBM Business Process Manager V7.5 . . . . .	3
Prestaciones de configuración de IBM Business Process Manager V7.5 . . . . .	3
El repositorio de Process Center . . . . .	4
Process Server y entornos de tiempo de ejecución . . . . .	6
Entornos de creación . . . . .	6
Novedades de IBM Business Process Manager V7.5 . . . . .	8
Accesibilidad en IBM Business Process Manager . . . . .	11
Disponibilidad de idiomas nacionales en IBM Business Process Manager . . . . .	11
Visión general de la gestión de procesos de negocio . . . . .	12
Visión general del modelado de procesos . . . . .	13
Desarrollo de procesos con Process Center . . . . .	13
Aplicaciones de proceso: visión general . . . . .	14
Ejecución y depuración de procesos con Inspector . . . . .	15
Despliegue y gestión de aplicaciones de proceso . . . . .	16
Creación, acceso e incorporación de servicios . . . . .	17
Acceso a servicios externos a una aplicación . . . . .	17
Crear o llamar a un servicio Web . . . . .	22

## Capítulo 2. Aprender más acerca de IBM Business Process Manager . . . . . 25

Mantenimiento de versiones . . . . .	25
Creación de versiones de aplicaciones de proceso . . . . .	25
Mantenimiento de versiones de módulos y bibliotecas . . . . .	26
Convenio de denominación . . . . .	27
Convenios de denominación para despliegues de servidor de Process Center . . . . .	28
Convenios de denominación para despliegues de Process Server . . . . .	30
Enlaces que tienen en cuenta la versión . . . . .	31
Invocación dinámica que tiene en cuenta la versión . . . . .	33
Despliegue de process applications con módulos Java y proyectos . . . . .	34
Despliegue de process applications con reglas de negocio y selectores . . . . .	34
Enlaces . . . . .	34
Visión general de los enlaces de exportación e importación . . . . .	36
Configuración de los enlaces de exportación e importación . . . . .	40
Transformación del formato de datos en importaciones y exportaciones . . . . .	40
Manejadores de datos . . . . .	41

Enlaces de datos . . . . .	43
Selectores de función en enlaces de exportación . . . . .	44
Manejo de errores . . . . .	46
Cómo se manejan los errores en los enlaces de exportación . . . . .	47
Cómo se manejan los errores en los enlaces de importación . . . . .	49
Interoperatividad entre módulos SCA y servicios Open SCA . . . . .	51
Tipos de enlace . . . . .	54
Selección de los enlaces adecuados . . . . .	54
Enlaces SCA . . . . .	55
Enlaces de servicio web . . . . .	55
Visión general de enlaces de servicios web . . . . .	55
Propagación de cabecera SOAP . . . . .	57
Propagación de cabecera de transporte . . . . .	60
Trabajo con enlaces de servicio web (JAX-WS) . . . . .	62
Adjuntos en mensajes SOAP . . . . .	65
Uso del enlace de estilo de documento WSDL con mensajes de varias partes . . . . .	78
Enlaces HTTP . . . . .	80
Visión general de enlaces HTTP . . . . .	80
Cabeceras HTTP . . . . .	81
Enlaces de datos HTTP . . . . .	85
Enlaces EJB . . . . .	88
Enlaces de importación EJB . . . . .	88
Enlaces de exportación EJB . . . . .	89
Propiedades de enlace EJB . . . . .	90
Enlaces EIS . . . . .	94
Visión general de los enlaces EIS . . . . .	95
Características clave de los enlaces EIS . . . . .	95
Propiedades dinámicas de la especificación de conexión y la especificación de interacción JCA . . . . .	98
Clientes externos con enlaces EIS . . . . .	99
Enlaces JMS . . . . .	100
Visión general de enlaces JMS . . . . .	100
Integración JMS y adaptadores de recursos . . . . .	103
Características clave de los enlaces JMS . . . . .	103
Cabeceras JMS . . . . .	104
Esquema de correlación de destinos de respuesta dinámicos temporales de JMS . . . . .	105
Clientes externos . . . . .	106
Resolución de problemas de enlaces JMS . . . . .	107
Manejo de excepciones . . . . .	108
Enlaces JMS genéricos . . . . .	109
Visión general de enlaces de JMS genérico . . . . .	109
Características clave de los enlaces JMS genéricos . . . . .	112
Cabeceras JMS genéricas . . . . .	113
Resolución de problemas de enlaces JMS genéricos . . . . .	114
Manejo de excepciones . . . . .	115
Enlaces JMS de WebSphere MQ . . . . .	116

Visión general de enlaces JMS de WebSphere MQ. . . . .	116	Manejo de excepciones . . . . .	132
Características clave de los enlaces JMS de WebSphere MQ. . . . .	119	Limitaciones de los enlaces. . . . .	133
Cabeceras JMS . . . . .	120	Limitaciones del enlace MQ . . . . .	133
Clientes externos . . . . .	121	Limitaciones de los enlaces JMS, MQ JMS y JMS genéricos . . . . .	133
Resolución de problemas de enlaces JMS de WebSphere MQ . . . . .	122	Objetos de negocio . . . . .	134
Manejo de excepciones . . . . .	123	Definición de objetos de negocio . . . . .	135
Enlaces de WebSphere MQ. . . . .	123	Utilización de objetos de negocio. . . . .	135
Visión general de enlaces WebSphere MQ	124	Objetos empresariales especiales . . . . .	137
Características clave de un enlace WebSphere MQ. . . . .	126	Modalidad de análisis de objetos de negocio	138
Cabeceras de WebSphere MQ . . . . .	128	Consideraciones cuando se selecciona la modalidad de análisis de objetos de negocio .	139
Adición estática de MQCIH a un enlace WebSphere MQ. . . . .	130	Ventajas del uso de la modalidad de análisis lento (lazy) frente al diligente (eager) . . . .	139
Clientes externos . . . . .	130	Consideraciones sobre la migración de aplicaciones y el desarrollo . . . . .	140
Resolución de problemas de enlaces WebSphere MQ. . . . .	131		

---

## Capítulo 1. Iniciación a IBM Business Process Manager V7.5

Entender qué prestaciones proporciona IBM® Business Process Manager para la gestión de procesos de negocio y cómo se relacionan entre sí las distintas fases de la gestión de procesos de negocio, como crear y desplegar aplicaciones de proceso.

La aplicación de proceso es el contenedor básico para procesos y sus componentes en IBM Business Process Manager. Los diseñadores de procesos crean process applications en los entornos de creación y pueden incluir servicios, tareas y artefactos necesarios para dar soporte a la ejecución.

Los Servicios de integración avanzados se implementan en Integration Designer y se asocian a las process applications. Desde Process Center, las aplicaciones de proceso se despliegan en Process Server, que es el entorno de ejecución de procesos de IBM Business Process Manager.

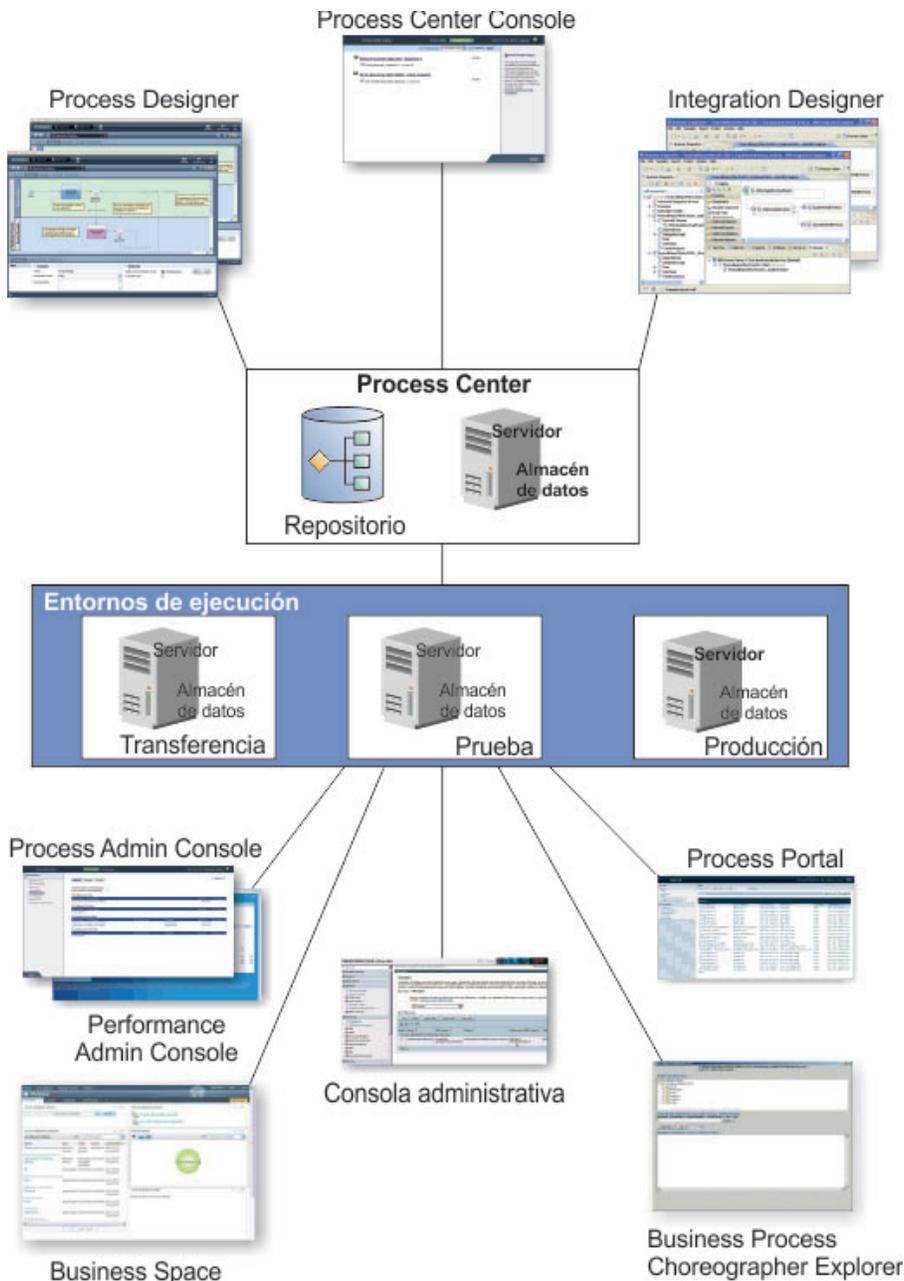
Del mismo modo, los procesos automatizados creados en Integration Designer pueden utilizar flujos de actividad de usuario que se hayan desarrollado en IBM Process Designer.

---

### Visión general del producto

Los componentes de IBM Business Process Manager proporcionan un repositorio de BPM unificado, herramientas para autores, administradores y usuarios y una plataforma de ejecución. Diferentes configuraciones del producto soportan diferentes niveles de complejidad y participación con la gestión de procesos de negocio.

El diagrama siguiente muestra una configuración típica de IBM Business Process Manager:



- Desde los entornos de creación IBM Process Designer y IBM Integration Designer, varios usuarios se conectan a Process Center.
- En los entornos de creación Process Designer y Integration Designer, los diseñadores de procesos y servicios crean process applications desplegables y toolkits reutilizables. Las process applications contienen modelos de proceso e implementaciones de servicio, incluyendo los archivos de soporte necesarios. Se almacenan en el repositorio de Process Center, donde se pueden compartir.
- Process Center incluye un servidor Process Center server y Business Performance Data Warehouse, lo que permite a los usuarios trabajar en IBM Process Designer para ejecutar sus process applications y almacenar datos de rendimiento a efectos de prueba y reproducción durante los trabajos de desarrollo.
- Desde la consola de Process Center, los administradores instalan aplicaciones de proceso que están preparadas para la transferencia, prueba o producción en los servidores Process Server de esos entornos.
- Desde la consola de Process Center, los administradores gestionan instancias activas de aplicaciones de proceso en todos los entornos configurados.

- Desde IBM Process Portal, los usuarios finales realizan las tareas asignadas. El servidor Process Center y Process Server en entornos de ejecución configurados pueden ejecutar las process applications que crean las tareas asignadas.
- Mediante Process Portal, los participantes del proceso pueden conectarse a un servidor Process Center o a un Process Server en cualquier entorno de ejecución configurado, dependiendo de si se está desarrollando, probando o publicando un proceso en un entorno de producción.
- El almacén de datos de rendimiento recupera datos rastreados a partir del servidor Process Server o del servidor de Process Center a intervalos regulares. Los usuarios pueden crear y ver informes que utilizan estos datos en los entornos de creación y IBM Process Portal.
- Desde la consola administrativa de procesos y la consola administrativa de rendimiento, los administradores pueden gestionar y mantener todos los servidores de ejecución.

## Configuraciones de IBM Business Process Manager V7.5

Las diferentes configuraciones de IBM Business Process Manager se correlacionan con puntos de entrada o etapas típicas del programa de gestión de procesos de negocio de una empresa.

Tabla 1. Configuraciones de IBM Business Process Manager

Configuración	Fase
Advanced	<p><b>Transformación</b></p> <p>Conjunto completo de las prestaciones de gestión de procesos de negocio</p> <ul style="list-style-type: none"> <li>• Soporte ampliado para la automatización de un alto volumen de procesos</li> <li>• Componentes SOA incorporados para servicio, integración y orquestación en toda la empresa</li> </ul>
Standard	<p><b>Programa</b></p> <p>Configurado para proyectos de gestión de procesos de negocio típicos</p> <ul style="list-style-type: none"> <li>• Para programas de mejora de varios proyectos con mucha implicación de negocio</li> <li>• Soporte de integración del sistema básico</li> <li>• Tiempo de materialización rápido y productividad de usuario mejorada</li> </ul>
Express	<p><b>Proyecto</b></p> <p>Configurado para el primer proyectos de gestión de procesos de negocio</p> <ul style="list-style-type: none"> <li>• Tiempo de materialización rápido: productividad de usuario mejorada</li> <li>• Bajo precio de entrada</li> <li>• Configuración e instalación sencillas</li> </ul>

## Prestaciones de configuración de IBM Business Process Manager V7.5

Conozca qué productos y prestaciones ofrece IBM para la gestión de procesos de negocio, y elija el adecuado para su empresa.

IBM Business Process Manager es una plataforma de gestión de procesos de negocio que combina prestaciones centradas en el hombre y en la integración para formar un producto unificado. Existen diversas configuraciones del producto destinadas a usuarios diferentes y que satisfacen necesidades empresariales diferentes. Las configuraciones del producto se pueden combinar en entornos de creación conjunta y entornos de ejecución desplegados en red.

Tabla 2. Prestaciones de configuración de IBM Business Process Manager

Prestación	Advanced	Standard	Express
Ejecución compatible de WebSphere Lombardi Edition	X	X	X
Process Designer (BPMN)	X	X	X

Tabla 2. Prestaciones de configuración de IBM Business Process Manager (continuación)

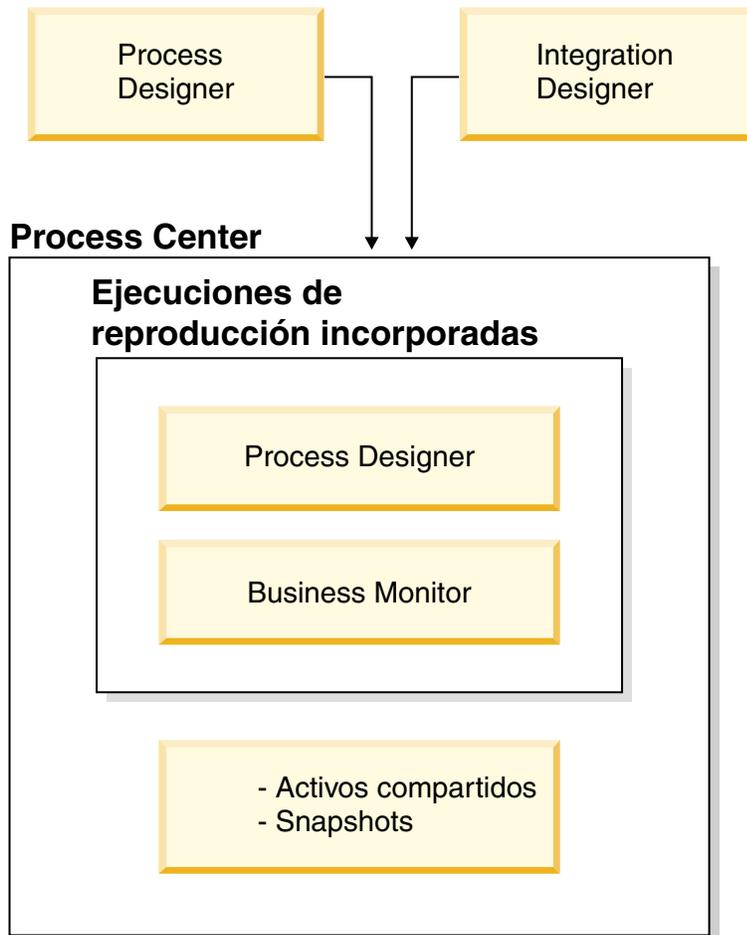
Prestación	Advanced	Standard	Express
Edición colaborativa / reproducción inmediata	X	X	X
Interfaces de usuario interactivas "coach de proceso"	X	X	X
Reglas de proceso basadas en ILOG	X	X	X
Supervisión y creación de informes en tiempo real	X	X	X
Analíticas de rendimiento y optimizador	X	X	X
Performance Data Warehouse	X	X	X
Process Center / repositorio de activos compartidos	X	X	X
Creadores y usuarios finales de procesos ilimitados	X	X	200 usuarios / 3 creadores
Alta disponibilidad: agrupación en clúster y núcleos ilimitados	X	X	<ul style="list-style-type: none"> <li>• Producto de 4 núcleos</li> <li>• Desarrollo de 2 núcleos</li> <li>• Sin clúster</li> </ul>
Ejecución compatible de WebSphere Process Server	X		
Integration Designer (BPEL / SOA)	X		
Enterprise Service Bus (ESB) incorporado	X		
Soporte de transacciones	X		
Adaptadores de integración	X		
Interfaz de usuario de Business Space flexible	X		

## El repositorio de Process Center

Process Center incluye un repositorio para todos los procesos, servicios y otros activos creados en los entornos de creación de IBM Business Process Manager, Process Designer y Integration Designer.

Process Center es un motor de ejecución en el que Process Designer y Integration Designer comparten activos, lo que permite que desarrollen conjuntamente procesos de negocio de una forma altamente interactiva. Estos procesos de negocio puede utilizar puntos de supervisión creados con Business Monitor Development Toolkit. El resultado es un proceso de negocio que se puede examinar durante la ejecución para comprobar su efectividad bajo condiciones de trabajo reales. Business Monitor proporciona una vista de panel de control con indicadores y tablas de puntuación. Puede añadir alertas y notificaciones que le permiten conocer en cada momento la marcha del proceso de negocio. Se pueden localizar y corregir las restricciones del rendimiento, deficiencias y errores de asignación de recursos en un proceso de negocio activo, con lo que mejora el rendimiento de los procesos de negocio.

En el diagrama que sigue, verá diversos componentes relacionados que utilizados conjuntamente le permiten crear procesos de negocio complejos.



La consola de Process Center proporciona las herramientas que necesita para mantener el repositorio.

- Desde la consola de Process Center, puede crear aplicaciones de proceso y toolkits, y otorgar a los usuarios acceso a esas aplicaciones de proceso y toolkits.
- En los entornos de creación, puede crear modelos de proceso, servicios y otros activos dentro de aplicaciones de proceso.
- Process Center incluye un servidor de Process Center y un almacén de datos de rendimiento, que permite a los usuarios que trabajan en los entornos de creación ejecutar procesos y almacenar datos de rendimiento con fines de prueba y reproducción.
- Desde la consola de Process Center, los administradores instalan aplicaciones de proceso que están preparadas para las fases de prueba o producción en los servidores de proceso de esos entornos.
- Desde la consola de Process Center, los administradores gestionan instancias activas de aplicaciones de proceso en entornos configurados.

La consola de Process Center proporciona una cómoda ubicación en la que crear y mantener contenedores de alto nivel, por ejemplo process applications y toolkits. Los administradores que no trabajan de forma activa en la vista Designer pueden utilizar la consola Process Center para proporcionar una infraestructura en la que los analistas y desarrolladores de BPM puedan crear sus procesos e implementaciones subyacentes. Otra tarea principal de los administradores es gestionar el acceso al repositorio de Process Center configurando la autorización correspondiente de los usuarios y grupos.

Estos usuarios con autorización adecuada pueden realizar algunas tareas administrativas directamente en Process Designer y Integration Designer. Por ejemplo, un desarrollador con acceso de escritura a la

aplicación de proceso que desee capturar el estado de todos los activos del proyecto para un determinado momento en el tiempo, puede crear una snapshot mientras trabaja en la vista Designer.

## **Process Server y entornos de tiempo de ejecución**

Process Server proporciona un entorno de ejecución de BPM único que puede soportar un rango de procesos de negocio, organización de servicios y posibilidades de integración.

En los entornos de creación, el servidor de procesos integrado en Process Center le permite ejecutar procesos a medida que los crea. Cuando esté preparado, puede instalar y ejecutar esos mismos procesos en los servidores de procesos de los entornos de ejecución. El componente Business Performance Data Warehouse recopila y agrega datos de proceso de los procesos que se ejecutan en los servidores de procesos. Puede utilizar estos datos para mejorar los procesos de negocio.

Process Admin Console le permite gestionar los process servers de los entornos de ejecución, por ejemplo, transferencia, prueba y producción, así como el servidor de procesos que forma parte de Process Center.

## **Entornos de creación**

IBM Business Process Manager Advanced ofrece dos entornos de creación. Utilice IBM Process Designer para modelar de forma eficiente procesos de negocio en los que intervienen tareas humanas. Utilice IBM Integration Designer para construir servicios autocontenidos o que invocan otros servicios existentes, como por ejemplo servicios web, aplicaciones de recursos de empresa o aplicaciones que se ejecutan en CICS e IMS.

Process Designer está disponible en todas las ediciones del producto. IBM Business Process Manager Advanced también ofrece Integration Designer con sus editores y adaptadores asociados.

### **Process Designer**

Un proceso es la unidad lógica principal en IBM Business Process Manager. Es el contenedor de todos los componentes de una definición de proceso, incluidos servicios, actividades y puntos de acceso; sucesos de temporizador, mensaje y excepción; líneas de secuencia, reglas y variables. Al modelar un proceso, está creando una definición de proceso de negocio reutilizable (BPD). Utilice IBM Process Designer para crear modelos de proceso que pueden contener tareas de usuario.

Process Designer le ayuda a desarrollar procesos de negocio. Mediante una herramienta gráfica de fácil utilización, puede crear una secuencia de acciones que forman un proceso de negocio, y puede reconstruir ese proceso a medida que las circunstancias cambien con el paso del tiempo. Si una o más actividades necesitan acceder a grandes sistemas o servicios que proporcionan datos para el proceso de negocio, por ejemplo, para obtener información sobre clientes, puede satisfacer esa necesidad utilizando Integration Designer. Mediante una interfaz simple, una actividad de Process Designer puede invocar un servicio creado en Integration Designer. Ese servicio puede utilizar flujos de mediación para transformar, encaminar y mejorar datos y adaptadores para acceder a sistemas de fondo de forma regular. Es decir, Process Designer está centrado en el proceso de negocio e Integration Designer está centrado en servicios automatizados para complementar el proceso de negocio.

Todos los proyectos de Process Designer están contenidos en process applications. Esas process applications y esos artefactos asociados se guardan en el repositorio de Process Center. Process applications que comparten activos colocados en toolkits.

IBM Business Process Manager proporciona varias interfaces de usuario para permitirle modelar, implementar, simular e inspeccionar procesos de negocio. Puede crear y gestionar process applications, toolkits, pistas y snapshots en Process Center Console. Puede crear modelos de proceso, informes y servicios simples en Process Designer. Puede ejecutar y depurar los procesos en el Inspector. Y puede ejecutar simulaciones en el Optimizer.

La process applications desarrolladas en Process Designer se pueden ejecutar en cualquier momento en el servidor Process Center o guardar en una snapshot y desplegar en el Process Server. Lo mismo vale para los servicios desplegados en Integration Designer y asociados a process applications.

## **Integration Designer**

Integration Designer proporciona editores y ayudas para ayudar a los desarrolladores a crear servicios y procesos automatizados complejos. Está disponible como un componente en IBM Business Process Manager Advanced o como un conjunto de herramientas autónomo para otros usos.

IBM Integration Designer se ha diseñado como entorno de desarrollo de integración completa para aquellos que crean aplicaciones integradas. Las aplicaciones integradas no son sencillas. Pueden llamar a aplicaciones de sistemas de información empresarial (EIS), implicar a procesos de negocio de diversos departamentos o empresas e invocar local o remotamente aplicaciones escritas en diversos lenguajes y ejecutadas en diversos sistemas operativos. Los componentes se crean y ensamblan en otras aplicaciones integradas (es decir, en aplicaciones creadas a partir de un conjunto de componentes) por medio de editores visuales. Los editores visuales presentan una capa de abstracción entre los componentes y sus implementaciones. Un desarrollador que utilice las herramientas puede ensamblar una aplicación integrada sin tener un conocimiento detallado de la implementación subyacente de cada componente.

Las herramientas de Integration Designer se basan en una arquitectura orientada a servicios. Los componentes son servicios, y una aplicación integrada que implica a muchos componentes también es un servicio. Los servicios creados se ajustan a los estándares de vanguardia ampliamente utilizados en la industria. Los procesos BPEL , que también se convierten en componentes, se crean de forma parecida con herramientas visuales de fácil utilización, que se ajustan al estándar industrial BPEL (Business Process Execution Language).

En el paradigma de Integration Designer, los componentes se ensamblan en módulos. Las importaciones y las exportaciones se utilizan para compartir datos entre módulos. Los artefactos colocados en una biblioteca se pueden compartir entre módulos.

Los módulos y las bibliotecas se pueden asociar con una aplicación de proceso para utilizarlos con Process Center y los procesos creados en Process Designer los pueden utilizar como servicios. En tales casos, también pueden desplegarse con la process application.

Como alternativa, los módulos y las bibliotecas pueden desplegarse directamente en el entorno de prueba o al Process Server. Puede utilizar módulos de mediación para crear flujos de mediación que puede desplegar en WebSphere Enterprise Service Bus o en el Process Server.

IBM Integration Designer proporciona también la posibilidad de crear tipos de datos y correlaciones xml que se pueden desplegar en el dispositivo WebSphere DataPower. Puede también intercambiar archivos con WebSphere DataPower.

## **Novedades de los productos de Business Process Management Versión 7.5**

Conozca las novedades de IBM Business Process Management en las páginas de novedades individuales para cada producto o componente.

Las características nuevas y las mejoras están centradas en estas necesidades de gestión de procesos de negocio de alto nivel:

- Mejorar la visibilidad de los procesos de negocio
- Maximizar la reutilización
- Gobernar los cambios
- Poder ampliar el ámbito y la escala de los proyectos

Las páginas siguientes describen las características nuevas de IBM Business Process Manager V7.5 y de otros productos y componentes de gestión de procesos de negocio.

- Novedades de IBM Business Process Manager V7.5
- Novedades de IBM Integration Designer Versión 7.5
- Novedades de Business Space powered by WebSphere Versión 7.5
- Novedades de IBM Business Monitor Versión 7.5

## **Novedades de IBM Business Process Manager V7.5**

IBM Business Process Manager incorpora funcionalidad nueva de WebSphere Process Server, WebSphere Lombardi Edition y IBM Integration Designer (anteriormente WebSphere Integration Developer) en un entorno de usuario unificado. Integra la funcionalidad más actual de estos tres productos y mejora las características de las versiones del producto anteriores.

IBM Business Process Manager es una plataforma de gestión de procesos de negocio exhaustiva y consumible que proporciona visibilidad y gestión de los procesos de negocio. Incluye herramientas y tiempo de ejecución para el diseño, la ejecución, la supervisión y la optimización de procesos. También está diseñada para ayudar a los propietarios de proceso y a los usuarios de negocio a involucrarse en la mejora de sus procesos de negocio.

Algunas de las características nuevas y mejoradas de la lista siguiente ya existían en versiones anteriores de uno de los productos (WebSphere Process Server, WebSphere Lombardi Edition o WebSphere Integration Developer). Sin embargo, las características son nuevas para los usuarios que anteriormente sólo tenían uno o dos de los productos pero no los tres.

Algunas características que existían anteriormente en versiones anteriores de los productos integrados se han sustituido por características nuevas en IBM Business Process Manager. Para obtener una lista de las características en desuso, consulte Características en desuso.

## **Características nuevas y mejoradas en IBM Business Process Manager V7.5**

IBM Business Process Manager V7.5 proporciona las características nuevas o mejoradas siguientes:

- Un entorno de creación unificado para crear procesos BPMN (Business Process Modeling Notation) centrados en los usuarios y basados en los estándares (mediante Process Designer) e integrándolos con procesos BPEL (Business Process Execution Language) centrados en el sistema (mediante Integration Designer).
- Un repositorio compartido y un entorno de ejecución común para los procesos BPMN y los procesos BPEL. Conjuntamente, el entorno de creación unificado, el repositorio compartido y el entorno de ejecución común proporcionan las mejoras siguientes:
  - Implementación y prueba gráfica de process applications, servicios, interfaces de usuario y reglas.
  - Diseño de procesos basado en estándares mediante BPMN.
  - Herramientas de IBM Integration Designer para construir servicios, transformaciones de datos, orquestaciones de BPEL e integración para aplicaciones y sistemas de fondo.
  - Creación de reglas de negocio que expresan la lógica de negocio de forma accesible, proporcionando la misma experiencia de creación de reglas que WebSphere ILOG JRules. El contenido de las reglas se puede exportar fácilmente a WebSphere ILOG JRules, proporcionando un punto de inicio para decisiones de negocio.
  - El motor de creación de gráficos IBM ILOG JViews Chart que se ha añadido a IBM Business Process Manager V7.5. Para los gráficos personalizados existentes, se proporcionan herramientas para pasar del formato CDL al formato de hoja de estilo en cascada (CSS).
  - Hojas de propiedades que hacen posible configurar detalles de implementación en lugar de codificarlos de modo que los usuarios con perfil menos técnico pueden participar en el diseño.

- Integración de ciclo de vida completa, manteniendo la solución sincronizada desde el diseño del modelo hasta el despliegue.
- Reproducción interactiva que posibilita la validación de las necesidades del proceso en cualquier momento.
- Biblioteca compartida de todos los activos del proceso para una implementación colaborativa y la posibilidad de reutilización mediante arrastrar y soltar.
- Control en tiempo real para procesos en curso mediante estas características:
  - El buzón de IBM Process Portal proporciona una vista consolidada de todas las tareas pendientes.
  - Una vista gráfica del estado de un proceso ayuda a los participantes del proceso a comprender los pasos restantes del proceso.
  - El modelado de sucesos explícito define el flujo de trabajo y el manejo de excepciones.
  - Los umbrales y los parámetros de proceso expuestos proporcionan un control en tiempo real durante el proceso en curso.
  - Las acciones ad hoc están soportadas; los usuarios designados pueden realizar tareas espontáneas y alterar el flujo de trabajo.
  - Los usuarios pueden comunicar y colaborar en tareas específicas durante la ejecución del proceso.
  - Los paneles de resultados en tiempo real proporcionan visibilidad del trabajo en progreso, conjuntamente con la capacidad de realizar una acción correctiva si es necesario.
  - La infraestructura para la interfaz de usuario de Business Space está incluida en Business Process Manager.
  - Una vista federada hace que sea posible realizar tareas, gestionar elementos de trabajo, hacer un seguimiento del rendimiento y responder a sucesos, todo en tiempo real.
  - Los usuarios también pueden realizar directamente tareas de proceso desde Microsoft Office y Microsoft SharePoint. Las integraciones de Microsoft se ofrecen como componentes de complemento opcionales.
- Prestaciones de supervisión mejoradas con visibilidad de procesos en interacciones del sistema y de los usuarios. IBM Business Process Manager soporta la visibilidad intuitiva de procesos dinámicos con estas características:
  - Las medidas de rendimiento se expresan en términos de los indicadores clave de rendimiento (ICRs) y de los acuerdos de nivel de servicio (SLA).
  - Business Performance Data Warehouse recopila y correlaciona automáticamente sucesos y datos de negocio con medidas de proceso de modelado para la visibilidad del proceso que siempre está actualizada.
  - El optimizador de procesos permite visualizar cuellos de botella de rendimiento y otros problemas directamente en el diagrama de modelo de proceso.
- Las características de WebSphere Process Server, incorporadas ahora en IBM Process Server, proporcionan la integridad de transacciones y la escalabilidad mejorada desde los primeros proyectos hasta las soluciones para toda la empresa:
  - WebSphere Application Server incorporado proporciona alta escalabilidad y disponibilidad.
  - Un sólo servidor de tiempo de ejecución BPM (Process Server) soporta todo el rango de procesos de negocio, orquestación de servicios e integración.
  - Los componentes de la arquitectura orientada a servicios (SOA) incorporada incluyen una infraestructura de conectividad de Enterprise Service Bus (ESB) flexible para integrar aplicaciones y servicios.
  - Los estándares soportados incluyen la ejecución BPMN y BPEL y muchos datos y estándares de servicios.
  - Los adaptadores de integración conectan los activos de aplicación e información con el ESB para acelerar los proyectos de integración de negocio con conexiones desplegables rápidamente, listas para la empresa y basadas en los procedimientos recomendados. Se incluye un conjunto de

- prestaciones exhaustivo para habilitar sus activos para servicio, incluyendo aplicaciones empaquetadas, personalizadas y heredadas, protocolos de tecnología y bases de datos.
- Se proporcionan prestaciones de reparación y recuperación enriquecidas incluyendo reintentos automáticos, reparación manual, compensación y almacén y reenvío.
  - Soporte para altos niveles de posibilidad de gestión y rendimiento con estas características:
    - Process Center proporciona un repositorio central escalable y un centro de control para organizar y gestionar todos los artefactos de proceso, las aplicaciones y los servicios creados como parte de un programa BPM.
    - Los toolkits gestionan artefactos de proceso para reutilizarlos entre varias process applications.
    - Las snapshots capturan el estado de todos los artefactos en una process application o un toolkit (incluyendo diagramas de proceso de negocio, reglas, datos, formularios, servicios y escenarios de simulación) en un momento específico con una sola pulsación del ratón.
    - El mantenimiento de versiones restaura cualquier snapshot histórica de una process application o un toolkit.
    - Un registro de Process Server proporciona herramientas centralizadas para instalar y hacer un seguimiento de versiones desplegadas de varios de procesos entre varios entornos de servidor de tiempo de ejecución.
    - El gobierno del ciclo de vida proporciona un control centralizado para gobernar el despliegue de procesos y servicios en el tiempo de ejecución de producción.
  - IBM Business Process Manager ofrece tres configuraciones de producto (Advanced, Standard y Express) para ajustarse al punto de entrada de una compañía en la gestión de procesos de negocio. Para conocer los detalles, consulte "Configuraciones de IBM Business Process Manager 7.5" en este Information Center.
  - IBM Business Process Manager proporciona herramientas de creación mejoradas y complementos opcionales para la integración con herramientas de productividad de usuario final:
    - Herramienta de diseño de procesos basada en estándares IBM Process Designer que forma parte del entorno de creación básico para las tres configuraciones de IBM Business Process Manager (Advanced, Standard y Express). La herramienta de diseño de procesos permite una composición rápida y un cambio de procesos continuo.
    - Entorno de creación de configuración avanzada IBM Integration Designer para construir visualmente servicios, transformaciones de datos, procesos BPEL e integración en aplicaciones y sistemas de fondo. El entorno de creación se utiliza en colaboración con la herramienta de diseño de procesos para crear soluciones de proceso robustas y escalables. Incluye un conjunto de adaptadores exhaustivo para habilitar los activos para servicio, incluyendo aplicaciones empaquetadas, personalizadas y heredadas, protocolos de tecnología y bases de datos. El entorno de creación es totalmente compatible con las versiones más recientes de WebSphere Integration Developer.
    - El complemento IBM Business Process Manager for Microsoft Office es un complemento opcional para las tres configuraciones (Advanced, Standard y Express). Los usuarios de Microsoft Office pueden utilizar IBM Business Process Manager para ver y ejecutar directamente tareas de IBM BPM desde su escritorio de Office. El complemento de Office instala un plug-in de IBM en el cliente del escritorio de Office de usuario final.
    - El complemento IBM Business Process Manager for Microsoft SharePoint es un complemento opcional para las tres configuraciones (Advanced, Standard y Express). Los usuarios de Microsoft SharePoint pueden utilizar IBM Business Process Manager para ver y ejecutar directamente componentes web que se sueltan en las páginas de portal de SharePoint. El complemento de SharePoint se instala en cualquier servidor SharePoint que se utilizará para albergar interacciones con procesos a través de los componentes web de IBM BPM.
  - IBM Business Process Manager V7.5 soporta las migraciones y las importaciones de datos siguientes:
    - Migración de process applications desde WebSphere Process Server V6.1, V6.1.2, V6.2 y V7.0, incluyendo activos de definición de procesos de tiempo de diseño e instancias de proceso en curso.

- Migración de process applications de Lombardi Teamworks 6.1, Teamworks 6.2, Teamworks 7.0 y WebSphere Lombardi Edition V7.1 y V7.2, incluyendo activos de definición de procesos de tiempo de diseño e instancias de proceso en curso.
- Importación de modelos de proceso en tiempo de diseño en formato BPMN 2.0 desde WebSphere Business Modeler V7.0, WebSphere Business Compass V7.0 y otras herramientas de modelado populares.
- Los widgets de IBM BPM Advanced para Business Space se pueden utilizar con IBM WebSphere Portal.

## **Novedades de Business Space powered by WebSphere**

Business Space un componente común de Business Process Manager y otros productos de IBM, se ha mejorado:

- Características nuevas de Business Space powered by WebSphere Versión 7.5
- Novedades de Business Space Versión 7.5 para WebSphere Portal Versión 7.0

## **Accesibilidad en IBM Business Process Manager**

Las funciones de accesibilidad ayudan a los usuarios con discapacidades físicas, como por ejemplo movilidad restringida o visión limitada, a utilizar satisfactoriamente los productos de tecnología de la información.

IBM se esfuerza por proporcionar productos con acceso que todos los usuarios puedan utilizar, independientemente de la edad o capacidad. Utilizar tecnologías asistenciales, como por ejemplo el software lector de pantalla y el sintetizador de voz digital para utilizar lo que se muestra en la pantalla. Consulte la documentación del producto de la tecnología asistencial para conocer los detalles acerca de cómo utilizar estas tecnologías con este producto.

Puede utilizar todas las características mediante el teclado en lugar del ratón.

Puede personalizar atributos de visualización como por ejemplo el color, el contraste y el tamaño del font.

Puede agrandar la información presentada en las vistas gráficas a fin de ver más detalle.

Puede solicitar una plantilla U.S. Section 508 Voluntary Product Accessibility Template (VPAT) en el sitio web de IBM en [http://www.ibm.com/able/product\\_accessibility/index.html](http://www.ibm.com/able/product_accessibility/index.html).

La documentación de Information Center incluye las siguientes características adicionales que facilitan la accesibilidad:

- La documentación está disponible en formato HTML para facilitar a los usuarios la aplicación de la tecnología del software lector de pantalla.
- Las imágenes de la documentación tienen texto alternativo para que los usuarios con disminuciones visuales puedan utilizar el contenido de las imágenes.

## **Disponibilidad de idiomas nacionales en IBM Business Process Manager**

IBM Business Process Manager soporta los lenguajes siguientes. Es posible que la documentación no esté totalmente traducida.

- Chino simplificado
- Chino tradicional
- Checo
- Inglés EE.UU.
- Francés

- Alemán
- Húngaro
- Italiano
- Japonés
- Coreano
- Polaco
- Portugués de Brasil
- Rumano
- Ruso
- Español

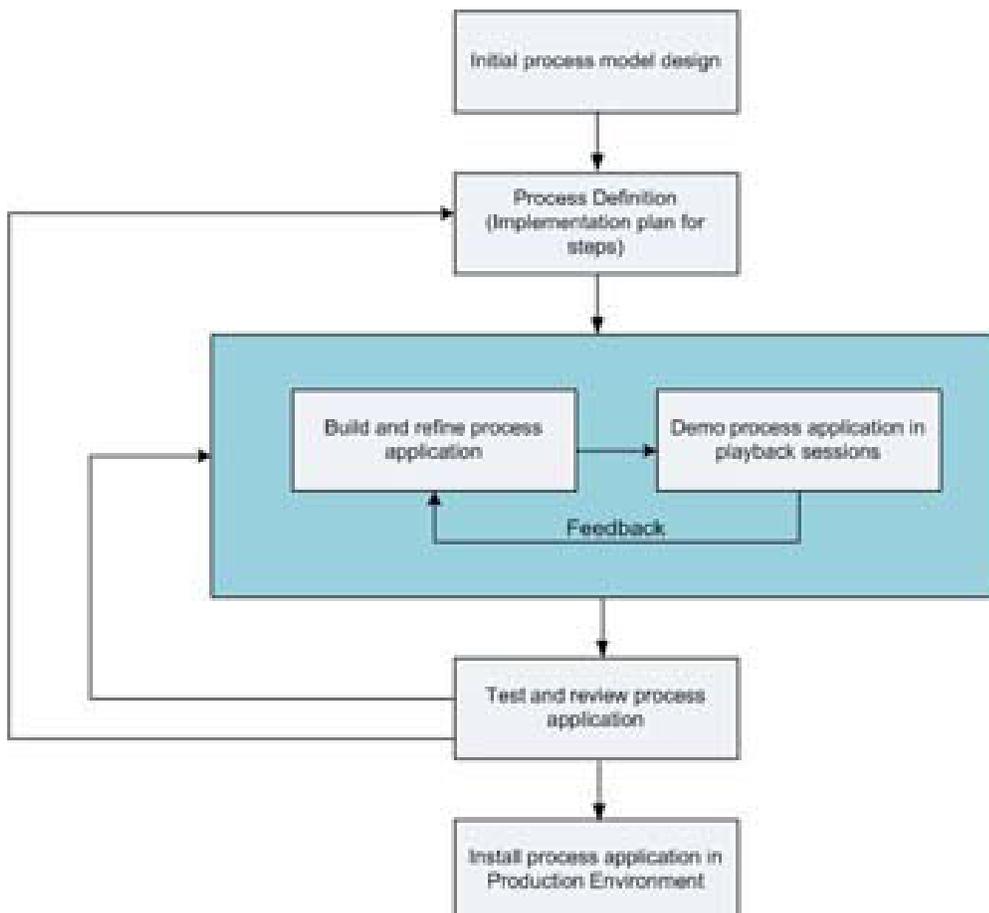
---

## Visión general de la gestión de procesos de negocio

Cuando desarrolla procesos en Process Designer, necesita planificar la instalación de aplicaciones de proceso en servidores de los entornos de prueba y producción.

El diagrama siguiente muestra el ciclo de vida de una tarea típica de desarrollo de procesos. El diagrama incluye pasos para crear y refinar un servicio de instalación a fin de poder instalar process applications en el entorno de producción.

Tal como muestra este diagrama, puede trabajar exclusivamente en el entorno de desarrollo. Pero necesita configurar Process Servers para los entornos de prueba y producción.



## Visión general del modelado de procesos

Un proceso es la unidad lógica principal en IBM Business Process Manager. Es el contenedor de todos los componentes de una definición de proceso, incluidos servicios, actividades y puntos de acceso; sucesos de temporizador, mensaje y excepción; líneas de secuencia, reglas y variables. Cuando modela un proceso, está creando una definición de proceso de negocio (BPD) reutilizable.

Los componentes de proceso le permiten definir el flujo de trabajo del proceso para usuarios finales, creando la lógica dentro de un proceso y realizando la integración con otras aplicaciones y fuentes de datos. Para comprender lo que ocurre dentro de un proceso durante la ejecución, es importante que conozca los componentes que forman un proceso durante el diseño.

## Creación de procesos de IBM BPM

En el desarrollo de procesos mediante IBM BPM suelen intervenir muchas personas diferentes pertenecientes a organizaciones diferentes. La cuestión principal es crear la mejor solución posible para cumplir los objetivos establecidos del proyecto. Para asegurar resultados satisfactorios, los miembros del equipo deben trabajar conjuntamente para obtener los requisitos del proceso y desarrollar iterativamente el modelo y sus implementaciones.

## Reutilización de elementos en Process Designer

Process Designer permite que los desarrolladores de procesos reutilicen elementos existentes de una o varias aplicaciones de proceso. Por ejemplo, si sabe que ya existen varios servicios que incluyen Coaches y otros elementos compartidos que usted y otros desarrolladores necesitan, puede acceder y reutilizar esos elementos mediante su inclusión en un toolkit. Luego, desde la aplicación de proceso, puede añadir una dependencia al toolkit donde residen los elementos compartidos. Esto le permite seleccionar uno de los servicios existentes al elegir la implementación para una actividad. Los elementos del toolkit pueden también ser utilizados por otros desarrolladores que trabajan en aplicaciones de proceso diferentes.

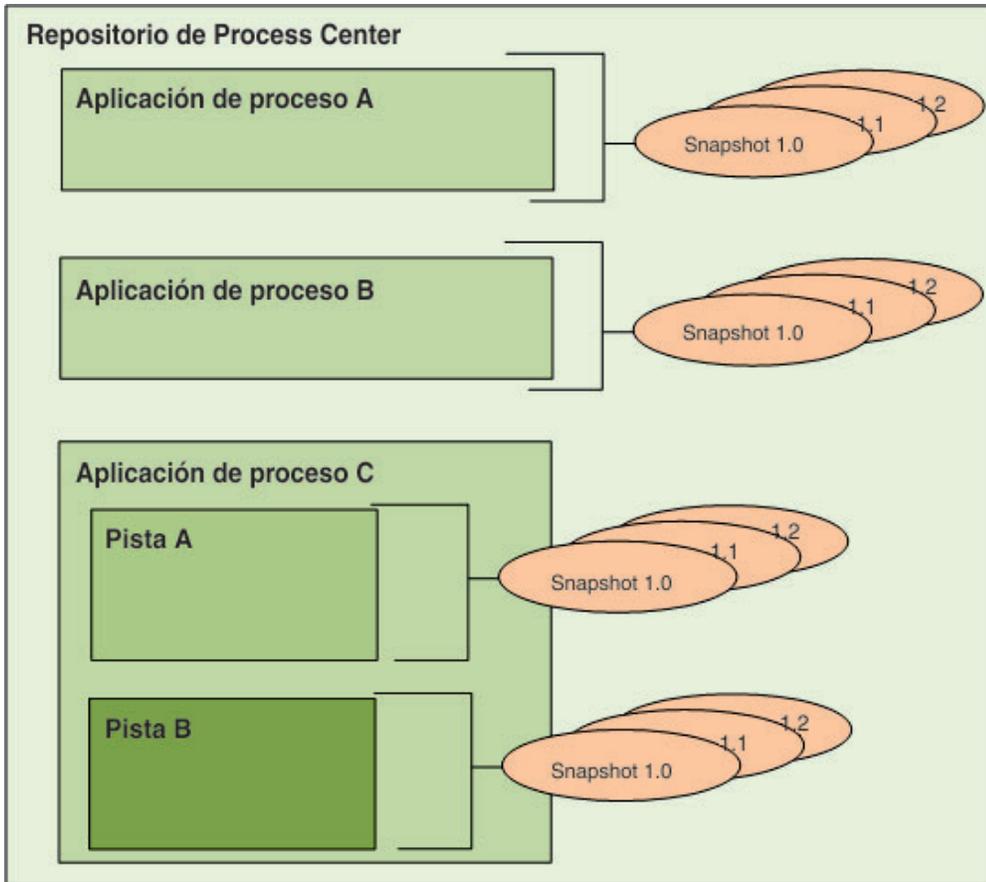
## Utilización de Designer en IBM Process Designer

La interfaz de Designer proporciona las herramientas que necesita para modelar los procesos en IBM BPM.

## Desarrollo de procesos con Process Center

IBM Process Center actúa como repositorio central para todos los activos de proyecto creados en Process Designer. Cuando varios clientes de Process Designer se conectan a Process Center, los usuarios pueden compartir elementos, tales como procesos y servicios, y también puede ver los cambios realizados por otros usuarios a medida que se producen. Process Center también se puede utilizar como un repositorio para los activos creados en IBM Integration Designer.

Cuando desarrolla procesos en Process Designer, existe una jerarquía en el repositorio de Process Center que está diseñada para ayudarle a gestionar los proyectos. La figura siguiente proporciona una visión general conceptual de la jerarquía del repositorio:



Como puede ver en el diagrama anterior, el repositorio de Process Center incluye los artefactos siguientes:

Aplicaciones de proceso	Contenedores para los modelos de proceso y las implementaciones de soporte que los analistas de gestión de procesos de negocio y los desarrolladores crean en el Diseñador de IBM Process Designer.
Pistas	Subdivisiones opcionales establecidas en una aplicación de proceso que están basadas en tareas de equipo o versiones de aplicaciones de proceso. Cuando están habilitadas, las pistas permiten realizar desarrollo en paralelo. Los administradores determinan si son necesarias pistas adicionales y, por lo tanto, si están habilitadas para cada aplicación de proceso.
Snapshots	Registran el estado de los elementos de una process application o pista en un punto específico en el tiempo. Normalmente las snapshots representan un punto de seguimiento o se utilizan para la reproducción o para la instalación. Puede comparar snapshots y revertir a snapshots anteriores. Si un administrador habilita las pistas para una aplicación de proceso, se utiliza una snapshot como base para una nueva pista.

## Aplicaciones de proceso: visión general

Una aplicación de proceso es un contenedor para modelos de proceso y sus implementaciones de soporte. Las aplicaciones de proceso están almacenadas en el repositorio. Una vez creados los artefactos, se ensamblan para formar una aplicación de proceso y se obtiene una snapshot. Puede probar, instalar y administrar la snapshot de la aplicación de proceso.

Las aplicaciones de proceso contienen algunos de los artefactos siguientes o todos ellos:

- Uno o varios modelos de proceso, también denominados Definiciones de proceso de negocio (Business Process Definition, BPD)

- Los servicios necesarios para implementar actividades o combinar con otros sistemas
- Módulos SCA (Service Component Architecture)
- Toolkits
- Bibliotecas de IBM Integration Designer
- Uno o varios espacios de trabajo
- Un modelo de IBM Business Monitor para supervisar el rendimiento de negocio.
- Cualquier otro elemento necesario para ejecutar el proceso

## Aplicaciones de proceso y aplicaciones de nivel de negocio

Cada aplicación de proceso tiene una aplicación de nivel de negocio (BLA), que actúa como contenedor para la aplicación de proceso y sus activos, los cuales incluyen modelos de supervisión, módulos SCA, toolkits y bibliotecas. Además, cada snapshot de una aplicación de proceso tiene su propia aplicación de nivel de negocio. Muchas de las tareas de administración que se realizan para una snapshot (tales como detener o iniciar una snapshot en un servidor de producción) se realizan a nivel de la aplicación de nivel de negocio, lo que permite una administración más rápida y simple de la snapshot y de todos sus recursos.

## Ejecución y depuración de procesos con Inspector

El Inspector de IBM Process Designer es la clave para realizar un desarrollo iterativo de procesos. Mediante el Inspector, los desarrolladores pueden ejecutar procesos y servicios en el servidor de Process Center o en los servidores remotos de ejecución Process Server. Además, todo un equipo de desarrollo puede utilizar Inspector para demostrar el diseño y la implementación de proceso actual en sesiones de reproducción. Las sesiones de reproducción ayudan a capturar información importante de distintas partes interesadas en un proceso, como por ejemplo la dirección, los usuarios finales y los analistas de negocio. Si adopta un enfoque iterativo al desarrollo de procesos, se asegura de que sus aplicaciones de proceso cumplan los objetivos y las necesidades de todas las personas implicadas.

El Inspector de IBM Process Designer incluye varias herramientas que le permiten ejecutar tareas tales como las siguientes en cada uno de los entornos configurados:

Tarea	Descripción
Gestionar instancias de procesos	Cuando ejecuta un proceso, puede ver todas las instancias de proceso ejecutadas anteriormente y actualmente en ejecución en los servidores de IBM Business Process Manager del entorno. Puede gestionar instancias en ejecución deteniéndolas y a continuación reanudándolas, por ejemplo. También puede gestionar instancias ejecutadas anteriormente filtrando o suprimiendo registros específicos.
Seguir paso a paso un proceso y depurarlo	Para una instancia seleccionada, vea el paso en ejecución actualmente y a continuación avance en el proceso, evaluando la ejecución del proceso paso a paso. Una visualización en árbol del proceso en combinación con los indicadores denominados señales en el diagrama de proceso facilita comprender en qué punto del proceso se encuentra. También tiene la ventaja de ver las variables utilizadas en cada paso y sus valores correspondientes (donde corresponda).

Consulte los temas siguientes para obtener información sobre cómo utilizar la interfaz de Inspector:

Para obtener información sobre cómo...	Consulte...
Gestionar instancias ejecutadas anteriormente y en ejecución actualmente del proceso en el servidor que seleccione.	Gestión de instancias de proceso
Seguir paso a paso la ejecución del proceso para asegurarse de que la BPD funcione de la forma esperada.	Seguimiento paso a paso de un proceso

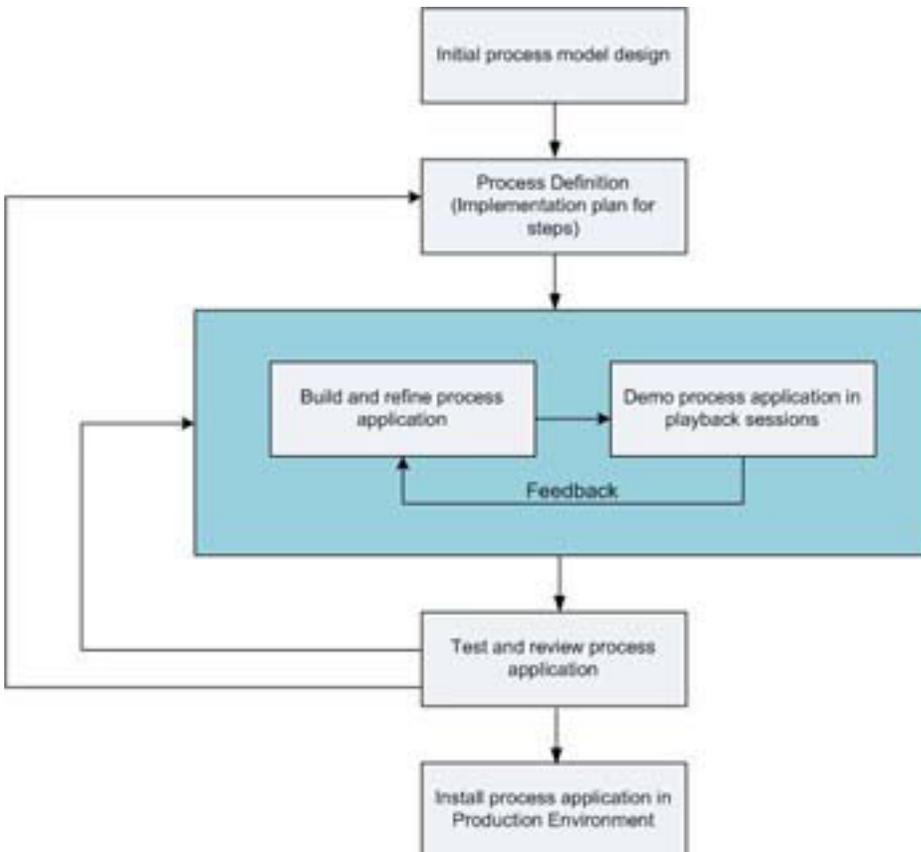
Para obtener información sobre cómo...	Consulte...
Examinar cada proceso o servicio subyacente en cada paso de la ejecución del proceso para realizar una inspección más detallada de la que simplemente proporciona seguir paso a paso el proceso.	Depuración de un proceso
Buscar fácilmente el origen de los errores generados al ejecutar la BPD y resolverlos.	Resolución de errores
Acceder y utilizar cada característica proporcionada por Inspector.	Referencia de Inspector

## Despliegue y gestión de aplicaciones de proceso

El ciclo de vida de la aplicación de proceso incluye desplegar y retirar el despliegue de snapshots y gestionar estados de snapshot. La creación de versiones también forma parte del ciclo de vida.

Cuando desarrolla procesos, puede sacar provecho del estilo iterativo utilizado por las herramientas de Process Designer. Los procesos evolucionan con el tiempo, inicialmente desde un estado de desarrollo, luego al estado de prueba y finalmente al estado de producción. Incluso en el estado de producción, los procesos pueden seguir evolucionando debido a necesidades cambiantes. Es importante estar preparado para el ciclo de vida progresivo de los procesos, lo que le ayudará a diseñar de manera eficaz desde el principio.

La figura siguiente muestra un enfoque iterativo al desarrollo de procesos:



Una configuración típica de Business Process Manager incluye tres entornos para permitir el desarrollo y despliegue final de los procesos.

Entorno	Descripción
Desarrollo	Cree y refine las aplicaciones de proceso en IBM Process Designer. Cree los modelos de proceso e implemente los pasos en estos modelos mediante Designer. Mediante Inspector, demuestre el progreso del desarrollo en sesiones de reproducción de forma que pueda evaluar y refinar rápidamente el prototipo. Utilizando la consola de Process Center, despliegue las aplicaciones de proceso en los entornos de prueba y producción.
Prueba	Utilizando la consola de Process Center, despliegue las aplicaciones de proceso en el servidor Process Server en el entorno de prueba para implementar pruebas formales de control de calidad. Puede utilizar Inspector como ayuda para verificar y solucionar problemas.
Producción	Una vez resueltos todos los problemas notificados por la etapa de prueba formal, utilice la consola de Process Center para desplegar las aplicaciones de proceso en el servidor Process Server del entorno de producción. Puede utilizar Inspector para investigar y solucionar los problemas notificados en el entorno de producción.

## Estrategias de publicación y entrega

Para asegurarse de que las aplicaciones de proceso que implementa y despliega cumplen las normas de calidad de su empresa, puede definir una estrategia de publicación y entrega. Cuando haya identificado los objetivos y requisitos para la publicación y entrega de aplicaciones de proceso nuevas y actualizadas, puede automatizar los procesos necesarios para aprobar e iniciar los programas.

Por ejemplo, puede encaminar un proceso hacia varios gestores diferentes a través de diversas estructuras jerárquicas de su empresa. Sólo después de que cada gestor apruebe el proceso nuevo o actualizado, se puede desplegar en el entorno de producción y entregar a los usuarios finales. Puede crear e implementar en IBM Business Process Manager Advanced los pasos que intervienen en dicha revisión para asegurarse de que todas las directrices corporativas se han satisfecho y que tiene las firmas necesarias. El paso final de la revisión podría ser la notificación al equipo de tecnologías de la información de que la aplicación de proceso aprobada está preparada para el despliegue.

Las secciones siguientes proporcionan información sobre el despliegue de los procesos y la gestión de estos una vez desplegados.

## Creación, acceso e incorporación de servicios

### Acceso a servicios externos a una aplicación

En este caso de ejemplo se tratan las distintas formas de acceder a servicios que son externos a la aplicación, y se proporcionan tareas de alto nivel para el acceso a estos servicios externos.

**Nota:** Esta situación es aplicable a WebSphere Enterprise Service Bus y IBM Business Process Manager. Los módulos de mediación se pueden desplegar en WebSphere Enterprise Service Bus y IBM Business Process Manager. Los módulos se pueden desplegar en IBM Business Process Manager.

En una aplicación de negocio integrada, los *servicios de negocio* interactúan entre ellos para proporcionar una función necesaria. Un servicio de negocio realiza una función o tarea repetitiva que contribuye a lograr un objetivo de negocio. Pero el trabajo de localizar un servicio y conectar no está relacionado con la función de negocio. La separación de la función de negocio de la tarea de gestionar las conexiones de servicio proporciona flexibilidad a la solución.

La interacción de servicio comienza cuando un *petionario de servicios* envía una petición a un *proveedor de servicios* para realizar una función de negocio. Esta petición se envía en forma de *mensaje*, que define la función a realizar. El proveedor de servicios realiza la función solicitada y envía el resultado en un mensaje al petionario del servicio. Por lo general, hay que procesar los mensajes para permitir que los servicios intercambien datos y para implementar otras funciones de TI a bajo nivel que son

independientes de las funciones y datos de negocio. Por ejemplo, direccionamiento, conversión de protocolo, transformación, reintento de una invocación errónea e invocación de servicio dinámico. Este proceso se conoce como *mediación*.



Existen dos tipos de módulos en IBM Integration Designer: módulos (o módulos de integración de negocio), que están pensados principalmente para contener lógica empresarial (tal como procesos de negocio, reglas de negocio y máquinas de estado de negocio), y módulos de mediación, que implementan flujos de mediación. Aunque existe un cierto solapamiento de función entre los dos tipos de módulos, en general, se recomienda que la lógica empresarial esté aislada en módulos de negocio, y la lógica de mediación sea realizada por módulos de mediación.

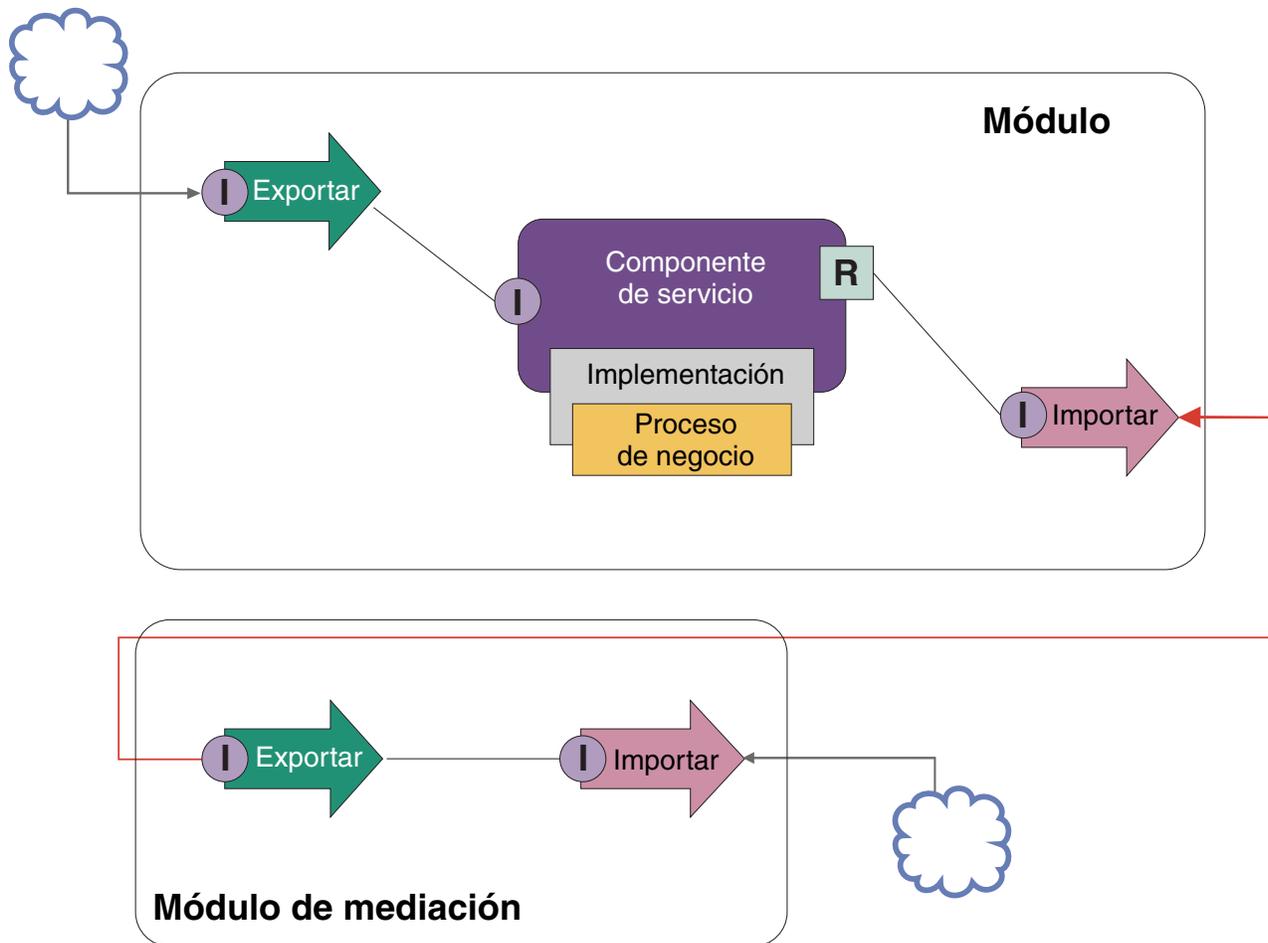
Pero no siempre hay una clara separación entre la lógica empresarial y la lógica de mediación. En estos casos, habrá que tener en cuenta la cantidad de *estado*, o datos en variables que tendrán que ser procesadas entre las invocaciones de servicio. En general, si se necesita poco (o ningún) proceso de estado, plantéese utilizar un componente de flujo de mediación. Si tiene que almacenar estado entre invocaciones de servicio, o si tiene datos que se tendrán que almacenar en variables y ser procesados, quizás sea mejor utilizar un componente de proceso de negocio. Por ejemplo, si está invocando varios servicios y grabando la información que estos devuelven porque, tras la invocación de todos los servicios, quiera realizar procesos adicionales con los datos devueltos, utilice un proceso de negocio en el que pueda fácilmente asignar la información devuelta a variables. En otras palabras, cuando tenga demasiados estados, habrá traspasado la línea a la lógica empresarial.

No hay un escenario de integración, y no hay una respuesta técnicamente equivocada. Las guías que aquí tratamos son métodos recomendados para permitir flexibilidad y reutilización, y se ofrecen como propuesta para que el usuario las valore. Como de costumbre, es el usuario el que debe valorar las ventajas e inconvenientes de la implementación de dichos patrones para sus aplicaciones de integración de negocio. Veamos algunas situaciones.

### **Acceso a un componente SCA**

Un ejemplo básico de acceso a un servicio es cuando una importación llama a otro componente SCA, sin precisar transformación en los datos. Incluso en esta situación, podría acceder al servicio externo desde un módulo de mediación, en vez de acceder directamente desde un módulo de negocio. Esto permitiría flexibilidad en el futuro para cambiar el punto final del servicio, la calidad del servicio o sus características (por ejemplo, añadir registro cronológico) sin que los componentes de negocio que consumen el servicio se vean afectados. Este patrón de arquitectura se conoce como "separación de problemas".

Antes de decidir implementar este patrón, valore los beneficios del patrón frente a los efectos potenciales de sobrecarga introducidos por otro módulo. Si su principal objetivo es la flexibilidad y va a hacer cambios frecuentes en los servicios a los que se accede, plantéese el uso de un módulo aparte, como se muestra aquí. Si lo más importante fuera el rendimiento y quiere actualizar y volver a desplegar la lógica empresarial, podría utilizar un único módulo.



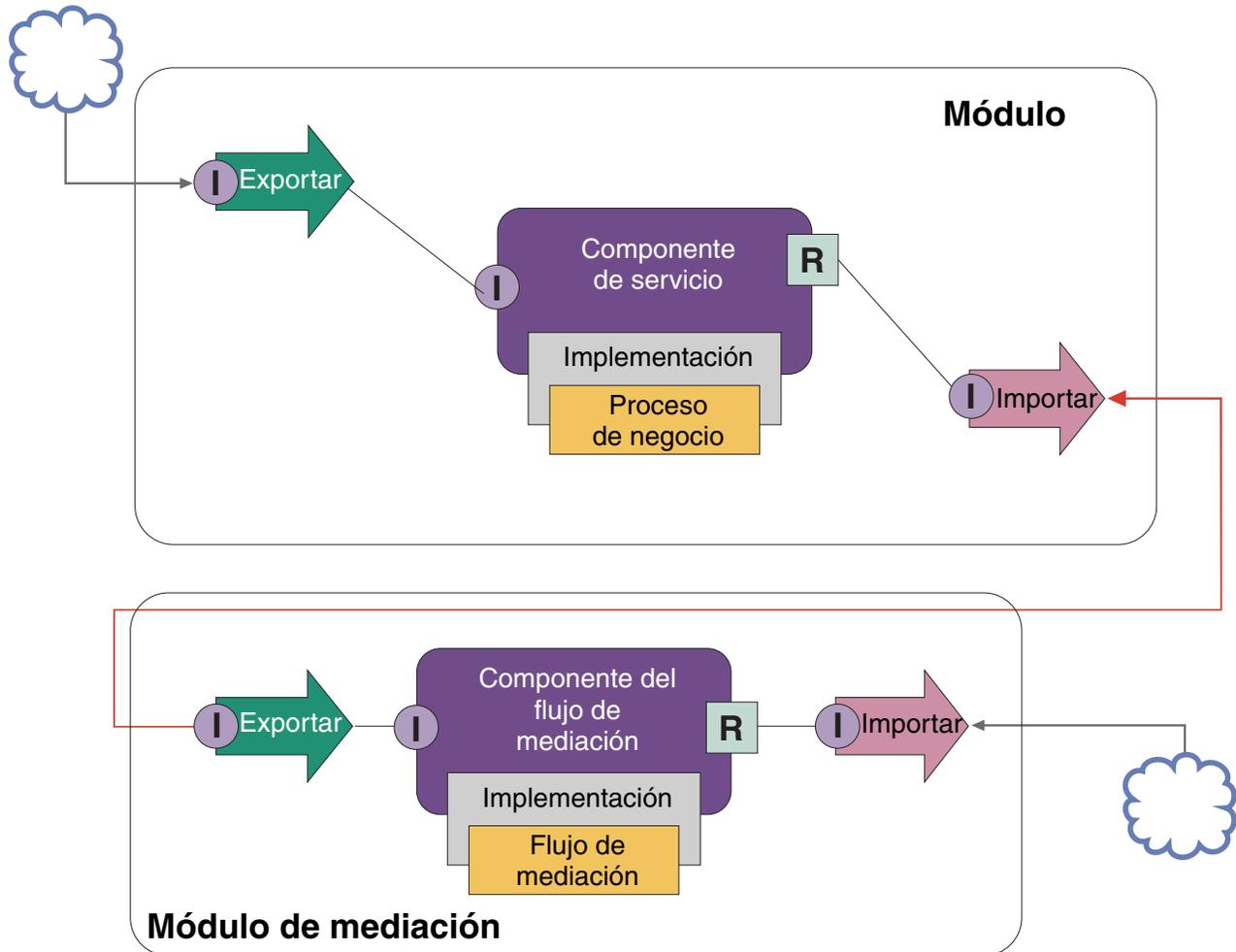
Estas son las tareas de alto nivel para conseguir este ejemplo.

1. Crear un módulo de mediación. Para instrucciones paso a paso, consulte Creación de módulos de mediación .
2. En el módulo de mediación, cree una importación con el enlace apropiado para el servicio externo al que quiera acceder. Para instrucciones paso a paso, consulte Creación de importaciones . Para obtener más información sobre los enlaces, consulte Enlaces
3. Cree una exportación y déle la misma interfaz que la importación. Para instrucciones paso a paso, consulte Creación de exportaciones .
4. Genere un enlace SCA para la exportación. Para instrucciones paso a paso, consulte Generar enlaces SCA .
5. En el ensamblaje del módulo de mediación, conecte la exportación con la importación. Guarde el módulo de mediación.
6. Crear un módulo. Para instrucciones paso a paso, consulte Creación de un módulo para servicios de negocio .
7. Añada una exportación y un componente.
8. En la vista Integración de negocio, arrastre la exportación que ha creado en el módulo de mediación (en el paso 4) al ensamblaje de módulo. Se creará una importación con el mismo enlace que la exportación.
9. Conecte la exportación con el componente, y el componente con la importación.
10. Añada la implementación del componente. Para obtener información sobre los tipos de implementación, consulte Implementaciones

Más adelante podrá añadir lógica de mediación, como un registro cronológico o el direccionamiento al módulo de mediación sin afectar al módulo de negocio.

## Añadir mediación

A veces no es suficiente simplemente invocar un servicio externo. Puede que sea necesario realizar primero un proceso, añadiendo un módulo de mediación como intermediario entre el peticionario y el proveedor del servicio.



Estas son algunas de las funciones que el flujo de mediación intermediario podría realizar:

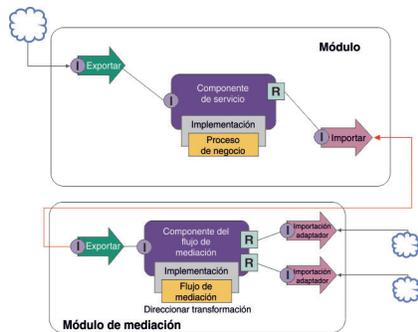
- Establecer cabeceras de protocolos. Para obtener más información, consulte el tema Conversión de protocolos en el Information Center de WebSphere Enterprise Service Bus.
- Transformación de parámetros o interfaz, mediante el uso de una primitiva de Transformación XSL o Correlación de objetos de negocio. Transformación de mensajes
- Seleccionar un servicio concreto de una lista estática, utilizando una primitiva Filtro de mensajes. Filtro de mensajes
- Invocación de varios servicios para acumular los resultados, utilizando las primitivas Diseminación y Abanico de entrada. Agregar y difundir mensajes
- Tratar los errores de invocación de servicio reintentando el servicio o invocando otro diferente, mediante el uso de una primitiva Invocación de servicio. Reintentar una invocación de servicio errónea

- Direccinamiento dinámico, eligiendo el servicio a utilizar en tiempo de ejecución, en vez de en tiempo de integración, lo que permite que los servicios estén menos cohesionados y los negocios reaccionen más rápidamente a los cambios. Se pueden añadir servicios nuevos sin tocar los módulos que se han desplegado en el entorno de ejecución. El direccionamiento dinámico es más productivo cuando se utiliza con un registro, lo que precisa del uso de una primitiva de mediación de búsqueda de punto final. Selección dinámica de puntos finales

## Acceso a Sistemas de información de empresa (Enterprise Information Systems)

Los servicios y artefactos situados en sistemas externos se pueden importar a Integration Designer. Un asistente descubre aplicaciones y datos en los sistemas de información de empresa (EIS) y permite generar servicios a partir de las aplicaciones y datos descubiertos. Los artefactos generados son interfaces y objetos de negocio, que los componentes de un módulo pueden utilizar.

El uso de un módulo de mediación intermedio entre un módulo y un sistema principal hace que sea más reutilizable. En el ejemplo siguiente, se utiliza un flujo de mediación para direccionar al sistema principal correcto y para transformar los datos en el formato necesario para el sistema principal.



Estas son las tareas de alto nivel para este ejemplo:

1. Utilizar el asistente de servicios externos para conectar al sistema principal. El uso del asistente de servicios externos para acceder a servicios externos sigue un patrón similar, independientemente del adaptador que utilice. Para obtener información sobre el uso del asistente de servicio externo, consulte Patrón para el acceso a servicios externos con adaptadores
2. Crear un módulo. Para instrucciones paso a paso, consulte Creación de un módulo para servicios de negocio .
3. Añadir una exportación, un componente y una importación con enlace SCA. Para obtener más información, consulte Llamada a servicios
4. Añadir una interfaz a la exportación, y conectar dicha exportación con el componente.
5. Añada la implementación del componente. En la implementación, establezca una propiedad que indique el servicio de sistema principal al que se accederá. Para obtener información sobre los tipos de implementación, consulte Implementaciones
6. Crear un módulo de mediación con una exportación que tenga un enlace de SCA y la misma interfaz que la importación del módulo creado en el paso 2.
7. Conectar la exportación a un componente de flujo de mediación.
8. Crear una importación para cada sistema principal al que quiera acceder, utilizando el adaptador de salida adecuado desde la paleta del editor de ensamblaje.
9. Conectar el componente de flujo de mediación a las importaciones.
10. Implementar el componente del flujo de mediación. Utilice una primitiva Filtro de mensajes para elegir la importación basada en un conjunto de propiedades en la lógica empresarial y utilice una primitiva de transformación XSL para cada importación de adaptador. Filtro de mensajes.

11. En el módulo, seleccione la exportación del módulo de mediación como el servicio a importar en el módulo. Para obtener información paso a paso, consulte Invocación de un servicio desde otro módulo.

Más adelante podrá realizar cambios como añadir un adaptador o cambiar un adaptador para que apunte a un sistema principal distinto, con un impacto mínimo en la lógica empresarial.

## Acceso a sistemas de mensajería

Para que el módulo de Arquitectura de componente de servicio (SCA) se comunique con JMS, MQ o un cliente de mensajería MQ JMS, tiene que crear interfaces, objetos de negocio y enlaces para importaciones y exportaciones. Consulte Correlación de un mensaje con una interfaz SCA.

Los flujos de mediación utilizan mensajes, que proporcionan acceso a información de contexto y cabecera, además de a objetos de negocio. Si quiere acceder a información de cabecera JMS o a una propiedad de JMS personalizada, utilice un flujo de mediación. Si está realizando una integración con un sistema MQ y quiere acceder a la información de cabecera de MQ, utilice un flujo de mediación.

## Crear o llamar a un servicio Web

Los servicios Web son aplicaciones autocontenidas que realizan funciones de negocio, que van desde una simple consulta a interacciones complejas de procesos de negocio. Puede llamar a un servicio Web existente o puede desarrollar un servicio web nuevo para que se ajuste a sus necesidades. Este caso de ejemplo describirá los pasos y le señalará información adicional.

Aunque no podrá crear todos los servicios desde cero utilizando IBM Integration Designer, algunos de los servicios se crearán de esta forma. Cuando trabaje con el editor de ensamblaje y el editor de proceso de negocio para ensamblar servicios en un proceso de negocio, es probable que detecte que faltan algunos servicios. Por tanto puede ser útil crear esos servicios que faltan utilizando herramientas de IBM Integration Designer. Lo contrario también se cumple: una vez creado un proceso nuevo, puede decidir que sería útil exponer todas o algunas operaciones del proceso como servicios para consumo de otros.

**Nota:** Esta situación es aplicable a usuarios de IBM Integration Designer para IBM Process Server y WebSphere Enterprise Service Bus.

Existen varias razones para desarrollar servicios web utilizando IBM Integration Designer:

- La creación de servicios en IBM Integration Designer le permite implementar el servicio utilizando reglas de negocio.
- El desarrollo en IBM Integration Designer le permite desarrollar un servicio Java y exponerlo como servicio web y a través de SCA.
- La correlación de interfaces sin tener que crear código es una ventaja. Puede sacar todas las correlaciones de datos del código Java, dejando un simple programa Java a modo de caja negra para el desarrollador de Java.
- IBM Integration Designer muestra todos los servicios y relaciones en un único lugar.
- La capacidad de refactorizar también ayuda en el desarrollo de servicios web mediante IBM Integration Designer.

Tenga en cuenta que los servicios Web no deben considerarse la solución a todos los problemas de integración. Sin embargo, al igual que cualquier otro procedimiento de tecnología o arquitectura, existen ventajas inherentes a la utilización de los servicios Web en el lugar y el momento adecuados.

## Exportaciones, importaciones y enlaces

IBM Integration Designer le permite importar servicios web estándar y utilizar estos servicios en aplicaciones compuestas.

En IBM Integration Designer, utilice el editor de ensamblaje para desarrollar servicios. Siga el proceso estándar para crear módulos, módulos de mediación, bibliotecas y componentes. A continuación, puede utilizar exportaciones, importaciones y enlaces para compartir y acceder a dichos servicios. A continuación se indican los pasos para dichas tareas básicas y los enlaces que conducen a información más detallada de cada tarea.

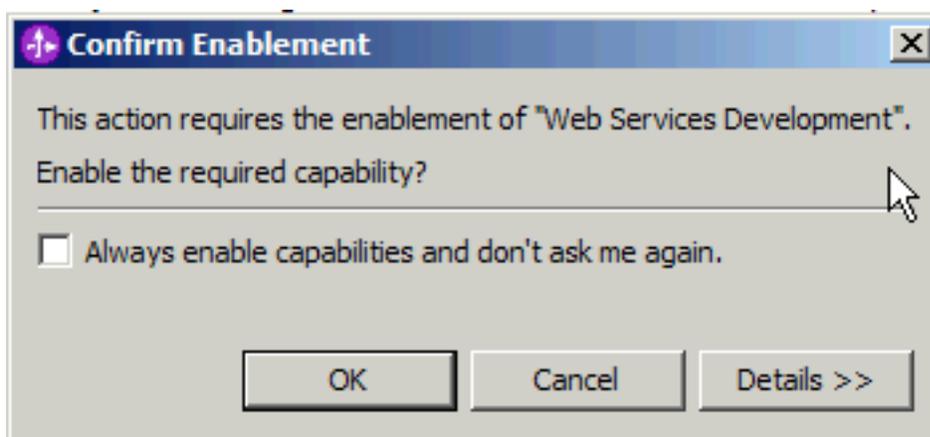
Puede utilizar cualquiera de los dos enlaces para los servicios Web: un enlace de servicio Web o un enlace HTTP. Un enlace de servicio web proporciona una especificación para la transmisión de mensajes hacia y desde un servicio web. Las herramientas ayudan a generar automáticamente un enlace de servicio web. Un enlace HTTP es un protocolo estándar de petición y respuesta entre clientes y el servidor, según se define en el protocolo HTTP publicado por el consorcio World Wide Web (W3C). Necesitará proporcionar alguna información inicial de configuración de enlace si utiliza un enlace HTTP.

1. Crear una exportación para publicar el servicio de módulo para su uso por parte de otros módulos.
2. Genere un enlace para la exportación.
  - Generar un enlace de servicio web para la exportación.
  - Generar un enlace de exportación HTTP.
3. Crear una importación para llamar a un servicio existente que no sea parte del módulo que está ensamblando.
  - Generar un enlace de servicio web para la importación.
  - Generar un enlace de importación HTTP.

Lea el tema de enlace si quiere invocar un servicio web desde JavaServer Pages.

### Posibilidades de desarrollo de servicios web

Cuando abre un editor asociado al proceso de creación de servicios Web, podría encontrarse con el error siguiente:



IBM Integration Designer proporciona una función de filtrado conocida como *prestaciones*. En los valores de preferencias, las funciones y herramientas están categorizadas en prestaciones, y puede habilitar e inhabilitar categorías de prestaciones o un subconjunto de funciones de cualquier categoría. Consulte la sección Prestaciones para obtener más información.



---

## Capítulo 2. Aprender más acerca de IBM Business Process Manager

Utilice esta sección como punto de partida para investigar las tecnologías utilizadas en y por IBM Business Process Manager

---

### Mantenimiento de versiones

El ciclo de vida de una aplicación de procesos comienza con la creación de la aplicación de procesos y continúa con un ciclo de actualización, despliegue, despliegue conjunto, anulación de despliegue y archivado de la aplicación de procesos. El *mantenimiento de versiones* es un mecanismo utilizado para gestionar el ciclo de vida de la aplicación de proceso mediante la identificación exclusiva de las versiones individuales de la aplicación de proceso.

La forma en que el mantenimiento de versiones trabaja en IBM Business Process Manager depende del tipo de aplicación que esté desplegando: una aplicación de proceso desplegada desde el repositorio en IBM Process Center, o una aplicación empresarial desplegada directamente desde IBM Integration Designer.

Las process applications y los toolkits que se despliegan en un entorno de ejecución del Process Center están sujetos de forma predeterminada al mantenimiento de versiones. Para las aplicaciones empresariales, puede elegir crear versiones de módulos y bibliotecas en IBM Integration Designer.

Además, puede crear versiones de una tarea de usuario o de una máquina de estado, de forma que las distintas versiones de la tarea o de la máquina de estado puedan coexistir en el entorno de ejecución.

### Creación de versiones de aplicaciones de proceso

La creación de versiones es la capacidad del entorno de ejecución de identificar snapshots en el ciclo vital de una aplicación de proceso y poder ejecutar simultáneamente varias snapshots al mismo tiempo.

Puede considerar la aplicación de proceso como un contenedor. Las snapshots, el despliegue y la creación de versiones se gestionan todos a nivel de contenedor, no a nivel de los artefactos contenidos en el contenedor. Las snapshots se gestionan desde la consola de Process Center.

Los cambios se guardan dinámicamente en el repositorio de Process Center por sugerencia de la pista de trabajo. La process application permanece en ese nivel de sugerencia hasta que decida crear una snapshot (sn1). Una process application puede desplegarse en un servidor Process Center o en un Process Server con fines de prueba, transferencia o producción.

Si realiza cambios y desea desplegar una nueva versión, debe crear una snapshot nueva (sn2). Puede eliminar sn1 o dejarla en ejecución en el servidor cuando despliega sn2.

### Contexto de versión

El contexto de versión son los metadatos que identifican una versión. El usuario asigna ese identificador, pero IBM recomienda utilizar un sistema de números de versión de tres dígitos, con el formato <major>.<minor>.<service>. Consulte los temas sobre los convenios de denominación para obtener una descripción más detallada de este sistema de creación de versiones.

IBM Business Process Manager asigna un espacio de nombres global para cada aplicación de proceso. El espacio de nombres global es específicamente el extremo de la aplicación de proceso o una snapshot específica de la aplicación de proceso. El nombre de versión utilizado por el servidor no puede ser más

largo que siete caracteres, por lo que el nombre asignado será un acrónimo que contiene caracteres del nombre de snapshot asignado. Los acrónimos de snapshot serán iguales a sus nombres de snapshot si éstos siguen el formato recomendado VRM (versión-release-modificación) de IBM y no tienen más de siete caracteres. Por ejemplo, el nombre de snapshot 1.0.0 tendrá el acrónimo 1.0.0, y el nombre de snapshot 10.3.0 tendrá el acrónimo 10.3.0. El acrónimo de snapshot es exclusivo dentro del contexto de la aplicación de proceso en el ámbito del servidor de Process Center. Por este motivo, no puede editar el acrónimo de snapshot.

## **Creación de versiones de process applications y toolkits de Process Designer**

Para crear versiones de process applications y toolkits almacenados en el repositorio de Process Center, puede guardar y designar snapshots. Esto le permite comparar una snapshot con otra para encontrar diferencias. Por ejemplo, si un desarrollador ha arreglado un problema de un servicio y ha realizado una snapshot de la process applications o el toolkit contenedor en ese punto y, a continuación, otro desarrollador ha realizado varios otros cambios en el mismo servicio y ha realizado otra snapshot, el gestor del proyecto podría comparar las dos snapshots para determinar qué cambios se han realizado y quién los ha realizado. Si el gestor de proyectos ha decidido que los cambios adicionales en el servicio no eran necesarios, el gestor de proyectos puede revertir a la snapshot del arreglo original.

Normalmente puede tomar una snapshot de una process application cada vez que está listo para desplegar en producción o para probar la integración. Para desplegar en un servidor de proceso autónomo debe tomar una snapshot de la process application. Para los toolkits, la cosa cambia ligeramente; puede tomar una snapshot de un toolkit cuando esté listo para que las process applications utilicen este toolkit. Después, si desea actualizar el toolkit, debe tomar otra snapshot de la "sugerencia" cuando esté listo y entonces los propietarios de las process applications y los toolkits pueden decidir si desean moverse a la snapshot nueva. La sugerencia es una snapshot especial y la única clase de snapshot en la que puede cambiar el contenido, pero sólo puede ejecutarla en el servidor de Process Center. No puede desplegar una sugerencia en Process Server.

## **Aplicaciones de procesos en varios clústeres**

Puede desplegar la misma versión de una process application en varios clústeres dentro de la misma célula. Para diferenciar entre esos varios despliegues de la misma versión de la process application, cree una snapshot para cada despliegue e incluya un ID exclusivo de célula en el nombre de snapshot (por ejemplo, v1.0\_cell1\_1 y v1.0\_cell1\_2). Hablando estrictamente, cada snapshot es una versión nueva de la process application (desde una perspectiva de gestión de ciclo de vida pura), pero el contenido y la función son iguales.

Cuando despliega una aplicación en un clúster, se realiza una sincronización automática de los nodos.

## **Mantenimiento de versiones de módulos y bibliotecas**

Si un módulo o biblioteca está contenido en una process application o un toolkit, toma el ciclo de vida de la process application o el toolkit (versiones, snapshots, pistas, etc.). Los nombres de los módulos y bibliotecas deben ser exclusivos dentro del ámbito de una aplicación de procesos o kit de herramientas.

En este tema se describe el mantenimiento de las versiones de los módulos y las bibliotecas que se utilizan con process applications. Observe, sin embargo, que si despliega módulos directamente desde IBM Integration Designer para Process Server, puede continuar para siguiendo el procedimiento de asignación de números de versión a los módulos durante el despliegue, tal como se describe en "Creación de módulos y bibliotecas versionadas".

Un módulo o biblioteca que esté asociado al IBM Process Center debe tener sus bibliotecas dependientes en la misma aplicación de procesos o en un kit de herramientas dependiente.

En la tabla siguiente se enumeran las selecciones que puede hacer en el editor de dependencias en IBM Integration Designer cuando una biblioteca está asociada a una process application o un toolkit:

Tabla 3. Dependencias para bibliotecas de Módulo, Process App o Toolkit y Global

Ámbito de biblioteca	Descripción	Puede depender de . . .
Módulo	En el servidor ya existe una copia de esta biblioteca para cada módulo que la utiliza.	Una biblioteca de ámbito de módulo puede depender de todos los tipos de bibliotecas.
Process App (aplicación de procesos) o Toolkit (kit de herramientas)	La biblioteca es compartida por casi todos los módulos del ámbito de la aplicación de procesos o del kit de herramientas. Esta configuración entra en vigor si el despliegue se realiza mediante el IBM Process Center. Si el despliegue se produce fuera del IBM Process Center, la biblioteca se copia en cada módulo. <b>Nota:</b> Las bibliotecas creadas en IBM Integration Designer versión 7.5 tienen un nivel de compartimiento de <b>Process App o Toolkit</b> de forma predeterminada.	Una biblioteca de este tipo sólo puede depender de las bibliotecas globales.
Global	La biblioteca la comparten todos los módulos que se estén ejecutando.	Una biblioteca global solo puede depender de otras bibliotecas globales. <b>Nota:</b> Debe configurar una biblioteca compartida de WebSphere para desplegar la biblioteca global. Consulte "Dependencias de módulos y bibliotecas" para obtener más información.

## Convenio de denominación

Un convenio de denominación se utiliza para diferenciar las diferentes versiones de una process application proceso mientras pasa por el ciclo de vida de actualización, despliegue, despliegue conjunto, retirada de despliegue y archivado.

En esta sección se proporcionan las convenciones que se utilizan para identificar de forma exclusiva las versiones de una aplicación de procesos.

Un *contexto de versión* es una combinación de acrónimos que describen de forma exclusiva una aplicación de procesos o un kit de herramientas. Cada tipo de acrónimo tiene un convenio de denominación. El acrónimo está limitado a una longitud máxima de siete caracteres del conjunto de caracteres [a-z0-9\_], excepto el acrónimo de snapshot, que también puede incluir un punto.

- El acrónimo de la aplicación de procesos se crea cuando se crea la aplicación de procesos. Puede tener una longitud máxima de siete caracteres.
- El acrónimo de la instantánea se crea automáticamente cuando se crea la instantánea. Puede tener una longitud máxima de siete caracteres.

Si el nombre de instantánea cumple los criterios de un acrónimo de instantánea válido, el nombre de la instantánea y el acrónimo será el mismo.

**Nota:** Cuando se utiliza una función de ruteado version-aware para componente de flujo de mediación, deberá nombrar su instantánea de forma que se ajuste al esquema `<versión>.<release>.<modificación>` (por ejemplo, **1.0.0**). Dado que el acrónimo de la instantánea está limitado a siete caracteres, el valor de los dígitos está limitado a un máximo de cinco dígitos en total (cinco dígitos más dos puntos). Por lo tanto, deberá tener cuidado cuando aumente el campo de los dígitos, ya que todo lo que supere los primeros siete caracteres se verá truncado.

Por ejemplo, el nombre de instantánea **11.22.33** da como resultado un acrónimo de instantánea **11.22.3**.

- El acrónimo de seguimiento se genera automáticamente a partir del primer carácter de cada palabra del nombre de la pista. Por ejemplo, un seguimiento nuevo creado con el nombre de **Mi seguimiento nuevo** daría como resultado un acrónimo con el valor **MSN**.

El nombre del seguimiento predeterminado y el acrónimo son **Principal**. El despliegue en un IBM Process Center incluye el acrónimo de pista en el contexto de la versión si el acrónimo del seguimiento no es **Principal**.

Una definición de proceso de negocio en una aplicación de procesos se identifica normalmente por el acrónimo de la aplicación de procesos, el acrónimo de la instantánea y el nombre de la definición del proceso de negocio. Elija nombres exclusivos para las definiciones de proceso de negocio siempre que sea posible. Cuando existen nombres duplicados, podría encontrarse con los problemas siguientes:

- Es posible que no pueda exponer las definiciones de proceso de negocio como servicios web sin algún tipo de mediación.
- No podrá invocar una definición de proceso de negocio creada en IBM Process Designer desde un proceso BPEL creado en IBM Integration Designer.

El contexto de versión varía, dependiendo de la forma en que se despliegue la aplicación de procesos.

### **Convenios de denominación para despliegues de servidor de Process Center**

En el servidor IBM Process Center, puede desplegar una snapshot de una process application, así como una snapshot de un toolkit. Además, puede desplegar la sugerencia de una process application o la sugerencia de un toolkit. (Una *sugerencia* es la versión de trabajo actual de la process application o del toolkit.) El contenido de la versión variará en función del tipo de despliegue.

En el caso de aplicaciones de procesos, la sugerencia de aplicaciones de procesos o la instantánea concreta de la aplicaciones de procesos sirve para identificar la versión de forma exclusiva.

Los toolkits se pueden desplegar con una o más aplicaciones de procesos, pero el ciclo de vida de cada toolkit está enlazado con el ciclo de vida de la aplicación de procesos. Cada aplicación de proceso tiene su propia copia del toolkit o los toolkits dependientes desplegados en el servidor. Un toolkit desplegado no está compartido entre aplicaciones de proceso.

Si el seguimiento asociado a la aplicación de procesos lleva un nombre diferente al predeterminado **Principal**, el acrónimo del seguimiento también será parte del contexto de la versión.

### **Instantáneas de aplicación de procesos**

En el caso de despliegues de instantáneas de aplicaciones de procesos, el contexto de la versión sería una combinación de los siguientes elementos:

- Acrónimo de la aplicación de procesos
- Acrónimo del seguimiento de la aplicación de procesos (si se utiliza un seguimiento distinto a **Principal**)
- Acrónimo de la instantánea de la aplicación de procesos

### **Kits de herramientas autónomo**

En los despliegues de instantáneas de kits de herramientas, el contexto de la versión es una combinación de los siguientes elementos:

- Acrónimo del nombre del kit de herramientas
- Acrónimo del seguimiento del kit de herramientas (si se utiliza un seguimiento distinto a **Principal**)
- Acrónimo de instantánea del kit de herramientas

## Sugerencias

Las sugerencias de las aplicaciones de procesos se utilizan durante las pruebas iterativas en Process Designer. Sólo se pueden desplegar en servidores de Process Center.

En los despliegues de sugerencias de aplicación de procesos, el contenido de la versión es una combinación de los siguientes elementos:

- Acrónimo de la aplicación de procesos
- Acrónimo del seguimiento de la aplicación de procesos (si se utiliza un seguimiento distinto a **Principal**)
- "Sugerencia"

Las sugerencias de los kits de herramientas se utilizan durante las pruebas iterativas en Process Designer. No se despliegan en un servidor de producción.

En los despliegues de sugerencias de kits de herramientas, el contexto de la versión es una combinación de los siguientes elementos:

- Acrónimo del nombre del kit de herramientas
- Acrónimo del seguimiento del kit de herramientas (si se utiliza un seguimiento distinto a **Principal**)
- "Sugerencia"

## Ejemplos

Los recursos se deben nombrar e identificar de forma exclusiva empleando el contenido de la versión.

- La tabla siguiente muestra un ejemplo de nombres que se identifican de forma exclusiva. En este ejemplo, una sugerencia de process application utiliza el nombre de la pista predeterminada (**Principal**):

Tabla 4. Sugerencia de la process application con el nombre de la pista predeterminada

Nombre de la aplicación de procesos	<b>Aplicación de procesos 1</b>
Acrónimo de la aplicación de procesos	<b>PA1</b>
Seguimiento de la aplicación de procesos	<b>Principal</b>
Acrónimo de seguimiento de la aplicación de procesos	"" (cuando el seguimiento es <b>Principal</b> )
Instantánea de la aplicación de procesos	
Acrónimo de la instantánea de la aplicación de procesos	<b>Sugerencia</b>

Cualquier módulo SCA asociado a esta sugerencia de process application incluye el contexto de versión, tal como se muestra en la tabla siguiente:

Tabla 5. Módulos SCA y archivos EAR que tienen en cuenta la versión

Nombre del módulo SCA	Nombre que tiene en cuenta la versión	Nombre de EAR/aplicación que tiene en cuenta la versión
M1	PA1-Sugerencia-M1	PA1-Sugerencia-M1.ear
M2	PA1-Sugerencia-M2	PA1-Sugerencia-M2.ear

- La tabla siguiente muestra un ejemplo de una sugerencia de una process application que utiliza un nombre de pista no predeterminado:

Tabla 6. Sugerencia de la process application con el nombre de la pista no predeterminado

Nombre de la aplicación de procesos	<b>Aplicación de procesos 1</b>
-------------------------------------	---------------------------------

Tabla 6. Sugerencia de la process application con el nombre de la pista no predeterminado (continuación)

Acrónimo de la aplicación de procesos	<b>PA1</b>
Seguimiento de la aplicación de procesos	<b>Rastreo1</b>
Acrónimo de seguimiento de la aplicación de procesos	<b>T1</b>
Instantánea de la aplicación de procesos	
Acrónimo de la instantánea de la aplicación de procesos	<b>Sugerencia</b>

Cualquier módulo SCA asociado a esta sugerencia de process application incluye el contexto de versión, tal como se muestra en la tabla siguiente:

Tabla 7. Módulos SCA y archivos EAR que tienen en cuenta la versión

Nombre del módulo SCA	Nombre que tiene en cuenta la versión	Nombre de EAR/aplicación que tiene en cuenta la versión
M1	PA1-T1-Sugerencia-M1	PA1-T1-Sugerencia-M1.ear
M2	PA1-T1-Sugerencia-M2	PA1-T1-Sugerencia-M2.ear

## Convenios de denominación para despliegues de Process Server

En el Process Server, puede desplegar la snapshot de una process application. El acrónimo de la aplicación de procesos sirve para identificar la versión de forma exclusiva.

En los despliegues de instantáneas de aplicación de procesos, el contenido de la versión es una combinación de los siguientes elementos:

- Acrónimo de la aplicación de proceso
- Acrónimo de la instantánea de la aplicación de procesos

Los recursos se deben nombrar e identificar de forma exclusiva empleando el contenido de la versión. La tabla siguiente muestra un ejemplo de nombres que se identifican de forma exclusiva:

Tabla 8. Ejemplo de nombres y acrónimos

Nombre de la aplicación de proceso	Aplicación de procesos 1
Acrónimo de la aplicación de procesos	<b>PA1</b>
Instantánea de la aplicación de procesos	<b>1.0.0</b>
Acrónimo de la instantánea de la aplicación de procesos	<b>1.0.0</b>

Un recurso, como un módulo o una biblioteca, tiene el contexto de la versión como parte de su identificador.

La tabla siguiente muestra un ejemplo de dos módulos y cómo los archivos EAR asociados incluyen el contexto de la versión.

Tabla 9. Módulos SCA y archivos EAR que tienen en cuenta la versión

Nombre del módulo SCA	Nombre que tiene en cuenta la versión	Nombre de EAR/aplicación que tiene en cuenta la versión
M1	PA1-1.0.0-M1	PA1-1.0.0-M1.ear
M2	PA1-1.0.0-M2	PA1-1.0.0-M2.ear

En la tabla siguiente se muestra un ejemplo de dos bibliotecas de ámbito de process application y cómo los archivos JAR asociados incluyen el contexto de versión:

Tabla 10. Bibliotecas de ámbito de process application y archivos JAR que tienen en cuenta la versión

Nombre de biblioteca de ámbito de process application de SCA	Nombre que tiene en cuenta la versión	Nombre de JAR que tiene en cuenta la versión
Bib1	PA1-1.0.0-Bib1	PA1-1.0.0-Bib1.jar
Bib2	PA1-1.0.0-Bib2	PA1-1.0.0-Bib2.jar

## Enlaces que tienen en cuenta la versión

Las aplicaciones de proceso pueden contener módulos SCA que incluyan enlaces de importación y exportación. Cuando despliega aplicaciones de forma conjunta, el enlace para cada versión de la aplicación debe ser único. Algunos enlaces se actualizan automáticamente durante el despliegue, para asegurar la exclusividad entre versiones. En otros casos, tiene que actualizar el enlace después del despliegue, para asegurar que son únicos.

Un enlace *version-aware* (que tenga en cuenta la versión) tiene como ámbito una versión particular de una aplicación de proceso, que garantiza su exclusividad entre aplicaciones de proceso. En las secciones siguientes se describen los enlaces que se actualizan automáticamente a *version-aware* (que tienen en cuenta la versión) así como las acciones que necesite realizar en tiempo de ejecución cuando un enlace no es *version-aware*. Para obtener información sobre aspectos a considerar al crear módulos, consulte “Consideraciones sobre la utilización de enlaces”.

## SCA

El destino de un enlace SCA se cambia de nombre automáticamente para que sea *version-aware* durante el despliegue si los enlaces de importación y exportación del módulo están definidos en el mismo ámbito que la aplicación de proceso.

Si los enlaces no están definidos en el mismo ámbito de aplicación de proceso, se registra un mensaje informativo. Debe modificar el enlace de importación tras el despliegue para cambiar la dirección de destino del punto final. Puede utilizar la consola administrativa para cambiar la dirección de destino del punto final.

## Servicio web (JAX-WS o JAX-RPC)

La dirección de destino de punto final de un enlace de servicio web se renombra automáticamente para que sea *version-aware* durante el despliegue, si se cumplen todas las condiciones siguientes:

- Ha seguido las normas de denominación predeterminadas para la dirección:  
**`http://ip:puerto/NombreMóduloWeb/sca/NombreExportación`**
- La dirección de punto final es SOAP/HTTP.
- Los enlaces de importación y exportación del módulo están definidos en el mismo ámbito que la aplicación de proceso.

Si no se cumplen estas condiciones, se registra un mensaje informativo. La acción que realizará dependerá de cómo está desplegando su aplicación de proceso:

- Si está desplegando su aplicación de proceso de forma conjunta, debe cambiar el nombre manualmente al URL de punto final SOAP/HTTP o a la cola de destino SOAP/JMS para que sea único entre versiones de la aplicación de proceso. Puede utilizar la consola administrativa tras el despliegue para cambiar la dirección de destino del punto final.
- Si sólo está desplegando una versión única de la aplicación de proceso, puede ignorar este mensaje

Para el despliegue conjunto de snapshots de enlace de servicio web SOAP/ JMS, la acción que realice depende de cómo despliegue la process application:

- Si la importación y exportación de destino se encuentran en la misma process application, realice los pasos siguientes antes de publicar la process application en Process Center y crear la instantánea:
  1. Cambie el URL de punto final de la exportación. Asegúrese de que el destino y la fábrica de conexiones son exclusivos.
  2. Cambie el URL de punto final de la importación para que sea el mismo que el que ha especificado para la exportación en el paso anterior.
- Si la importación y la exportación de destino se encuentran en aplicaciones de proceso diferentes, siga estos pasos:
  1. Cambie el URL de punto final de la exportación. Asegúrese de que el destino y la fábrica de conexiones son exclusivos.
  2. Publicar la aplicación de proceso en Process Center.
  3. Cree la snapshot.
  4. Despliegue la process application en el Process Server.
  5. Utilice la consola administrativa de WebSphere para cambiar el URL de punto final de la importación correspondiente para que sea el mismo que el que se especifica para la exportación.

## HTTP

La dirección URL de punto final de un enlace HTTP se renombra automáticamente para que sea version-aware durante el despliegue, si se cumplen todas las condiciones siguientes:

- Ha seguido las normas de denominación predeterminadas para la dirección:  
**http (s)://ip:puerto/NombreMóduloWeb/rutaentradacontextoExportación**
- Los enlaces de importación y exportación del módulo están definidos en el mismo ámbito que la aplicación de proceso.

Si no se cumplen estas condiciones, se registra un mensaje informativo. La acción que realizará dependerá de cómo está desplegando su aplicación de proceso:

- Si está desplegando su aplicación de proceso de forma conjunta, debe cambiar el nombre manualmente al URL de punto final para que sea único entre versiones de la aplicación de proceso. Puede utilizar la consola administrativa tras el despliegue para cambiar la dirección de destino del punto final.
- Si sólo está desplegando una versión única de la aplicación de proceso, puede ignorar este mensaje

## JMS y JMS genérico

Los JMS generados por el sistema y los enlaces JMS genéricos tiene en cuenta la versión automáticamente.

**Nota:** Para que los enlaces tengan en cuenta la versión, no se produce un renombrado automático de los JMS definidos por el usuario y de los enlaces genéricos JMS. Cuando el enlace ha sido definido por el usuario, se deben renombrar los siguientes atributos para que sean únicos entre las versiones de la aplicación de proceso:

- Configuración de punto final
- Recibir cola de destino
- Nombre del puerto de escucha (si está definido)

Establecer el destino de Envío correspondiente si cambia el punto final del módulo de destino.

## MQ/JMS y MQ

No se produce un cambio de nombre automático durante el despliegue para habilitar que los enlaces de tipo MQ/JMS o MQ sean version-aware.

Debe cambiar el nombre de los atributos siguientes para que sean únicos entre versiones de la aplicación de proceso:

- Configuración de punto final
- Recibir cola de destino

Establecer el destino de Envío correspondiente si cambia el punto final del módulo de destino.

## EJB

No se produce un cambio de nombre automático durante el despliegue para habilitar que los enlaces de tipo EJB sean version-aware.

Debe cambiar el nombre del atributo Nombres JNDI siguiente para que sea único entre versiones de la aplicación de proceso.

Tenga en cuenta que las aplicaciones cliente también se deben actualizar para que utilicen los nuevos nombres JNDI.

## EIS

Un adaptador de recursos se renombra automáticamente para tener en cuenta la versión durante el despliegue siempre que el nombre de recurso predeterminado (*ModuleNameApp:Adapter Description*) no se haya modificado.

Si el nombre de recurso predeterminado se ha modificado, los nombres de adaptador de recursos deben ser exclusivos entre versiones de la process application.

Si los nombres de adaptador de recursos no son exclusivos, un mensaje informativo se anota cronológicamente durante el despliegue para avisarle. Puede renombrar manualmente los adaptadores de recursos después del despliegue utilizando la consola administrativa.

## Invocación dinámica que tiene en cuenta la versión

Puede configurar componentes de flujo de mediación para direccionar mensajes a puntos finales determinados dinámicamente en tiempo de ejecución. Al crear el módulo de mediación, puede configurar la búsqueda de puntos finales para utilizar un direccionamiento que tiene en cuenta la versión.

Si utiliza el estilo VRM (<versión>.<release>.<modificación>) para la snapshot, puede exportar el archivo EAR de process application a WebSphere Service Registry and Repository (WSRR). Al crear el módulo de mediación, configura la búsqueda de puntos finales para utilizar un direccionamiento que tiene en cuenta la versión. Por ejemplo, selecciona **Devolver el punto final que coincida con la última versión compatible de los servicios basados en módulos SCA** en el campo **Política de coincidencia** y seleccione **SCA** para **Tipo de enlace**.

Las versiones futuras de la process application se despliegan en el servidor y se publican en WSRR y búsqueda de puntos finales de módulo de mediación invoca dinámicamente la última versión compatible del punto final de servicio.

Tenga en cuenta que, como alternativa, puede establecer el destino en SMOHeader y el valor lo puede llevar el mensaje de solicitud.

## Despliegue de process applications con módulos Java y proyectos

Las process applications pueden contener módulos Java EE y proyectos Java personalizados. Cuando despliegue aplicaciones de forma conjunta, el módulo personalizado de Java EE para cada versión de la aplicación debe ser exclusivo.

Tenga en cuenta que los módulos Java EE y los proyectos Java personalizados se despliegan en un servidor si se despliegan con un módulo SCA que tiene una dependencia declaradas sobre ellos. Si no selecciona **Desplegar con módulo** (que es el valor predeterminado) al declarar la dependencia, debe desplegar el módulo o el proyecto manualmente.

## Despliegue de process applications con reglas de negocio y selectores

Si está desplegando varias versiones de una aplicación de proceso que incluyan una regla de negocio o un componente del selector, tenga en cuenta el modo en que los metadatos asociados se utiliza en las versiones.

Los metadatos dinámicos de una regla de negocio o componente selector los definen en el tiempo de ejecución el nombre del componente, el espacio de nombres de destino y el tipo de componente. Si dos o más versiones de una aplicación de procesos contienen una regla de negocio o selector se despliegan en el mismo entorno de ejecución, compartirán los mismos metadatos de la lógica de reglas (regla de negocio ) o direccionamiento.

Para habilitar cada versión de la norma de negocio o componente selector para que utilice sus propios metadatos dinámicos (lógica de reglas o direccionamiento), refactorice el espacio de nombres de destino de forma que sea exclusivo para cada versión de la aplicación de procesos.

---

## Enlaces

En la base de la arquitectura orientada a servicios se encuentra el concepto del *servicio*, una unidad de funcionalidad que se ejecuta mediante una interacción entre dispositivos de cálculo. Una *exportación* define la interfaz externa (o punto de acceso) de un módulo, de forma que los componentes SCA (Service Component Architecture) del módulo pueden proporcionar sus servicios a los clientes externos. Una *importación* define una interfaz para los servicios fuera de un módulo, de forma que los servicios puedan invocarse desde dentro del módulo. Los *enlaces* específicos del protocolo se utilizan con las importaciones y exportaciones para especificar el medio de transporte de los datos dentro o fuera del módulo.

## Exportaciones

Los clientes externos pueden invocar componentes SCA en un módulo de integración a través de una amplia variedad de protocolos (por ejemplo, HTTP, JMS, MQ y RMI/IIOP), con datos en distintos formatos (por ejemplo, XML, CSV, COBOL y JavaBean). Las exportaciones son componentes que reciben estas solicitudes de orígenes externos y, a continuación, invocan los componentes de IBM Business Process Manager utilizando el modelo de programación SCA.

Por ejemplo, en la siguiente figura, una exportación recibe una petición a través del protocolo HTTP de una aplicación de cliente. Los datos se transforman en un objeto de negocio, con el formato utilizado por el componente SCA. A continuación, el componente se invoca con dicho objeto de datos.

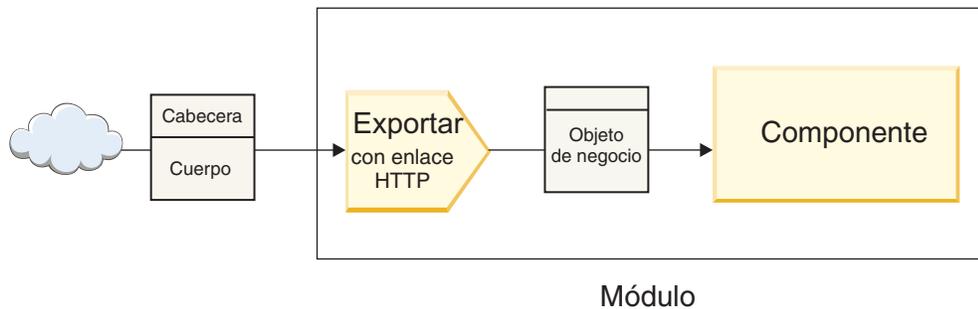


Figura 1. Una exportación con el enlace HTTP

## Importaciones

Un componente SCA puede desear invocar un servicio externo no SCA que espere datos en otro formato. El componente SCA utiliza una importación para invocar el servicio externo utilizando el modelo de programación SCA. A continuación, la importación invoca el servicio de destino de la forma que espera el servicio.

Por ejemplo, en la siguiente figura, la importación envía una petición desde un componente SCA a un servicio externo. El objeto de negocio, que tiene el formato utilizado por el componente SCA, se transforma en el formato esperado por el servicio y se invoca el servicio.

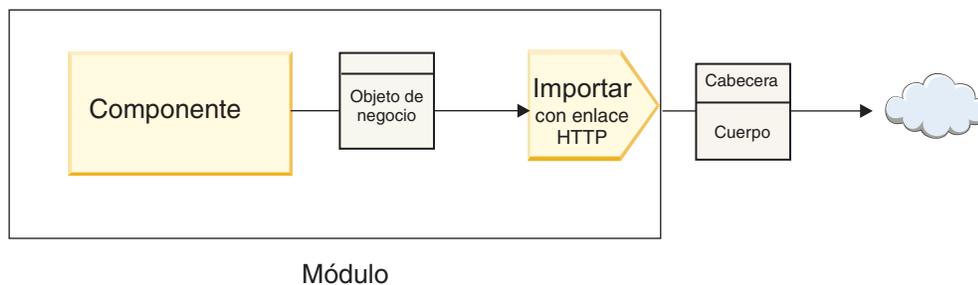


Figura 2. Una importación con el enlace HTTP

## Lista de enlaces

Utilice WebSphere Integration Developer para generar un enlace para una importación o una exportación, y para configurar el enlace. Los tipos de enlaces que hay disponibles se describen en la lista siguiente.

- SCA

El enlace SCA, que es el valor por omisión, permite al servicio comunicarse con servicios en otros módulos SCA. Utilice una importación con un enlace SCA para acceder a un servicio en otro módulo SCA. Utilice una exportación con un enlace SCA para ofrecer un servicio a otros módulos SCA.

- Servicio web

Un enlace de servicio Web permite acceder a un servicio externo mediante mensajes SOAP interoperables y calidades de servicio. También puede utilizar los enlaces de servicio web para incluir adjuntos como parte del mensaje SOAP.

El enlace de servicio Web puede utilizar un protocolo de transporte SOAP/HTTP (SOAP a través de HTTP) o SOAP/JMS (SOAP a través de JMS). Independientemente del transporte (HTTP o JMS) utilizado para transportar los mensajes SOAP, los enlaces de servicio web siempre manejan las interacciones de petición/respuesta de forma síncrona.

- HTTP

El enlace HTTP le permite acceder a un servicio externo mediante el protocolo HTTP, donde se utilicen mensajes que no sean SOAP, o donde se requiera acceso HTTP directo. Este enlace se utiliza cuando

está trabajando con servicios Web basados en el modelo HTTP (es decir, servicios que utilizan operaciones de interfaz HTTP conocidas como, por ejemplo, GET, PUT, DELETE, etc.).

- Enterprise JavaBeans (EJB)

Los enlaces EJB permiten a los componentes SCA interactuar con los servicios proporcionados por la lógica empresarial Java EE que se ejecuta en un servidor Java EE.

- EIS

El enlace EIS (sistema de información de empresa), cuando se utiliza con un adaptador de recursos JCA, permite acceder a los servicios en un sistema de información de empresa o hacer que los servicios estén disponibles para el EIS.

- Enlaces JMS

Los enlaces JMS (Java Message Service), JMS genéricos y MQ JMS (WebSphere MQ JMS) se utilizan para las interacciones con los sistemas de mensajería, donde la comunicación asíncrona mediante colas de mensajes es fundamental para la fiabilidad.

Una exportación con uno de los enlaces JMS observa una cola para ver la llegada de un mensaje y envía la respuesta de forma asíncrona, si la hay, a la cola de respuesta. Una importación con uno de los enlaces JMS crea y envía un mensaje a la cola JMS y observa una cola para ver la llegada de la respuesta, si la hay.

- JMS

El enlace JMS permite acceder al proveedor JMS incorporado de WebSphere.

- JMS genérico

El enlace JMS genérico permite acceder a un sistema de mensajería de un proveedor que no es IBM.

- MQ JMS

El enlace MQ JMS permite acceder al subconjunto JMS de un sistema de mensajería de WebSphere MQ. Utilice este enlace cuando el subconjunto JMS de funciones sea suficiente para la aplicación.

- MQ

El enlace de WebSphere MQ permite la comunicación con las aplicaciones MQ nativas, con lo cual se pueden llevar a la infraestructura SOA (arquitectura orientada a servicios) y se proporciona acceso a la información de cabecera específica de MQ. Utilice este enlace cuando necesite utilizar funciones nativas MQ.

## Visión general de los enlaces de exportación e importación

Una exportación permite crear servicios en un módulo de integración disponible para los clientes externos y una importación permite que los componentes SCA de un módulo de integración invoquen servicios externos. El enlace asociado con la exportación o la importación especifica la relación entre los mensajes de protocolo y los objetos de negocio. También especifica el modo en que se seleccionan las operaciones y los errores.

### Flujo de información a través de una exportación

Una exportación recibe una petición, que está dirigida para el componente al cual está conectada la exportación, a través de un transporte específico determinado por el enlace asociado (por ejemplo, HTTP).

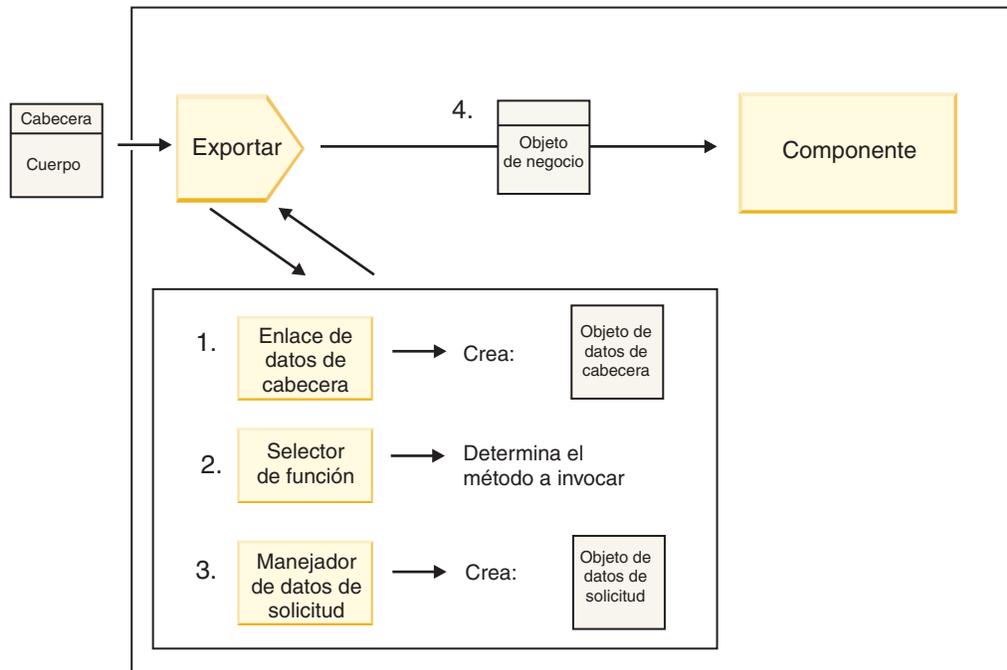


Figura 3. Flujo de una petición a través de la exportación a un componente

Cuando la exportación recibe la solicitud, se produce la secuencia de sucesos siguiente:

1. Sólo para los enlaces de WebSphere MQ, el enlace de datos de cabecera transforma la cabecera de protocolo en un objeto de datos de cabecera.
2. El selector de función determina el nombre de método nativo a partir del mensaje de protocolo. La configuración de exportación correlaciona el nombre del método nativo con el nombre de una operación efectuada en la interfaz de la exportación.
3. El manejador de datos de petición o el enlace de datos en el método transforma la petición en un objeto de negocio de petición.
4. La exportación invoca el método de componente con el objeto de negocio de solicitud.
  - El enlace de exportación HTTP, el enlace de exportación de servicio web y el enlace de exportación EJB invocan el componente SCA de forma síncrona.
  - Los enlaces de exportación JMS, JMS genéricos, MQ JMS, y de WebSphere MQ invocan el componente SCA de forma asíncrona.

Tenga en cuenta que una exportación puede propagar las cabeceras y las propiedades de usuario que recibe a través del protocolo, si se ha habilitado la propagación de contexto. Los componentes que están conectados con la exportación pueden acceder a estas cabeceras y propiedades de usuario. Para obtener más información, consulte el tema “Propagación” en el Centro de información de WebSphere Integration Developer.

Si es una operación bidireccional, el componente devuelve una respuesta.

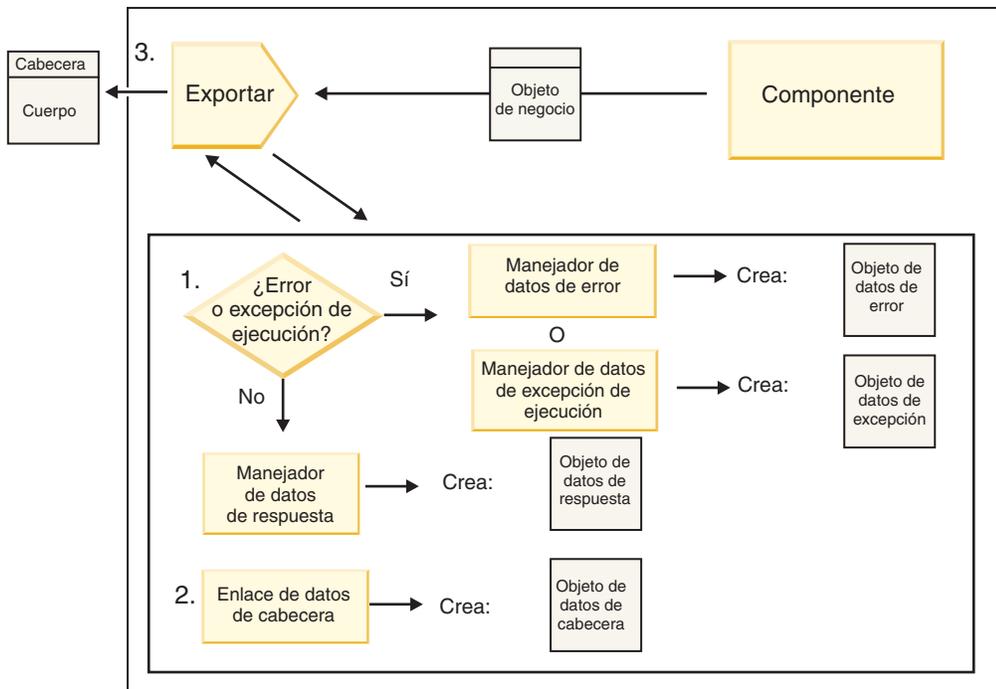


Figura 4. Flujo de retorno de una respuesta a través de la exportación

Se produce la secuencia de pasos siguiente:

1. Si el enlace de exportación recibe un mensaje de respuesta normal, el manejador de datos de respuesta o el enlace de datos en el método transforma el objeto de negocio en una respuesta. Si la respuesta es un error, el manejador de datos de error o el enlace de datos en el método transforma el error en una respuesta de error. Solamente en el caso de los enlaces de exportación HTTP, si la respuesta es una excepción de tiempo de ejecución, se invoca el manejador de datos de excepción de tiempo de ejecución, si está configurado.
2. Sólo para los enlaces de WebSphere MQ, el enlace de datos de cabecera transforma los objetos de datos de cabecera en cabeceras de protocolo.
3. La exportación envía la respuesta de servicio a través del transporte.

### Flujo de información a través de una importación

Los componentes envían solicitudes a los servicios situados fuera del módulo, mediante una importación. La petición se envía, a través de un transporte específico determinado por el enlace asociado.

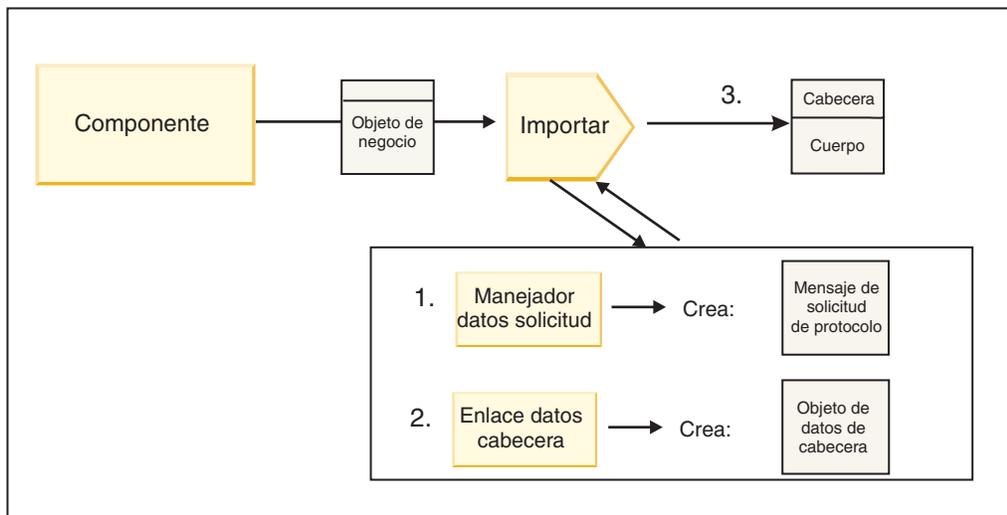


Figura 5. Flujo desde un componente a través de la importación a un servicio

El componente invoca la importación con un objeto de negocio de solicitud.

**Nota:**

- El componente que realiza la llamada debe invocar el enlace de importación HTTP, el enlace de importación de servicio web y el enlace de importación EJB de forma síncrona.
- Los enlaces de importación JMS, JMS genéricos, MQ JMS, y de WebSphere MQ deben invocarse de forma asíncrona.

Después de que el componente invoca la importación, se produce la siguiente secuencia de sucesos:

1. El manejador de datos de petición o el enlace de datos en el método transforma el objeto de negocio de petición en un mensaje de petición de protocolo.
2. Sólo para los enlaces de WebSphere MQ, el enlace de datos de cabecera en el método establece el objeto de negocio de cabecera en la cabecera de protocolo.
3. La importación invoca el servicio con la solicitud de servicio a través del transporte.

Si es una operación bidireccional, el servicio devuelve una respuesta y se produce la siguiente secuencia de pasos:

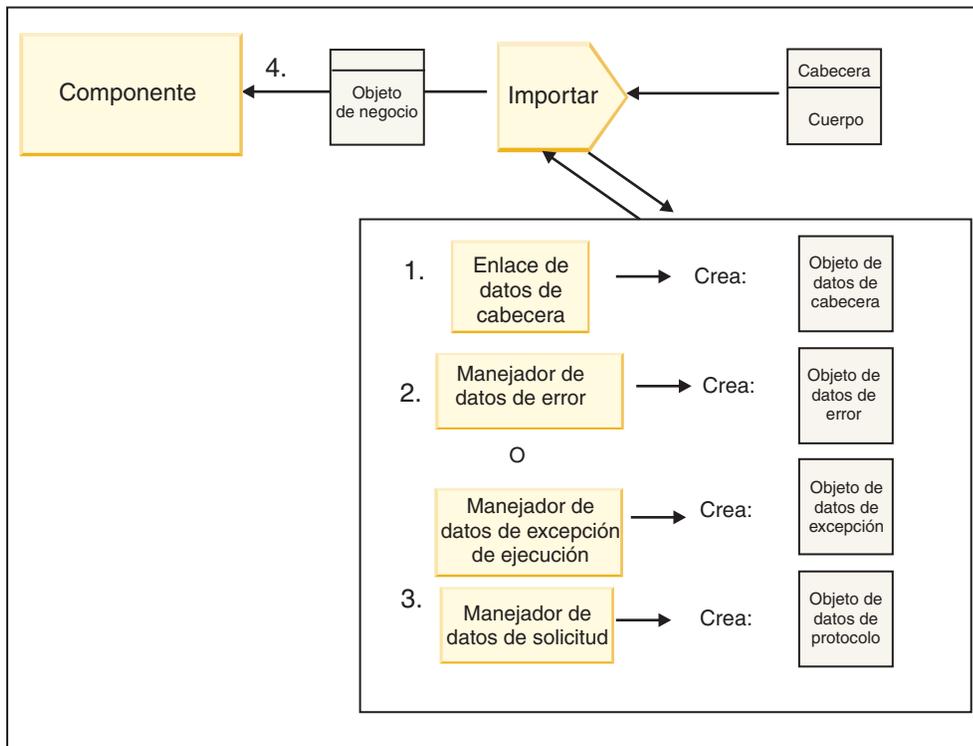


Figura 6. Flujo de retorno de una respuesta a través de la importación

1. Sólo para los enlaces de WebSphere MQ, el enlace de datos de cabecera transforma la cabecera de protocolo en un objeto de datos de cabecera.
2. Se determina si la respuesta es un error.
  - Si la respuesta es un error, el selector de error inspecciona el error para determinar con qué error de WSDL se correlaciona. A continuación, el manejador de datos de error en el método transforma el error en una respuesta de error.
  - Si la respuesta es una excepción de tiempo de ejecución, se invoca el manejador de datos de excepción de tiempo de ejecución, si está configurado.
3. El manejador de datos de respuesta o el enlace en el método transforma la respuesta en un objeto de negocio de respuesta.
4. La importación devuelve el objeto de negocio de respuesta al componente.

## Configuración de los enlaces de exportación e importación

Uno de los aspectos clave de los enlaces de exportación e importación es la transformación de datos, que indica cómo se correlacionan (deserializan) los datos desde un formato conectado nativo a un objeto de negocio, o cómo se correlacionan (serializan) desde un objeto de negocio a un formato conectado nativo. Para los enlaces asociados con las exportaciones, también puede especificar un selector de función para indicar qué operación debe ejecutarse en los datos. Para los enlaces asociados con las exportaciones o importaciones, puede indicar cómo deben manejarse los errores que se producen durante el proceso.

Asimismo, especifique información específica del transporte en los enlaces. Por ejemplo, para un enlace HTTP, especifique el URL de punto final. Para el enlace HTTP, la información específica del transporte se describe en los temas “Generación de un enlace de importación HTTP” y “Generación de un enlace de exportación HTTP”. Puede también encontrar información sobre otros enlaces en el Information Center.

## Transformación del formato de datos en importaciones y exportaciones

Cuando se configura un enlace de exportación o importación en WebSphere Integration Developer, una de las propiedades de configuración que especifica es el formato de los datos que utiliza el enlace.

- Para los enlaces de exportación, en los que una aplicación de cliente envía peticiones y recibe respuestas de un componente SCA, debe indicar el formato de los datos nativos. Dependiendo del formato, el sistema selecciona el manejador de datos o el enlace de datos correspondiente para transformar los datos nativos en un objeto de negocio (que utiliza el componente SCA) y, a la inversa, para transformar un objeto de negocio en datos nativos (que es la respuesta a la aplicación de cliente).
- Para los enlaces de importación, en los que un componente SCA envía peticiones y recibe respuestas de un servicio fuera del módulo, debe indicar el formato de los datos nativos. Dependiendo del formato, el sistema selecciona el manejador de datos o el enlace de datos correspondiente para transformar el objeto de negocio en datos nativos y viceversa.

IBM Business Process Manager proporciona un conjunto de formatos de datos predefinidos y los manejadores de datos o enlaces de datos correspondientes que dan soporte a los formatos. También puede crear sus propios manejadores de datos personalizados y registrar el formato de datos para los manejadores de datos. Para obtener más información, consulte el tema “Desarrollo de manejadores de datos” en el Centro de información de WebSphere Integration Developer.

- Los *manejadores de datos* tienen un protocolo neutro y transforman datos de un formato a otro. En IBM Business Process Manager, los manejadores de datos normalmente transforman datos nativos (por ejemplo, XML, CSV y COBOL) en un objeto de negocio, y un objeto de negocio en datos nativos. Como tiene un protocolo neutro, puede reutilizar el mismo manejador de datos con varios enlaces de exportación e importación. Por ejemplo, puede utilizar el mismo manejador de datos XML con un enlace de exportación o importación HTTP o con un enlace de exportación o importación JMS.
- Los *enlaces de datos* también transforman datos nativos en un objeto de negocio (y viceversa), pero son específicos del protocolo. Por ejemplo, un enlace de datos HTTP sólo puede utilizarse con un enlace de exportación o importación HTTP. A diferencia de los manejadores de datos, un enlace de datos HTTP no puede reutilizarse con un enlace de exportación o importación MQ.

**Nota:** Hay tres enlaces de datos HTTP en desuso (HTTPStreamDataBindingSOAP, HTTPStreamDataBindingXML y HTTPServiceGatewayDataBinding) a partir de IBM Business Process Manager Versión 7.0. Utilice los manejadores de datos siempre que sea posible.

Como se ha indicado anteriormente, puede crear manejadores de datos personalizados, si es necesario. También puede crear enlaces de datos personalizados; no obstante, resulta recomendable crear manejadores de datos personalizados porque se pueden utilizar entre varios enlaces.

### Manejadores de datos:

Los manejadores de datos se configuran respecto a los enlaces de exportación e importación para transformar datos de un formato a otro mediante un protocolo neutro. Se proporcionan varios manejadores de datos como parte del producto, pero también puede crear manejadores de datos propios, si es necesario. Puede asociar un manejador de datos con un enlace de exportación o importación en uno de dos niveles: puede asociarlo con todas las operaciones de la interfaz de la exportación o importación, o puede asociarlo con una operación específica para la petición o respuesta.

### Manejadores de datos predefinidos

Utilice IBM Integration Designer para especificar el manejador de datos que desee utilizar.

Los manejadores de datos predefinidos para que los utilice el usuario se enumeran en la tabla siguiente, que también describe cómo el manejador de datos transforma datos de entrada y salida.

**Nota:** Excepto allí donde se indique específicamente, dichos manejadores de datos se pueden utilizar con enlaces JMS, JMS genéricos, MQ JMS, WebSphere MQ y HTTP. Consulte el tema “Manejadores de datos”, en el Centro de información de Integration Designer, para obtener información más detallada.

Tabla 11. Manejadores de datos predefinidos

Manejador de datos	Datos nativos a objeto de negocio	Objeto de negocio a datos nativos
ATOM	Analiza canales de información ATOM en un objeto de negocio de canal de información ATOM.	Serializa un objeto de negocio de canal de información ATOM en canales de información ATOM.
Delimitado	Analiza los datos delimitados en un objeto de negocio.	Serializa un objeto de negocio en datos delimitados, incluido el formato CSV.
Anchura fija	Analiza datos de anchura fija en un objeto de negocio.	Serializa un objeto de negocio en datos de anchura fija.
Gestionado por WTX	Delega la transformación del formato de datos a WebSphere Transformation Extender (WTX). El nombre de correlación WTX se deriva del manejador de datos.	Delega la transformación del formato de datos a WebSphere Transformation Extender (WTX). El nombre de correlación WTX se deriva del manejador de datos.
Gestionado por WTX Invoker	Delega la transformación del formato de datos a WebSphere Transformation Extender (WTX). El nombre de correlación WTX lo proporciona el usuario.	Delega la transformación del formato de datos a WebSphere Transformation Extender (WTX). El nombre de correlación WTX lo proporciona el usuario.
JAXB	Serializa los beans Java en un objeto de negocio utilizando las reglas de correlación definidas por la especificación JAXB (Java Architecture for XML Binding).	Deserializa un objeto de negocio en beans Java utilizando las reglas de correlación definidas por la especificación JAXB.
JAXWS <b>Nota:</b> El manejador de datos JAXWS sólo puede utilizarse con el enlace EJB.	Lo utiliza un enlace EJB para transformar un objeto Java de respuesta o un objeto Java de excepción en un objeto de negocio de respuesta utilizando las reglas de correlación definidas por la especificación JAX-WS (Java API for XML Web Services).	Lo utiliza un enlace EJB para transformar un objeto de negocio en parámetros de método Java de salida utilizando las reglas de correlación definidas por la especificación JAX-WS.
JSON	Analiza datos JSON en un objeto de negocio.	Serializa un objeto de negocio en datos JSON.
Cuerpo nativo	Analiza los bytes nativos, el texto, correlación, corriente u objeto en uno de cinco objetos de negocio base (texto, bytes, correlación, corriente u objeto).	Transforma los cinco objetos de negocio base en byte, texto, correlación, corriente u objeto.
SOAP	Analiza el mensaje SOAP (y la cabecera) en un objeto de negocio.	Serializa un objeto de negocio en datos a un mensaje SOAP.
XML	Analiza datos XML en un objeto de negocio.	Serializa un objeto de negocio en datos XML.
UTF8XMLDataHandler	Analiza datos XML codificados UTF-8 en un objeto de negocio.	Serializa un objeto de negocio en datos XML codificados UTF-8 cuando se envía un mensaje.

### Creación de un manejador de datos

Puede encontrar información detallada sobre la creación de un manejador de datos en el tema “Desarrollo de manejadores de datos” en el Centro de información de Integration Designer.

## Enlaces de datos:

Los enlaces de datos se configuran respecto a los enlaces de exportación e importación para transformar datos de un formato a otro. Los enlaces de datos son específicos de un protocolo. Se proporcionan varios enlaces de datos como parte del producto, pero también puede crear enlaces de datos propios, si es necesario. Puede asociar un enlace de datos con un enlace de exportación o importación en uno de dos niveles; puede asociarlo con todas las operaciones de la interfaz de la exportación o importación, o puede asociarlo con una operación específica para la petición o respuesta.

WebSphere Integration Developer se utiliza para especificar el enlace de datos que desea utilizar o para crear un enlace de datos propio. Puede encontrar una descripción sobre la creación de enlaces de datos en la sección “Visión general de enlaces JMS, MQ JMS y enlaces genéricos” del Centro de información de WebSphere Integration Developer.

## Enlaces JMS

En la tabla siguiente se enumeran los enlaces de datos que se pueden utilizar con:

- Enlaces JMS
- Enlaces JMS genéricos
- Enlaces JMS de WebSphere MQ

La tabla también incluye una descripción de las tareas que los enlaces de datos realizan.

*Tabla 12. Enlaces de datos predefinidos para enlaces JMS*

Enlace de datos	Datos nativos a objeto de negocio	Objeto de negocio a datos nativos
Objeto Java serializado	Transforma el objeto Java serializado en un objeto de negocio (que se correlaciona como el tipo de entrada o salida en WSDL).	Serializa un objeto de negocio en el objeto serializado Java en el mensaje de objeto JMS.
Bytes incluidos	Extrae los bytes del mensaje de bytes de entrada JMS y los envuelve en el objeto de negocio JMSBytesBody.	Extrae los bytes del objeto de negocio JMSBytesBody y los envuelve en el mensaje de bytes JMS de salida.
Entrada de correlación envuelta	Extrae el nombre, el valor y la información de tipo de cada entrada del mensaje de correlación JMS de entrada y crea una lista de objetos de negocio MapEntry. A continuación, envuelve la lista en el objeto de negocio JMSTextBody.	Extrae el nombre, el valor y la información de tipo de la lista MapEntry en el objeto de negocio JMSMapBody y crea las entradas correspondientes en el mensaje de correlación JMS de salida.
Objeto envuelto	Extrae el objeto del mensaje de objeto JMS de entrada y lo envuelve en el objeto de negocio JMSObjectBody.	Extrae el objeto del objeto de negocio JMSObjectBody y lo envuelve en el mensaje de objeto JMS de salida.
Texto incluido	Extrae el texto del mensaje de texto JMS de entrada y lo envuelve en el objeto de negocio JMSTextBody.	Extrae el texto del objeto de negocio JMSTextBody y lo envuelve en el mensaje de texto JMS de salida.

## Enlaces de WebSphere MQ

En la tabla siguiente se enumeran los enlaces de datos que se pueden utilizar con WebSphere MQ y se describen las tareas que los enlaces de datos realizan.

Tabla 13. Enlaces de datos predefinidos para enlaces de WebSphere MQ

Enlace de datos	Datos nativos a objeto de negocio	Objeto de negocio a datos nativos
Objeto Java serializado	Transforma el objeto Java serializado procedente del mensaje entrante en un objeto de negocio (que se correlaciona como el tipo de entrada o salida en WSDL).	Transforma un objeto de negocio en el objeto serializado Java en el mensaje de salida.
Bytes incluidos	Extrae los bytes del mensaje de bytes MQ sin estructurar y los envuelve en el objeto de negocio JMSBytesBody.	Extrae los bytes de un objeto de negocio JMSBytesBody y recorta los bytes en el mensaje de bytes MQ sin estructurar de salida.
Texto incluido	Extrae el texto del mensaje de texto MQ sin estructurar y lo envuelve en el objeto de negocio JMSTextBody.	Extrae el texto del objeto de negocio JMSTextBody y lo envuelve en un mensaje de texto MQ sin estructurar.
Entrada de corriente envuelta	Extrae el nombre y la información de tipo de cada entrada del mensaje de corriente JMS de entrada y crea una lista de objetos de negocio StreamEntry. A continuación, envuelve la lista en el objeto de negocio JMSStreamBody.	Extrae el nombre y la información de tipo de la lista StreamEntry en el objeto de negocio JMSStreamBody y crea las entradas correspondientes en el mensaje JMSStreamMessage de salida.

Además de los enlaces de datos listados en Tabla 13, WebSphere MQ también utiliza enlaces de datos de cabecera. Consulte el Centro de información de WebSphere Integration Developer para obtener detalles.

## Enlaces HTTP

En la tabla siguiente se enumeran los enlaces de datos que se pueden utilizar con HTTP y se describen las tareas que los enlaces de datos realizan.

Tabla 14. Enlaces de datos predefinidos para enlaces HTTP

Enlace de datos	Datos nativos a objeto de negocio	Objeto de negocio a datos nativos
Bytes incluidos	Extrae los bytes del cuerpo del mensaje HTTP de entrada y los envuelve en el objeto de negocio HTTPBytes.	Extrae los bytes del objeto de negocio HTTPBytes y los añade al cuerpo del mensaje HTTP de salida.
Texto incluido	Extrae el texto del cuerpo del mensaje HTTP de entrada y lo envuelve en el objeto de negocio HTTPText.	Extrae el texto del objeto de negocio HTTPText y los añade al cuerpo del mensaje HTTP de salida.

## Selectores de función en enlaces de exportación

Un selector de función para indicar qué operación debe ejecutarse en los datos para un mensaje de petición. Los selectores de función se configuran como parte de un enlace de exportación.

Consideremos una exportación SCA que expone una interfaz. La interfaz contiene dos operaciones: Crear y Actualizar. La exportación tiene un enlace JMS que lee de una cola.

Cuando llega un mensaje a la cola, se pasa la exportación a los datos asociados, pero, ¿qué operación de la interfaz de la exportación debe invocarse en el componente conectado? La operación viene determinada por la configuración del selector de función y del enlace de exportación.

El selector de función devuelve el nombre de función nativa (el nombre de función en el sistema cliente que ha enviado el mensaje). A continuación, el nombre de función nativa se correlaciona con el nombre

de función u operación en la interfaz asociada a la exportación. Por ejemplo, en la siguiente figura, el selector de función devuelve el nombre de función nativa (CRT) del mensaje entrante, el nombre de función nativa se correlaciona con la operación Crear, y el objeto de negocio se envía al componente SCA con la operación Crear.

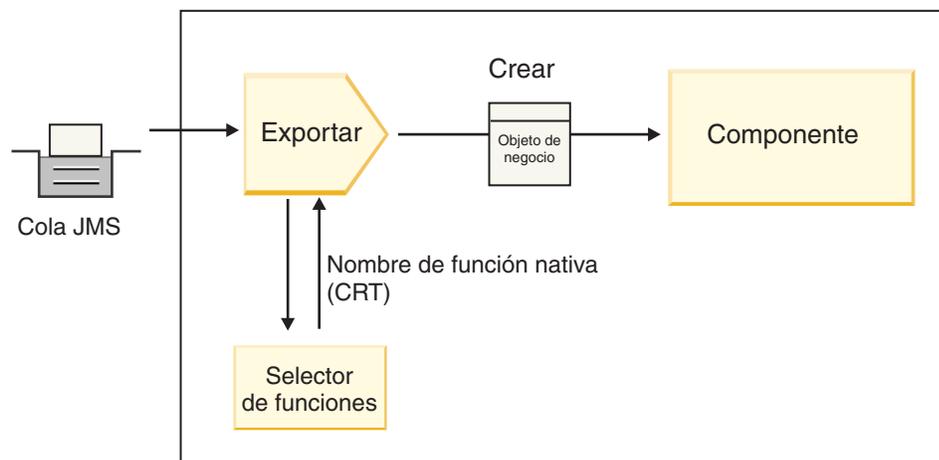


Figura 7. El selector de función

Si la interfaz sólo tiene una operación, no es necesario especificar ningún selector de función.

Hay disponibles varios selectores de función preempaquetados, que se enumeran en las secciones siguientes.

## Enlaces JMS

En la tabla siguiente se enumeran los selectores de función que se pueden utilizar con:

- Enlaces JMS
- Enlaces JMS genéricos
- Enlaces JMS de WebSphere MQ

Tabla 15. Selectores de función predefinidos para enlaces JMS

Selector de función	Descripción
Selector de función JMS para enlaces de datos JMS simples	Utiliza la propiedad JMSType del mensaje para seleccionar el nombre de la operación.
Selector de función de la propiedad de cabecera JMS	Devuelve el valor de String Property de JMS, TargetFunctionName, de la cabecera.
Selector de función de pasarela de servicio JMS	Determina si la petición es una operación unidireccional o bidireccional, examinando la propiedad JMSReplyTo, establecida por el cliente.

## Enlaces de WebSphere MQ

En la tabla siguiente se enumeran los selectores de función que se pueden utilizar con enlaces WebSphere MQ:

Tabla 16. Selectores de función predefinidos para enlaces de WebSphere MQ

Selector de función	Descripción
Selección de función handleMessage de MQ	Devuelve handleMessage como un valor que se correlaciona, utilizando los enlaces de método de exportación, al nombre de una operación de la interfaz.
MQ utiliza el selector de función por omisión de JMS.	Lee la operación nativa de la propiedad TargetFunctionName de la carpeta de una cabecera MQRFH2.
MQ utiliza el formato del cuerpo del mensaje como función nativa	Encuentra el campo Format de la última cabecera y devuelve dicho campo como un valor String.
Selección de función tipo de MQ	Crea un método en el enlace de exportación recuperando un URL que contiene las propiedades Msd, Set, Type y Format, encontradas en la cabecera MQRFH2.
Selector de función de pasarela de servicio MQ	Utiliza la propiedad MsgType de la cabecera MQMD para determinar el nombre de la operación.

## Enlaces HTTP

En la tabla siguiente se enumeran los selectores de función que se pueden utilizar con enlaces HTTP.

Tabla 17. Selectores de función predefinidos para enlaces HTTP

Selector de función	Descripción
Selector de función HTTP basado en la cabecera TargetFunctionName	Utiliza la propiedad de la cabecera HTTP TargetFunctionName del cliente para determinar qué operación debe invocarse en tiempo de ejecución desde la exportación.
Selector de función HTTP basado en el URL y el método HTTP.	Utiliza la vía de acceso relativa del URL añadido con el método HTTP del cliente para determinar la operación nativa definida en la exportación.
Selector de función de pasarela de servicio HTTP basado en URL con un nombre de operación.	Determina el método para invocar basado en el URL si se ha añadido "operationMode = oneWay" al URL de petición.

**Nota:** Puede también crear su propio selector de función, utilizando IBM Integration Designer. En el Information Center de IBM Integration Designer se proporciona información sobre cómo crear un selector de función. Por ejemplo, en el tema "Visión general de selectores de función MQ", se puede encontrar una descripción de un selector de función para los enlaces de WebSphere MQ.

## Manejo de errores

Puede configurar los enlaces de importación y exportación para que manejen los errores (por ejemplo, las excepciones empresariales) que se producen durante el proceso especificando manejadores de datos de error. Puede configurar un manejador de datos de error a tres niveles: puede asociar un manejador de datos de error con una anomalía, con una operación, o para todas las operaciones con un enlace.

Un manejador de datos de error procesa los datos de error y los transforma al formato correcto para que se puedan enviar en el enlace de exportación o importación.

- Para un enlace de exportación, el manejador de datos de error transforma el objeto de negocio de excepción enviado desde el componente en un mensaje de respuesta que pueda utilizarse en la aplicación cliente.
- Para un enlace de importación, el manejador de datos de error transforma los datos de error o el mensaje de respuesta enviado desde un servicio en un objeto de negocio de excepción que pueda utilizarse en el componente SCA.

Para los enlaces de importación, el enlace invoca el selector de error, que determina si el mensaje de respuesta es una respuesta normal, un error de empresa o una excepción de tiempo de ejecución.

Puede especificar un manejador de datos de error para un determinado error, para una operación y para todas las operaciones con un enlace.

- Si el manejador de datos de error se establece a los tres niveles, se invoca el manejador de datos asociado con un determinado error.
- Si los manejadores de datos de error se establecen en los niveles de operación y enlace, se invoca el manejador de datos asociado con la operación.

En IBM Integration Designer se utilizan dos editores para especificar el manejo de errores. El editor de interfaz se utiliza para indicar si habrá un error en una operación. Una vez generado un enlace con esta interfaz, el editor en la vista de propiedades permite configurar cómo se manejará el error. Para obtener más información, consulte el tema "Selectores de errores" en el Information Center de IBM Integration Designer.

### Cómo se manejan los errores en los enlaces de exportación:

Cuando se produce un error durante el proceso de la petición de una aplicación de cliente, el enlace de exportación puede devolver la información de error al cliente. Configure el enlace de exportación para especificar cómo se debe procesar y devolver el error al cliente.

Configure el enlace de exportación utilizando IBM Integration Designer.

Durante el proceso de la solicitud, un cliente invoca una exportación con una solicitud, y la exportación invoca el componente SCA. Durante el proceso de la solicitud, el componente SCA puede devolver una respuesta empresarial, o puede generar una excepción empresarial de servicio o una excepción de tiempo de ejecución de servicio. Cuando esto ocurre, el enlace de exportación transforma la excepción en un mensaje de error y lo envía al cliente, tal como se muestra en la siguiente figura y se describe en las siguientes secciones.



Figura 8. Cómo se envía la información de error desde el componente mediante el enlace de exportación al cliente

Puede crear un manejador de datos personalizado o un enlace de datos para manejar los errores.

### Errores de empresa

Los errores de empresa son anomalías o excepciones de empresa que se producen durante el proceso.

Considere la siguiente interfaz, que tiene una operación createCustomer. Esta operación tiene definidos dos errores de empresa: CustomerAlreadyExists y MissingCustomerId.

Operaciones y sus parámetros

	Nombre	Tipo
▼ crearCliente		
Entrada(s)	entrada	InfoCliente
Salida(s)	salida	InfoCliente
Error	El cliente ya existe	BO-Cliente ya existe
Error	FaltaIDCliente	BO-FaltaIDCliente

Figura 9. Interfaz con dos errores

En este ejemplo, si un sistema cliente envía una petición para crear un cliente (de este componente SCA) y el cliente ya existe, el componente genera un error `CustomerAlreadyExists` en la exportación. La exportación debe propagar este error de empresa de nuevo al cliente que realiza la llamada. Para ello, utiliza el manejador de datos de error que se ha configurado en el enlace de exportación.

Cuando el enlace de exportación recibe un error de empresa, se produce el siguiente proceso:

1. El enlace determina qué manejador de datos de error invocar para manejar el error. Si la excepción empresarial de servicio contiene el nombre del error, se invoca el manejador de datos configurado en el error. Si la excepción empresarial de servicio no contiene el nombre del error, el nombre del error se deriva haciendo coincidir los tipos de error.
2. El enlace invoca el manejador de datos de error con el objeto de datos de la excepción empresarial de servicio.
3. El manejador de datos de error transforma el objeto de datos de error en un mensaje de respuesta y lo devuelve al enlace de exportación.
4. La exportación devuelve el mensaje de respuesta al cliente.

Si la excepción empresarial de servicio contiene el nombre del error, se invoca el manejador de datos configurado en el error. Si la excepción empresarial de servicio no contiene el nombre del error, el nombre del error se deriva haciendo coincidir los tipos de error.

### Excepciones de tiempo de ejecución

Una excepción de tiempo de ejecución es una excepción que se produce en la aplicación SCA durante el proceso de una petición que no se corresponde con un error de empresa. A diferencia de los errores de empresa, las excepciones de tiempo de ejecución no se definen en la interfaz.

En determinados escenarios, si lo desea, puede propagar estas excepciones de tiempo de ejecución a la aplicación de cliente para que esta pueda realizar la acción correspondiente.

Por ejemplo, si un sistema cliente envía una petición (al componente SCA) para crear un cliente y se produce un error de autorización durante el proceso de la solicitud, el componente genera una excepción de tiempo de ejecución. Esta excepción de tiempo de ejecución debe volver a propagarse al cliente que realiza la llamada, para que pueda realizar la acción correspondiente para la autorización. Para ello, se utiliza el manejador de datos de excepción de tiempo de ejecución configurado en el enlace de exportación.

**Nota:** Puede configurar un manejador de datos de excepción de tiempo de ejecución sólo en enlaces HTTP.

El proceso de una excepción de tiempo de ejecución es parecido al proceso de un error de empresa. Si se ha configurado un manejador de datos de excepción de tiempo de ejecución, se produce el siguiente proceso:

1. El enlace de exportación invoca el manejador de datos adecuado con la excepción de tiempo de ejecución de servicio.
2. El manejador de datos transforma el objeto de datos de error en un mensaje de respuesta y lo devuelve al enlace de exportación.
3. La exportación devuelve el mensaje de respuesta al cliente.

El manejo de errores y el manejo de excepciones de tiempo de ejecución son opcionales. Si no desea propagar los errores o las excepciones de tiempo de ejecución al cliente que realiza la llamada, no configure el manejador de datos de error o el manejador de datos de excepción de tiempo de ejecución.

### Cómo se manejan los errores en los enlaces de importación:

Un componente utiliza una importación para enviar una petición a un servicio fuera del módulo. Cuando se produce un error durante el proceso de la solicitud, el servicio devuelve el error al enlace de importación. Puede configurar el enlace de importación para especificar cómo se debe procesar y devolver el error al componente.

Configure el enlace de importación utilizando IBM Integration Designer. Puede especificar un manejador de datos de error (o un enlace de datos), así como un selector de error.

### Manejadores de datos de error

El servicio que procesa la petición envía al enlace de importación la información de error, en forma de excepción o un mensaje de respuesta que contiene los datos de error.

El enlace de importación transforma la excepción de servicio o el mensaje de respuesta en una excepción empresarial de servicio o una excepción de tiempo de ejecución de servicio, tal como se muestra en la siguiente figura y se describe en las siguientes secciones.

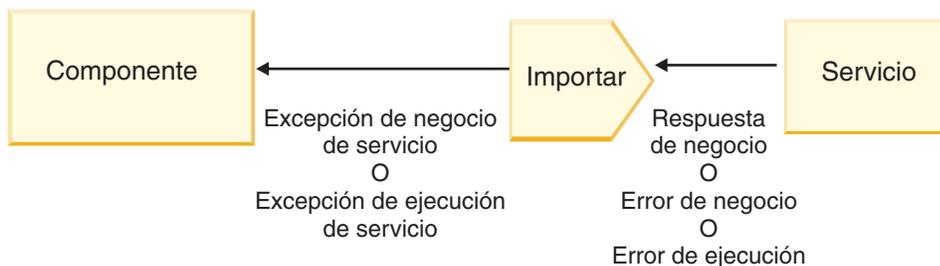


Figura 10. Cómo se envía la información de error desde el servicio mediante la importación al componente

Puede crear un manejador de datos personalizado o un enlace de datos para manejar los errores.

### Selectores de error

Cuando configura un enlace de importación, puede especificar un selector de error. El selector de error determina si la respuesta de importación es una respuesta real, una excepción empresarial o un error de tiempo de ejecución. También determina, a partir del cuerpo o la cabecera de la respuesta, el nombre de error nativo, que la configuración del enlace correlaciona al nombre de un error de la interfaz asociada.

Hay disponibles dos tipos de selectores de error preempaquetados que se pueden utilizar con las importaciones JMS, MQ JMS, JMS genéricas, WebSphere MQ y HTTP:

Tabla 18. Selectores de error preempaquetados

Tipo de selector de error	Descripción
Basado en cabecera	Determina si un mensaje de respuesta es un error de empresa, una excepción de tiempo de ejecución, o un mensaje normal basado en las cabeceras del mensaje de respuesta de entrada.
SOAP	Determina si el mensaje SOAP de respuesta es una respuesta normal, un error de empresa o una excepción de tiempo de ejecución.

A continuación se muestran ejemplos de selectores de error basados en cabecera y el selector de error SOAP.

- Selector de error basado en cabecera

Si una aplicación quiere indicar que el mensaje de entrada es un error de empresa, deben haber dos cabeceras en el mensaje de entrada para los errores de empresa, tal como se muestra a continuación:

```
Header name = FaultType, Header value = Business
Header name = FaultName, Header value = <nombre de error nativo definido por el usuario>
```

Si una aplicación quiere indicar que el mensaje de entrada es una excepción de tiempo de ejecución, entonces debe haber una cabecera en el mensaje de entrada, tal como se muestra a continuación:

```
Header name = FaultType, Header value = Runtime
```

- Selector de error SOAP

Una error de empresa debe enviarse como parte del mensaje SOAP con la cabecera SOAP personalizada siguiente. "CustomerAlreadyExists" es el nombre del error, en este caso.

```
<ibmSoap:BusinessFaultName
xmlns:ibmSoap="http://www.ibm.com/soap">CustomerAlreadyExists
</ibmSoap:BusinessFaultName>
```

El selector de error es opcional. Si no especifica ningún selector de error, el enlace de importación no puede determinar el tipo de respuesta. Por tanto, el enlace la trata como si fuera una respuesta de empresa y llama al manejador de datos de respuesta o al enlace de datos de respuesta.

Puede crear un selector de error personalizado. Los pasos para crear un selector de error personalizado se proporcionan en el tema "Desarrollo de un selector de error personalizado" del Information Center de IBM Integration Designer.

## Errores de empresa

Un error de empresa puede producirse cuando hay un error en el proceso de una solicitud. Por ejemplo, si envía una petición para crear un cliente y el cliente ya existe, el servicio envía una excepción empresarial al enlace de importación.

Cuando el enlace recibe una excepción empresarial, los pasos del proceso dependen de si se ha configurado un selector de error para el enlace.

- Si no se ha configurado un selector de error, el enlace invoca el manejador de datos de respuesta o el enlace de datos de respuesta.
- Si se ha configurado un selector de error, se produce el siguiente proceso:
  1. El enlace de importación invoca el selector de error para determinar si la respuesta es un error de empresa, una respuesta de empresa, o un error de tiempo de ejecución.
  2. Si la respuesta es un error de empresa, el enlace de importación invoca el selector de error para proporcionar el nombre de error nativo.
  3. El enlace de importación determina el error WSDL que corresponda al nombre de error nativo que haya devuelto el selector de error.

4. El enlace de importación determina el manejador de datos de error que se haya configurado para este error WSDL.
5. El enlace de importación invoca este manejador de datos con los datos de error.
6. El manejador de datos de error transforma los datos de error en un objeto de datos y lo devuelve al enlace de importación.
7. El enlace de importación construye un objeto de excepción empresarial de servicio con el objeto de datos y el nombre de error.
8. La importación devuelve el objeto de excepción empresarial de servicio al componente.

### Excepciones de tiempo de ejecución

Una excepción de tiempo de ejecución puede producirse cuando hay un problema de comunicación con el servicio. El proceso de una excepción de tiempo de ejecución es parecido al proceso de una excepción empresarial. Si se ha configurado un selector de error, se produce el siguiente proceso:

1. El enlace de importación invoca el manejador de datos de excepción de tiempo de ejecución con los datos de excepción.
2. El manejador de datos de excepción de tiempo de ejecución transforma los datos de excepción en un objeto de excepción de tiempo de ejecución de servicio y los devuelve al enlace de importación.
3. La importación devuelve el objeto de excepción tiempo de ejecución de servicio al componente.

## Interoperatividad entre módulos SCA y servicios Open SCA

El paquete de características de IBM WebSphere Application Server V7.0 para Service Component Architecture (SCA) proporciona un modelo de programación sencillo, pero potente, para construir aplicaciones basadas en las especificaciones de Open SCA. Los módulos SCA de IBM Business Process Manager utilizan los enlaces de importación y exportación para interoperar con servicios de Open SCA desarrollados en un entorno de Rational Application Developer y alojado por el paquete de características de WebSphere Application Server para Service Component Architecture.

Una aplicación SCA invoca una aplicación Open SCA a través de un enlace de importación. Una aplicación SCA recibe una llamada de una aplicación Open SCA a través de un enlace de exportación. Se muestra una lista de enlaces soportados en “Invocación de servicios en enlaces interoperables” en la página 53.

### Invocación de servicios Open SCA desde módulos SCA

Las aplicaciones SCA desarrolladas con IBM Integration Designer pueden invocar aplicaciones Open SCA desarrolladas en un entorno de Rational Application Developer. Esta sección proporciona un ejemplo de invocación de un servicio Open SCA desde un módulo SCA utilizando un enlace de importación SCA.

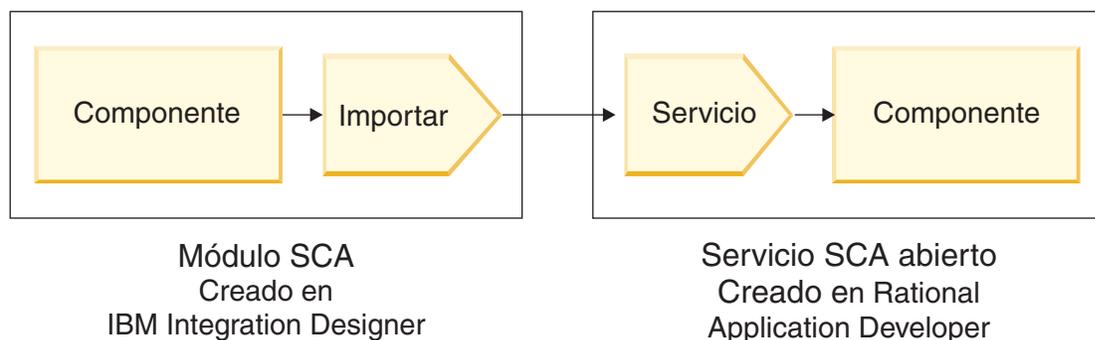


Figura 11. Componente de un módulo SCA que invoca un servicio Open SCA

No es necesaria ninguna configuración especial para invocar un servicio Open SCA.

Para conectarse a un servicio Open SCA a través de un enlace de importación SCA, proporcione el nombre de componente y el nombre del servicio Open SCA en el enlace de importación.

1. Para obtener el nombre del componente de destino y del servicio del compuesto Open SCA, realice los pasos siguientes:
  - a. Asegúrese de que la pestaña **Propiedades** esté abierta pulsando **Ventana > Mostrar vista > Propiedades**.
  - b. Abra el editor compuesto efectuando una doble pulsación en el diagrama compuesto que contiene el componente y el servicio. Por ejemplo, para un componente denominado **customer**, el diagrama compuesto es **customer.composite\_diagram**.
  - c. Pulse el componente de destino.
  - d. En el campo **Nombre** de la pestaña **Propiedades**, escriba el nombre del componente de destino.
  - e. Pulse el icono de servicio asociado con el componente.
  - f. En el campo **Nombre** de la pestaña **Propiedades**, escriba el nombre del servicio.
2. Para configurar la importación de IBM Business Process Manager para conectarlo al servicio Open SCA, realice los pasos siguientes:
  - a. En IBM Integration Designer, vaya a la pestaña **Propiedades** de la importación SCA que desee conectar al servicio Open SCA.
  - b. En el campo **Nombre de módulo**, entre el nombre de componente del paso 1d.
  - c. En el campo **Nombre de la exportación**, entre el nombre de servicio del paso 1f.
  - d. Guarde el trabajo pulsando Control+S.

## Invocación de módulos SCA desde servicios Open SCA

Las aplicaciones Open SCA desarrolladas en un entorno de Rational Application Developer pueden invocar aplicaciones SCA desarrolladas con IBM Integration Designer. Esta sección proporciona un ejemplo de invocación de un módulo SCA (mediante un enlace de exportación SCA) de un servicio Open SCA.

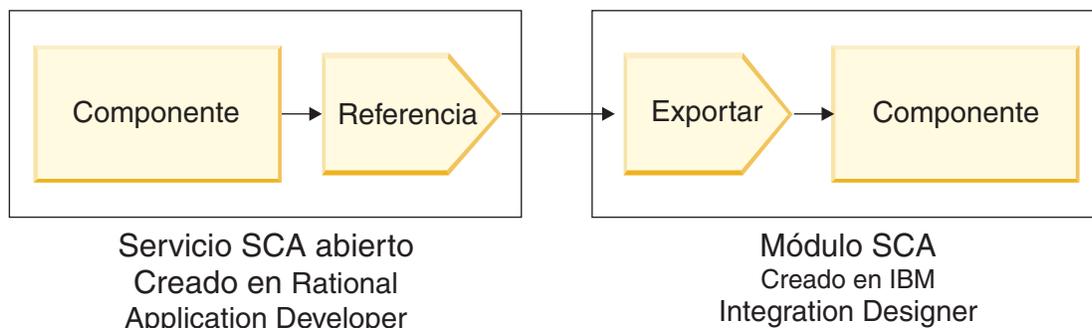


Figura 12. Servicio Open SCA que invoca el componente de un módulo SCA

Para conectarse a un componente SCA por medio de un enlace de referencia Open SCA, proporcione el nombre de módulo y el nombre de exportación.

1. Para obtener el nombre del módulo de destino y de la exportación, realice los pasos siguientes:
  - a. En IBM Integration Designer, abra el módulo en el editor de ensamblaje efectuando una doble pulsación en el módulo.
  - b. Pulse la exportación.
  - c. En el campo **Nombre** de la pestaña **Propiedades**, escriba el nombre de la exportación.

2. Configure la referencia de Open SCA que desea conectar al módulo de IBM Business Process Manager y la exportación:
  - a. En Rational Application Developer, abra el editor compuesto efectuando una doble pulsación en el diagrama compuesto que contiene el componente y el servicio.
  - b. Pulse el icono de referencia de la referencia de componente para visualizar las propiedades de referencia en la pestaña **Propiedades**.
  - c. Pulse la pestaña **Enlace** situada en el lado izquierdo de la página.
  - d. Pulse **Enlaces** y, a continuación, pulse **Añadir**.
  - e. Seleccione el enlace **SCA**.
  - f. En el campo **Uri**, entre el nombre de módulo de IBM Business Process Manager, seguido de una barra inclinada ("/"), seguida del nombre de exportación (que ha determinado en el paso 1c en la página 52).
  - g. Pulse **Aceptar**.
  - h. Guarde el trabajo pulsando Control+S.

## Invocación de servicios en enlaces interoperables

Los siguientes enlaces están soportados para la interoperatividad con un servicio Open SCA.

- Enlace SCA

En IBM Business Process Manager, cuando un módulo SCA invoca un servicio Open SCA mediante un enlace de importación SCA, están permitidos los estilos de invocación siguientes:

- Asíncrono (unidireccional)
- Síncrono (petición/respuesta)

La interfaz de importación SCA y la interfaz de servicio Open SCA deben utilizar una interfaz WSDL compatible con la interoperatividad de servicios web (WS-I).

Tenga en cuenta que el enlace SCA soporta la transacción y la propagación del contexto de seguridad.

- El enlace de servicio web (JAX-WS) con el protocolo SOAP1.1/HTTP o SOAP1.2/HTTP

La interfaz de importación SCA y la interfaz de servicio Open SCA deben utilizar una interfaz WSDL compatible con la interoperatividad de servicios web (WS-I).

Además, están soportadas las siguientes calidades de servicio:

- Transacción atómica de servicios web
- Seguridad de servicios web

- Enlace EJB

Se utiliza una interfaz Java para definir la interacción entre un módulo SCA y un servicio Open SCA, cuando se utiliza el enlace EJB.

Tenga en cuenta que el enlace EJB soporta la transacción y la propagación del contexto de seguridad.

- Enlaces JMS

La interfaz de importación SCA y la interfaz de servicio Open SCA deben utilizar una interfaz WSDL compatible con la interoperatividad de servicios web (WS-I).

Están soportados los siguientes proveedores JMS:

- WebSphere Platform Messaging (Enlace JMS)
- WebSphere MQ (Enlace JMS de MQ)

**Nota:** Los gráficos de negocio no funcionan entre sí en los enlaces SCA y, por lo tanto, no están soportados en las interfaces utilizadas para interoperar con el paquete de características de WebSphere Application Server para Service Component Architecture.

## Tipos de enlace

Utilice los *enlaces* específicos del protocolo con las importaciones y exportaciones para especifica los medios de transporte de datos dentro o fuera de un módulo.

### Selección de los enlaces adecuados

Existen varios enlaces que están disponibles para adaptarse a las necesidades de la aplicación.

Los enlaces disponibles en WebSphere Integration Developer proporcionan una gama de opciones. Esta lista le ayuda a identificar cuándo un tipo de enlace podría ser más idóneo para las necesidades de la aplicación.

Considere la posibilidad de utilizar un enlace *SCA* cuando sean aplicables estos factores:

- Todos los servicios están contenidos en módulos WebSphere Integration Developer; es decir, no hay servicios externos.
- Desea separar la función en diferentes módulos SCA que interactúan directamente entre ellos.
- Los módulos están fuertemente acoplados.

Considere la posibilidad de utilizar un enlace de *servicio Web* cuando sean aplicables estos factores:

- Necesita acceder a un servicio externo a través de Internet o proporcionar un servicio a través de Internet.
- Los servicios están unidos ligeramente.
- Es preferible la comunicación síncrona; es decir, una solicitud de un servicio puede esperar una respuesta de otro.
- El protocolo del servicio externo al que está accediendo o el servicio que desea proporcionar es SOAP/HTTP o SOAP/JMS.

Considere la posibilidad de utilizar un enlace *HTTP* cuando sean aplicables estos factores:

- Necesita acceder a un servicio externo en Internet o proporcionar un servicio en Internet y está trabajando con otros servicios Web basados en el modelo HTTP (es decir, está utilizando operaciones de interfaz HTTP conocidas públicamente, por ejemplo GET, PUT, DELETE, etc.).
- Los servicios están unidos ligeramente.
- Es preferible la comunicación síncrona; es decir, una solicitud de un servicio puede esperar una respuesta de otro.

Considere la posibilidad de utilizar un enlace *EJB* cuando sean aplicables estos factores:

- El enlace es para un servicio importado que es a su vez un EJB o al que es necesario que accedan los clientes EJB.
- El servicio importado está débilmente acoplado.
- No son necesarias interacciones EJB con estado.
- Es preferible la comunicación síncrona; es decir, una solicitud de un servicio puede esperar la respuesta de otro servicio.

Considere la posibilidad de utilizar un enlace *EIS* cuando sean aplicables estos factores:

- Necesita acceder a un servicio en un sistema EIS utilizando un adaptador de recursos.
- Se prefiere la transmisión de datos síncrona respecto a la asíncrona.

Considere la posibilidad de utilizar un enlace *JMS* cuando sean aplicables estos factores:

**Nota:** Existen varios tipos de enlaces JMS. Si piensa intercambiar mensajes SOAP utilizando JMS, tenga en cuenta la posibilidad de utilizar el enlace de servicio Web con el protocolo SOAP/JMS. Consulte “Enlaces de servicio web” en la página 55.

- Necesita acceder a un sistema de mensajería.
- Los servicios están unidos ligeramente.
- Se prefiere la transmisión de datos asíncrona respecto a la síncrona.

Considere un enlace *JMS genérico* cuando sean aplicables estos factores:

- Necesita acceder a un sistema de mensajería de proveedor que no es de IBM.
- Los servicios están unidos ligeramente.
- La fiabilidad es más importante que el rendimiento; es decir, la transmisión de datos asíncrona es preferible a la síncrona.

Considere la posibilidad de utilizar un enlace *MQ* cuando sean aplicables estos factores:

- Necesita acceder a un sistema de mensajería WebSphere MQ y necesita utilizar las funciones nativas de MQ.
- Los servicios están unidos ligeramente.
- La fiabilidad es más importante que el rendimiento; es decir, la transmisión de datos asíncrona es preferible a la síncrona.

Considere la posibilidad de utilizar un enlace *JMS MQ* cuando sean aplicables estos factores:

- Necesita acceder a un sistema de mensajería WebSphere MQ pero puede hacerlo dentro de un contexto JMS; es decir, el subconjunto JMS de funciones es suficiente para la aplicación.
- Los servicios están unidos ligeramente.
- La fiabilidad es más importante que el rendimiento; es decir, la transmisión de datos asíncrona es preferible a la síncrona.

## Enlaces SCA

Un enlace SCA (Service Component Architecture) permite a un servicio comunicarse con otros servicios de otros módulos. Una importación con un enlace SCA le permite acceder a un servicio de otro módulo SCA. Una exportación con un enlace SCA le permite ofrecer un servicio a otros módulos.

Utilice WebSphere Integration Developer para generar y configurar enlaces SCA en importaciones y exportaciones en módulos SCA.

Si los módulos se ejecutan en el mismo servidor o se despliegan en el mismo clúster, un enlace SCA es el enlace más fácil y rápido para utilizar.

Después de que se despliegue en el servidor el módulo que contiene el enlace SCA, puede utilizar la consola administrativa para ver la información sobre el enlace o, en el caso de un enlace de importación, para cambiar las propiedades seleccionadas del enlace.

## Enlaces de servicio web

Un enlace de servicio web es la forma de transmitir mensajes desde un componente SCA (Service Component Architecture) a un servicio web (y viceversa).

### Visión general de enlaces de servicios web:

Un enlace de importación de servicios Web permite llamar a un servicio Web externo desde el componente SCA (Service Component Architecture). Un enlace de exportación de servicios Web permite exponer los componentes SCA a clientes como servicios Web.

Con un enlace de servicio web, acceda a los servicios externos utilizando los mensajes SOAP interoperables y las calidades de servicio (QoS).

Utilice Integration Designer para generar y configurar enlaces de servicio Web en importaciones y exportaciones en módulos SCA. Los siguientes tipos de enlaces de servicio web están disponibles:

- SOAP1.2/HTTP y SOAP1.1/HTTP

Estos enlaces se basan en la API de Java para los servicios web XML (JAX-WS), una API de programación Java para crear servicios web.

- Utilice SOAP1.2/HTTP si el servicio web cumple la especificación SOAP 1.2.
- Utilice SOAP1.1/HTTP si el servicio web es compatible con la especificación SOAP 1.1.

**Importante:** Al desplegar una aplicación con un enlace de servicio web (JAX-WS), el servidor de destino no debe tener seleccionada la opción **Iniciar componentes como sea necesario**. Consulte “Verificación de la configuración del servidor” en la página 64 para obtener más información.

Cuando seleccione uno de estos enlaces, puede enviar los adjuntos con los mensajes SOAP.

Los enlaces de servicio web trabajan con los mensajes SOAP estándar. Sin embargo a través del uso de uno de los enlaces JAX-WS de servicio web, puede personalizar la forma en la que se analizan o escriben los mensajes SOAP. Por ejemplo, puede manejar elementos no estándar en mensajes SOAP o aplicar proceso adicional en el mensaje SOAP. Cuando configure el enlace, especifique un manejador de datos personalizados que realice este proceso en el mensaje SOAP.

Puede utilizar conjuntos de políticas con un enlace de servicio web (JAX-WS). Un conjunto de políticas es una colección de tipos de política, cada una de las cuales proporciona una calidad de servicio (QoS). Por ejemplo, el conjunto de políticas WSAddressing proporciona una manera de dirigir servicios web y mensajes de manera uniforme, a la vez que es neutral en cuanto al transporte. Debe utilizar Integration Designer para seleccionar el conjunto de políticas para el enlace.

**Nota:** Si desea utilizar un conjunto de políticas Security Assertion Markup Language (SAML), debe realizar tareas de configuración adicionales, tal como se describe en “Importación de conjuntos de políticas SAML” en la página 62.

- SOAP1.1/HTTP

Utilice este enlace si desea crear servicios web que utilicen un mensaje codificado con SOAP basado en el RPC basado en la API de Java para XML (JAX-RPC).

- SOAP1.1/JMS

Utilice este enlace para enviar o recibir mensajes SOAP utilizando un destino JMS (Java Message Service).

Independientemente del transporte (HTTP o JMS) que se utiliza para transportar el mensaje SOAP, los enlaces de servicio web siempre manejan las interacciones de petición/respuesta de forma síncrona. La hebra que realiza la invocación en el proveedor de servicios se bloquea hasta que se recibe una respuesta del proveedor. Consulte la “invocación síncrona” si desea más información sobre este estilo de invocación.

**Importante:** Las combinaciones siguientes de los enlaces de servicio Web no se pueden utilizar en exportaciones en el mismo módulo. Si necesita exponer componentes utilizando más de uno de estos enlaces de exportación, necesita tener cada uno en un módulo independiente y, a continuación, conectarse esos módulos a los componentes utilizando el enlace SCA:

- SOAP 1.1/JMS y SOAP 1.1/HTTP utilizando JAX-RPC
- SOAP 1.1/HTTP utilizando JAX-RPC y SOAP 1.1/HTTP utilizando JAX-WS
- SOAP 1.1/HTTP utilizando JAX-RPC y SOAP 1.2/HTTP utilizando JAX-WS

Tras desplegarse en el servidor el módulo SCA que contiene el enlace Web, podrá utilizar la consola administrativa para ver información sobre el enlace, o para cambiar propiedades seleccionadas de este último.

**Nota:** Los servicios web permiten a las aplicaciones interoperar utilizando descripciones estándar de servicios y formatos estándar para los mensajes que intercambian. Por ejemplo, los enlaces de importación y exportación de servicio web pueden interoperar con servicios que se han implementado mediante Web Services Enhancements (WSE) Versión 3.5 y Windows Communication Foundation (WCF) Versión 3.5 for Microsoft .NET. Cuando interopere con este tipo de servicios, debe asegurarse de lo siguiente:

- El archivo WSDL (lenguaje de descripción de servicios web) que se utiliza para acceder a una exportación de servicio web incluye un valor de acción SOAP no vacío para cada operación de la interfaz.
- El cliente de servicio Web establece la cabecera SOAPAction o la cabecera wsa:Action al enviar mensajes a una exportación de servicio Web.

### **Propagación de cabecera SOAP:**

Al manejar mensajes SOAP, podría necesitar acceder a información de determinadas cabeceras SOAP en los mensajes recibidos, asegurar que los mensajes con cabeceras SOAP se envían con valores específicos, o permitir que las cabeceras SOAP pasen un módulo.

Al configurar un enlace de servicio web en Integration Designer, puede indicar que desea que se propaguen las cabeceras SOAP.

- Cuando se reciben peticiones en una exportación o se reciben respuestas en una importación, puede accederse a la información de la cabecera SOAP, permitiendo así que la lógica del módulo se base en valores de cabecera y permitiendo que se modifiquen esas cabeceras.
- Cuando se envían peticiones desde una exportación o se envían respuestas desde una importación, las cabeceras SOAP pueden incluirse en esos mensajes.

El formato y la presencia de las cabeceras SOAP propagadas podrían resultar afectados por conjuntos de políticas configurados en la importación o la exportación, tal como se describe en Tabla 19 en la página 58.

Para configurar la propagación de cabeceras SOAP para una importación o exportación, seleccione (en la vista Propiedades de Integration Designer) el separador **Propagar cabecera de protocolo** y seleccione las opciones necesarias.

### **Cabecera WS-Addressing**

La cabecera WS-Addressing se puede propagar mediante el enlace de servicio web (JAX-WS).

Al propagar la cabecera WS-Addressing, tenga en cuenta la siguiente información:

- Si habilita la propagación para la cabecera WS-Addressing, la cabecera se propagará al módulo en las siguientes circunstancias:
  - Cuando las peticiones se reciben en una exportación
  - Cuando las respuestas se reciben en una importación
- La cabecera WS-Addressing no se propaga a mensajes de salida de IBM Business Process Manager (es decir, la cabecera no se propaga cuando se envían solicitudes desde una importación o cuando se envían respuestas desde la exportación).

### **Cabecera WS-Security**

La cabecera WS-Security puede propagarse mediante el enlace de servicio web (JAX-WS) y mediante el enlace de servicio web (JAX-RPC).

La especificación WS-Security de los servicios web describe mejoras en la mensajería SOAP para proporcionar calidad de protección mediante la integridad de mensajes, la confidencialidad de mensajes y

la autenticación de mensajes individuales. Estos mecanismos pueden utilizarse para acomodar una amplia gama de modelos de seguridad y tecnologías de cifrado.

Al propagar la cabecera WS-Security, tenga en cuenta la siguiente información:

- Si habilita la propagación para la cabecera WS-Security, la cabecera se propagará por el módulo en las siguientes circunstancias:
  - Cuando las peticiones se reciben en una exportación
  - Cuando las peticiones se envían desde una importación
  - Cuando las respuestas se reciben en una importación
- La cabecera *no* se propagará, de forma predeterminada, cuando se envíen respuestas desde la exportación. No obstante, si establece la propiedad JVM **WSSECURITY.ECHO.ENABLED** en **true**, la cabecera se propagará cuando se envíen respuestas desde la exportación. En este caso, si no se modifica la cabecera WS-Security en la vía de acceso de solicitud, las cabeceras WS-Security podrían pasar automáticamente de solicitudes a respuestas.
- El formato exacto del mensaje SOAP enviado desde una importación para una solicitud o desde una exportación para una respuesta podría no coincidir con el mensaje SOAP que se recibió originalmente. Por este motivo, deberá suponerse que cualquier firma digital no será válida. Si es necesaria una firma digital en los mensajes que se envían, debe establecerse utilizando el conjunto de políticas de seguridad adecuado, y las cabeceras WS-Security relacionadas con la firma digital en los mensajes recibidos deberán eliminarse en el módulo.

Para propagar la cabecera WS-Security, debe incluir el esquema WS-Security con el módulo de aplicación. Consulte “Cómo incluir el esquema WS-Security en un módulo de aplicación” en la página 59 para el procedimiento para incluir el esquema.

### Cómo se propagan las cabeceras

La manera en que se propagan las cabeceras depende del valor de política de seguridad en el enlace de importación o exportación, como se muestra en la Tabla 19:

Tabla 19. *Cómo se pasan las cabeceras de seguridad*

	Enlace de exportación sin política de seguridad	Enlace de exportación con política de seguridad
<b>Enlace de importación sin política de seguridad</b>	<p>Las cabeceras de seguridad se pasan tal cual a través del módulo. No se descifran.</p> <p>Las cabeceras se envían de salida con el mismo formato en que se recibieron.</p> <p>La firma digital podría ser no válida.</p>	<p>Las cabeceras de seguridad se descifran y se pasan a través del módulo con verificación y autenticación de firma.</p> <p>Las cabeceras descifradas se envían en la salida.</p> <p>La firma digital podría ser no válida.</p>
<b>Enlace de importación con política de seguridad</b>	<p>Las cabeceras de seguridad se pasan tal cual a través del módulo. No se descifran.</p> <p>Las cabeceras no se deben propagar en la importación. De lo contrario, se produce un error debido a la duplicación.</p>	<p>Las cabeceras de seguridad se descifran y se pasan a través del módulo con verificación y autenticación de firma.</p> <p>Las cabeceras no se deben propagar en la importación. De lo contrario, se produce un error debido a la duplicación.</p>

Configure los conjuntos de políticas adecuados en los enlaces de importación y exportación, ya que esto aísla el solicitante de servicio de los cambios realizados en la configuración o las peticiones QoS del solicitante de servicio. Se pueden utilizar cabeceras SOAP estándar visibles en un módulo para influir en

el proceso del módulo (por ejemplo, registro cronológico y rastreo). Propagar cabeceras SOAP a través de un módulo desde un mensaje recibido a un mensaje enviado significa que las ventajas de aislamiento del módulo quedan reducidas.

Las cabeceras estándar, como las cabeceras WS-Security, no deberán propagarse en una solicitud a una importación o una respuesta a una exportación cuando la importación o exportación tenga un conjunto de políticas asociado que normalmente daría como resultado la generación de esas cabeceras. De lo contrario, se producirá un error debido a la duplicación de las cabeceras. En vez de eso, las cabeceras deberán eliminarse de forma explícita o deberá configurarse el enlace de importación o exportación para impedir la propagación de cabeceras de protocolo.

### Acceso a cabeceras SOAP

Cuando se recibe un mensaje que contiene cabeceras SOAP desde una importación o exportación de servicio web, las cabeceras se colocan en la sección de cabeceras del objeto de mensaje de servicio (SMO). Puede acceder a la información de cabecera, tal como se describe en “Acceso a la información de cabecera de SOAP del SMO”.

### Cómo incluir el esquema WS-Security en un módulo de aplicación

El procedimiento siguiente expone los pasos para incluir el esquema en el módulo de aplicación:

- Si el sistema en el que se ejecuta Integration Designer tiene acceso a internet, realice los pasos siguientes:
  1. En la perspectiva Integración de negocio, seleccione **Dependencias** para el proyecto.
  2. Expanda **Recursos predefinidos** y seleccione **Archivos de esquema WS-Security 1.0** o **Archivos de esquema WS-Security 1.1** para importar el esquema al módulo.
  3. Limpie y vuelva a crear el proyecto.
- Si un sistema en el que se ejecuta Integration Designer no tiene acceso a internet, puede descargar el esquema a un segundo sistema que tenga acceso a internet. A continuación, puede copiarlo en el sistema en el que se ejecuta Integration Designer.
  1. Desde el sistema que tiene acceso a internet, descargue el esquema remoto:
    - a. Pulse **Archivo > Importar > Integración empresarial > WSDL y XSD**.
    - b. Seleccione **WSDL remoto** o **Archivo XSD**.
    - c. Importe los siguientes esquemas:
      - `http://www.w3.org/2003/05/soap-envelope/`
      - `http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-schema.xsd`
      - `http://www.w3.org/TR/xmlldsig-core/xmlldsig-core-schema.xsd`
  2. Copie los esquemas al sistema que no tiene acceso a internet.
  3. Desde el sistema que tiene acceso a internet, importe el esquema:
    - a. Pulse **Archivo > Importar > Integración empresarial > WSDL y XSD**.
    - b. Seleccione **WSDL local** o **Archivo XSD**.
  4. Cambie las ubicaciones del esquema para `oasis-wss-wssecurity_secext-1.1.xsd`:
    - a. Abra el esquema en `ubicación_lugar_trabajo/nombre_módulo/StandardImportFilesGen/oasis-wss-wssecurity_secext-1.1.xsd`.
    - b. Cambie:

```
<xs:import namespace='http://www.w3.org/2003/05/soap-envelope'
schemaLocation='http://www.w3.org/2003/05/soap-envelope'/>
a:
<xs:import namespace='http://www.w3.org/2003/05/soap-envelope'
schemaLocation='../w3/_2003/_05/soap_envelope.xsd'/>
```
    - c. Cambie:

```
<xs:import namespace='http://www.w3.org/2001/04/xmlenc#'
schemaLocation='http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-schema.xsd' />
```

a:

```
<xs:import namespace='http://www.w3.org/2001/04/xmlenc#'
schemaLocation='../w3/tr/_2002/rec_xmlenc_core_20021210/xenc-schema.xsd' />
```

5. Cambie la ubicación del esquema para oasis-200401-wss-wssecurity-secext-1.0.xsd:

a. Abra el esquema en *ubicación\_lugar\_trabajo/nombre\_módulo/StandardImportFilesGen/oasis-200401-wss-wssecurity-secext-1.0.xsd*.

b. Cambie:

```
<xsd:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="http://www.w3.org/TR/xmldsig-core/xmldsig-core-schema.xsd" />
```

a:

```
<xsd:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="../../w3/tr/_2002/rec_xmldsig_core_20020212/xmldsig-core-schema.xsd" />
```

6. Limpie y vuelva a crear el proyecto.

### Propagación de cabecera de transporte:

Cuando se manejan mensajes SOAP, es posible que tenga que acceder a información de determinadas cabeceras de transporte de mensajes recibidos; asegúrese de que los mensajes con cabeceras de transporte se han enviado con valores específicos, o que permiten a las cabeceras de transporte pasar entre un módulo.

Al configurar un enlace de servicio web en Integration Designer, puede indicar que desea que las cabeceras de transporte se propaguen.

- Cuando las peticiones se reciban en una exportación o las respuestas se reciban en una importación, se puede acceder a la información de cabecera de transporte, lo que permite que la lógica del módulo se base en los valores de cabecera y que permite que estas cabeceras se modifiquen.
- cuando se envían respuestas desde una exportación o se envían solicitudes desde una importación, las cabeceras de transporte pueden incluirse en dichos mensajes.

### Especificación de la propagación de cabeceras

Para configurar la propagación de cabeceras de transporte para una importación o exportación, siga estos pasos:

1. En la vista Propiedades de Integration Designer, seleccione **Enlace > Propagación**.
2. Defina la opción de propagación de tareas de transporte que le interese.

**Nota:** La propagación de tareas de transporte está inhabilitada de forma predeterminada y sólo puede desplegarse a un entorno de tiempo de ejecución versión 7.0.0.3 (o posterior). Asimismo, debe tener en cuenta que, para la versión 7.0.0.3, la propagación de cabeceras de transporte está limitada únicamente a cabeceras de transporte HTTP.

Si habilita la propagación de cabeceras de transporte, las cabeceras se propagarán por un módulo desde los mensajes recibidos y, si no elimina explícitamente las cabeceras, éstas se utilizarán en invocaciones subsiguientes de la misma hebra.

**Nota:** Las cabeceras de transporte no pueden propagarse si se utiliza el enlace de servicios web (JAX-RPC).

### Acceso a la información de cabecera

Cuando la propagación de cabecera de transporte está habilitada para los mensajes recibidos, todas las cabeceras de transporte (incluidas las cabeceras definidas por el cliente) son visibles en el objeto de

mensajes de servicio (SMO). Puede establecer las cabeceras en distintos valores o crear otras nuevas. No obstante, debe tener en cuenta que no se realiza ningún tipo de verificación ni validación en los valores establecidos, y que el uso de cabeceras inadecuadas o incorrectas puede dar lugar a problemas de ejecución del servicio web.

Considere la siguiente información acerca de la definición de cabeceras HTTP:

- Los cambios realizados en las cabeceras que se reservan para el motor de servicios web no se ajustarán en el mensaje de salida. Por ejemplo, las cabeceras de versión o método HTTP, Content-Type, Content-Length y SOAPAction están reservadas para el motor de servicios web.
- Si el valor de la cabecera es un número, el número (y no la serie) debe definirse directamente. Por ejemplo, utilice **Max-Forwards = 5** (en lugar de **Max-Forwards = Max-Forwards: 5**) y **Age = 300** (en lugar de **Age = Age: 300**).
- Si el mensaje de solicitud tiene un tamaño inferior a 32 KB, el motor de servicios web elimina la cabecera Transfer-Encoding y establece en su lugar la cabecera Content-Length en el tamaño fijado del mensaje.
- La cabecera Content-language se restablece mediante by WAS.channel.http en la vía de acceso de respuesta.
- Un valor no válido para Upgrade da lugar a un error tipo 500.
- Las siguientes cabeceras añaden el valor reservado por el motor de servicios web a los valores de cliente:
  - User-Agent
  - Cache-Control
  - Pragma
  - Accept
  - Conexión

Puede acceder a la información de cabecera mediante uno de estos métodos:

- Utilizando una primitiva de mediación para acceder a las estructuras de SMO  
Consulte los enlaces “Información relacionada” para encontrar información sobre el uso de primitivas de mediación.
- Utilizando la SPI de servicio de contexto

El siguiente código de ejemplo lee las cabeceras de transporte HTTP del servicio de contexto:

```
HeadersType headerType = ContextService.INSTANCE.getHeaders();
HTTPHeaderType httpHeaderType = headerType.getHTTPHeader();
List HTTPHeader httpHeaders = httpHeaderType.getHeader();
if(httpHeaders!=null){
    for(HTTPHeader httpHeader: httpHeaders){
        String httpHeadername = httpHeader.getName();
        String httpHeaderValue = httpHeader.getValue();
    }
}
List PropertyType properties = headerType.getProperties();
if(properties!=null){
    for(PropertyType property: properties){
        String propertyName = property.getName();
        String propertyValue = property.getValue().toString();
    }
}
```

## Resolución de problemas

Si detecta problemas al enviar las cabeceras revisadas, puede interceptar el mensaje TCP/IP utilizando herramientas tales como TCP/IP Monitor en Integration Designer. Para acceder a TCP/IP Monitor, seleccione **Ejecutar/Depurar > TCP/IP Monitor** en la página Preferencias.

También puede ver los valores de cabecera utilizando el rastreo de motor JAX-WS; **org.apache.axis2.\*=all: com.ibm.ws.websvcs.\*=all:**

### Trabajo con enlaces de servicio web (JAX-WS):

Cuando utilice enlaces de servicio web (JAX-WS) con las aplicaciones, puede añadir una calidad de servicio (QOS) SAML (Security Assertion Markup Language) al enlace. Primero debe utilizar la consola administrativa para importar el conjunto de políticas. También puede utilizar la consola administrativa para asegurarse de que el servidor esté bien configurado para utilizarlo con el enlace de servicio web (JAX-WS).

#### *Importación de conjuntos de políticas SAML:*

SAML (Security Assertion Markup Language) es un estándar OASIS basado en XML para intercambiar información sobre los atributos de identidad de usuario y seguridad. Cuando configure un enlace de servicio web (JAX-WS) en Integration Designer, puede especificar un conjunto de políticas SAML. Primero debe utilizar la consola administrativa de IBM Business Process Manager para establecer los conjuntos de políticas SAML como disponibles de modo que puedan importarse Integration Designer.

Los conjuntos de políticas SAML se encuentran normalmente en el directorio de configuración de perfiles:

*raíz\_perfil/config/templates/PolicySets*

Antes de empezar este procedimiento, verifique que los siguientes directorios (que contienen los conjuntos de políticas) se encuentren en el directorio de configuración de perfiles:

- SAML11 Valor predeterminado de Bearer WSHTTPS
- SAML20 Valor predeterminado de Bearer WSHTTPS
- SAML11 Valor predeterminado de Bearer WSSecurity
- SAML20 Valor predeterminado de Bearer WSSecurity
- SAML11 Valor predeterminado de HoK Public WSSecurity
- SAML20 Valor predeterminado de HoK Public WSSecurity
- SAML11 Valor predeterminado de HoK Symmetric WSSecurity
- SAML20 Valor predeterminado de HoK Symmetric WSSecurity
- Valor predeterminado de WSHTTPS de nombre de usuario

Si los directorios no se encuentran en el directorio de configuración de perfiles, cópielos en dicho directorio desde la siguiente ubicación:

*raíz\_servidor\_apl/profileTemplates/default/documents/config/templates/PolicySets*

Debe importar los conjuntos de políticas en la consola administrativa, seleccionar los que desea que estén disponibles para Integration Designer y luego guardar un archivo zip para cada uno de dichos conjuntos en una ubicación a la que Integration Designer tenga acceso.

1. Importe los conjuntos de políticas siguiendo estos pasos:
  - a. Desde la consola administrativa, pulse **Servicios > Conjuntos de políticas > Conjuntos de políticas de aplicaciones**.
  - b. Pulse **Importar > De repositorio predeterminado**.
  - c. Seleccione los conjuntos de políticas predeterminados SAML y pulse **Aceptar**.
2. Exporte los conjuntos de políticas de modo que puedan utilizarse en Integration Designer:
  - a. Desde la página de conjuntos de políticas de aplicación, seleccione el conjunto de políticas SAML que desea exportar y pulse **Exportar**.

**Nota:** Si la página de conjuntos de políticas de aplicación no se está visualizando, pulse **Servicios > Conjuntos de políticas > Conjuntos de políticas de aplicación** en la consola administrativa.

- b. En la siguiente página, pulse el enlace de archivo .zip para el conjunto de políticas.
- c. En la ventana de descarga de archivos, pulse **Guardar** e indique una ubicación a la que Integration Designer tenga acceso.
- d. Pulse **Atrás**.
- e. Siga los pasos 2a en la página 62 - 2d para cada conjunto de políticas que desee exportar.

Los conjuntos de políticas SAML se guardan en archivos .zip y están listos para ser importados en Integration Designer.

Importe los conjuntos de políticas en Integration Designer, como se describe en el tema “Conjuntos de políticas”.

*Invocación de servicios web que precisan de autenticación HTTP básica:*

La autenticación HTTP básica utiliza un nombre de usuario y contraseña para autenticar un cliente de servicio para un punto final seguro. Puede configurar la autenticación HTTP básica cuando se envía o reciben solicitudes de servicios web.

Configure la autenticación básica HTTP para las solicitudes de servicio web que se reciben mediante la configuración de los enlaces de exportación de la API de Java para servicios web XML (JAX-WS), según se describe en el tema “Creación y asignación de roles de seguridad a exportaciones de servicios web”.

La autenticación básica HTTP se puede habilitar para solicitudes de servicio web enviadas por un enlace de importación JAX-WS de dos formas posibles:

- Cuando se configura el enlace de importación en un módulo SCA, puede seleccionar el conjunto de políticas de autenticación HTTP proporcionado, denominado BPMHTTPBasicAuthentication (que se proporciona con los enlaces de importación (JAX-WS) de servicio web) u otro conjunto de políticas que incluya la política HTTPTransport.
- Cuando se construye el módulo SCA, puede utilizar las posibilidades del flujo de mediación para crear dinámicamente una nueva cabecera de autenticación HTTP y especificar la información de nombre de usuario y contraseña en la cabecera.

**Nota:** El conjunto de políticas tiene preferencia sobre el valor especificado en la cabecera. Si quiere utilizar el valor establecido en la cabecera de autenticación HTTP en tiempo de ejecución, no asocie un conjunto de políticas que incluya la política HTTPTransport. Concretamente, no utilice el conjunto de políticas BPMHTTPBasicAuthentication predeterminado y, si tiene definido un conjunto de políticas, asegúrese de que excluye la política HTTPTransport.

En el tema “Conjuntos de políticas de servicios web” del centro de información de WebSphere Application Server encontrará más información sobre los conjuntos de políticas y los enlaces de políticas de servicios web y cómo se utilizan. El enlace al tema está en “Información relacionada”.

- Para utilizar el conjunto de políticas proporcionado, realice los pasos siguientes: realice los pasos siguientes:
  1. Opcional: En la consola administrativa, cree un enlace de política general de cliente o edite uno existente que incluya la política HTTPTransport con los valores de ID de usuario y contraseña que quiera.
  2. En IBM Integration Designer, genere un enlace de importación de servicio web (JAX-WS) y asícielo al conjunto de políticas BPMHTTPBasicAuthentication.
  3. Realice *uno* de los pasos siguientes:

- En IBM Integration Designer, en las propiedades de enlace de importación de servicio web (JAX-WS), especifique el nombre del enlace de política general de cliente existente que incluye la política HTTPTransport.
  - Tras desplegar el módulo SCA, desde la consola administrativa del servidor de procesos, seleccione un enlace de política de cliente existente o cree uno nuevo y asócielo al enlace de importación.
4. Opcional: En la consola administrativa del servidor de procesos, edite el enlace de conjunto de políticas seleccionado para especificar el ID y la contraseña necesarios.
- Para especificar el nombre de usuario y contraseña en la cabecera de autenticación HTTP, realice uno de los grupos de pasos siguientes:
    - Utilice la primitiva de mediación HTTP Header Setter en IBM Integration Designer para crear la cabecera de autenticación y especifique el nombre de usuario y contraseña.
    - Si hiciera falta lógica adicional, utilice código Java en una primitiva de mediación personalizada (según se muestra en el ejemplo siguiente) para:
      1. Crear una cabecera de autenticación HTTP.
      2. Especificar la información de nombre de usuario y contraseña.
      3. Añadir la cabecera de autenticación HTTP a HTTPControl.
      4. Establecer el HTTPControl actualizado nuevamente en el servicio Context.

```
//Obtener HeaderInfoType de contextService
ContextService contextService = (ContextService) ServiceManager.INSTANCE
.locateService("com/ibm/bpm/context/ContextService");
HeaderInfoType headers = contextService.getHeaderInfo();
if(headers == null){
    headers = ContextObjectFactory.eINSTANCE.createHeaderInfoType();
}
//Obtener la cabecera HTTP y HTTP Control de HeaderInfoType
HTTPHeaderType httpHeaderType = headers.getHTTPHeader();
HTTPControl cp = httpHeaderType.getControl();
HeadersFactory factory = HeadersFactory.eINSTANCE;
if(cp == null){
    cp = factory.createHTTPControl();
}
//Crear la HTTPAuthentication nueva y establecer HTTPCredentials
HTTPAuthentication authorization = factory.createHTTPAuthentication();
HTTPCredentials credentials = factory.createHTTPCredentials();
authorization.setAuthenticationType(HTTPAuthenticationType.BASIC_LITERAL);
credentials.setUserId("USERNAME");
credentials.setPassword("PASSWORD");
authorization.setCredentials(credentials);
cp.setAuthentication(authorization);
httpHeaderType.setControl(cp);
// Establecer la información de cabecera nuevamente en el contexto de ejecución actual.
contextService.setHeaderInfo(headers);
```

*Verificación de la configuración del servidor:*

Si despliega una aplicación web con un enlace de servicio web (JAX-WS), debe asegurarse de que el servidor en el que despliega la aplicación no tenga la opción **Iniciar componentes según sea necesario** seleccionada.

Para verificar si esta opción está seleccionada, realice los siguientes pasos desde la consola administrativa:

1. Pulse **Servidores > Tipos de servidor > WebSphere Application Servers**.
2. Pulse el nombre del servidor.
3. Desde el separador Configuración, determine si la opción **Iniciar componentes según sea necesario** está seleccionada.
4. Realice uno de estos pasos:

- Si la opción **Iniciar componentes según sea necesario** está seleccionada, desmárquela y pulse **Aplicar**.
- Si la opción **Iniciar componentes según sea necesario** no está seleccionada, pulse **Cancelar**.

### Adjuntos en mensajes SOAP:

Puede enviar y recibir mensajes SOAP que incluyen datos binarios (como archivos PDF o imágenes JPEG) como adjuntos. Los adjuntos puede ser *referenciados* (es decir, representados explícitamente como partes de mensaje en la interfaz de servicio) o *no referenciados* (en los que se pueden incluir los números y tipos arbitrarios de adjuntos).

Un adjunto referenciado puede estar representado de una de las formas siguientes:

- Como un elemento `wsi:swaRef-typed` en el esquema de mensaje  
Los adjuntos definidos utilizando el tipo `wsi:swaRef` type son compatibles con Web Services Interoperability Organization (WS-I) *Attachments Profile Versión 1.0* (<http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>), que define cómo se relacionan elementos de mensaje con partes MIME.
- Como parte de mensaje de nivel superior, mediante un tipo de esquema binario  
Los adjuntos representados como partes de mensaje de nivel superior son compatibles con la especificación de *Mensajes SOAP con adjuntos* (<http://www.w3.org/TR/SOAP-attachments> ).  
También pueden configurarse adjuntos representados como partes de mensaje de nivel superior para garantizar que el documento y los mensajes WSDL producidos por el enlace cumplen la especificación *WS-I Attachments Profile versión 1.0* y la especificación *WS-I Basic Profile Versión 1.1* (<http://www.ws-i.org/Profiles/BasicProfile-1.1.html>).

Un adjunto no referenciado se transporta en un mensaje SOAP sin ninguna representación en el esquema del mensaje.

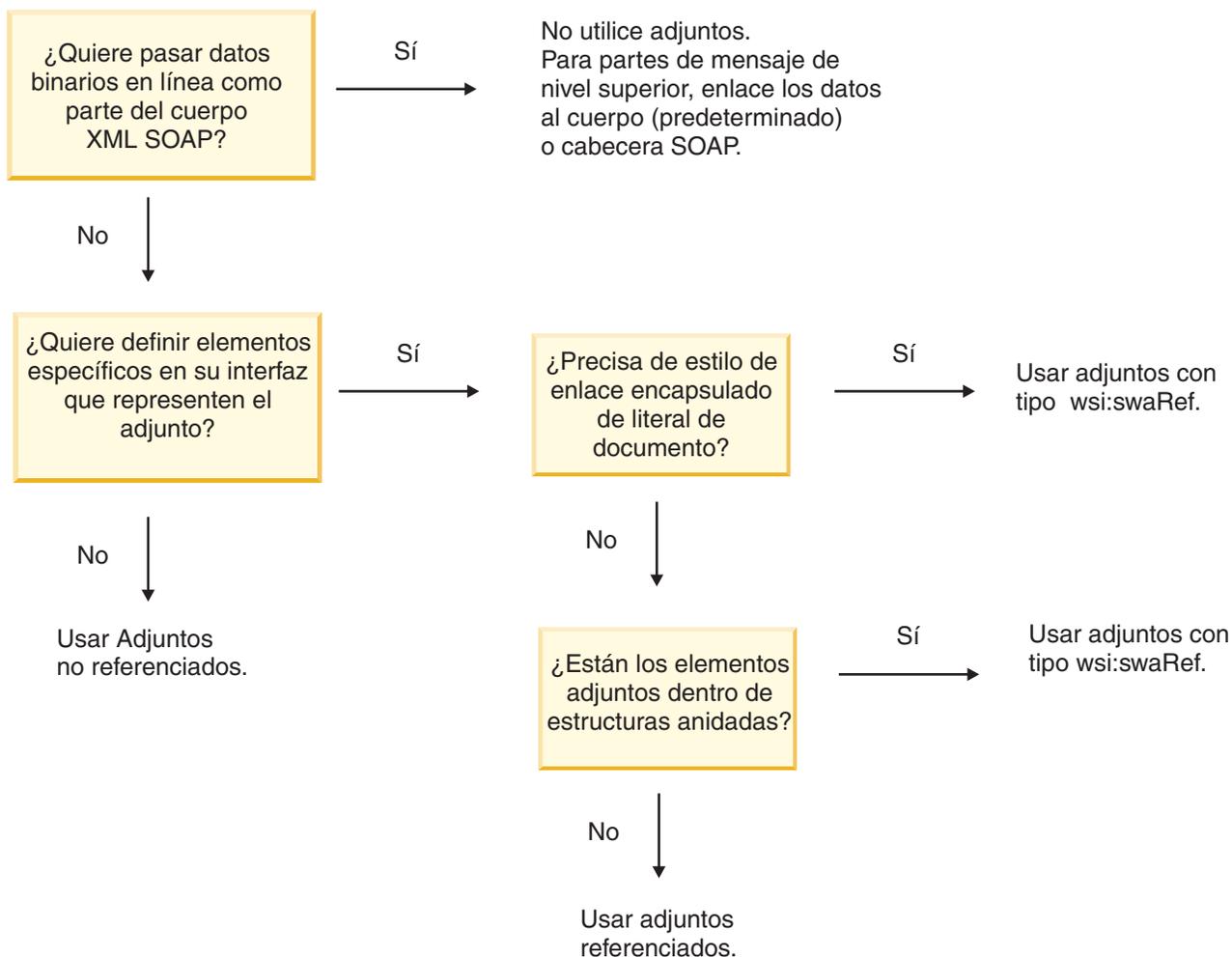
En todos los casos, el enlace SOAP de WSDL debe incluir un enlace MIME para los adjuntos a utilizar y el tamaño máximo de los archivos adjuntos no debe superar los 20 MB.

**Nota:** Para enviar o recibir mensajes SOAP con adjuntos, debe utilizar uno de los enlaces de servicio web basados en la API de Java para servicios web XML (JAX-WS).

*Cómo seleccionar el estilo de archivo adjunto adecuado:*

Cuando diseñe una nueva interfaz de servicio que incluya datos binarios, debe tener en cuenta cómo se transportan dichos datos binarios en los mensajes SOAP recibidos por el servicio.

Utilice el siguiente conjunto de preguntas para determinar el estilo de archivo adjunto adecuado:



*Adjuntos referenciados: elementos de tipo swaRef:*

Puede enviar y recibir mensajes SOAP que incluyan adjuntos representados en la interfaz de servicio como elementos de tipo swaRef.

Un elemento de tipo swaRef se define en Web Services Interoperability Organization (WS-I) *Attachments Profile* Versión 1.0 (<http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html>), que define cómo están relacionados los elementos de mensaje con partes MIME.

En el mensaje SOAP, el cuerpo SOAP contiene un elemento de tipo swaRef que identifica el ID de contenido del adjunto.

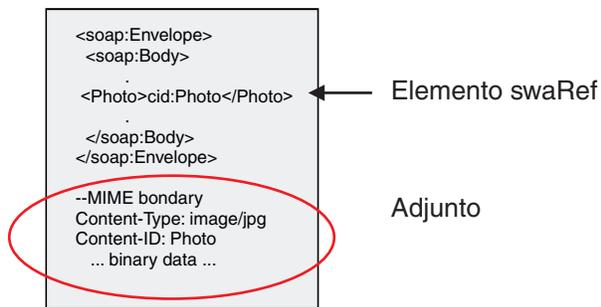


Figura 13. Un mensaje SOAP con un elemento swaRef

El WSDL para este mensaje SOAP contiene un elemento de tipo swaRef dentro de una parte de mensaje que identifica el adjunto.

```
<element name="sendPhoto">
  <complexType>
    <sequence>
      <element name="Photo" type="wsi:swaRef"/>
    </sequence>
  </complexType>
</element>
```

El WSDL también debe contener un enlace MIME que indica los mensajes de varias partes MIME que se deben utilizar.

**Nota:** El WSDL *no* incluya un enlace MIME para el elemento de mensaje de tipo swaRef específico, porque los enlaces MIME sólo se aplican en las partes de mensaje de nivel superior.

Los adjuntos representados como elementos de tipo swaRef se pueden propagar sólo entre componentes de flujo de mediación. Si otro tipo de componente debe acceder a un archivo adjunto, o el archivo adjunto se debe propagar a éste, utilice un componente de flujo de mediación para mover el archivo adjunto a una ubicación a la que pueda acceder ese componente.

### Proceso de entrada de adjuntos

Utilice Integration Designer para configurar un enlace de exportación para recibir el adjunto. Cree un módulo y sus operaciones e interfaz asociadas, que incluyen un elemento de tipo swaRef. Cree un enlace de servicio web (JAX-WS).

**Nota:** Consulte el tema “Trabajo con adjuntos” en el centro de información de Integration Designer si desea más información.

Cuando un cliente pasa un mensaje SOAP con un adjunto swaRef a un componente SCA (Service Component Architecture), en primer lugar, el enlace de exportación de servicio web (JAX-WS) elimina el adjunto. A continuación, analiza la parte SOAP del mensaje y crea un objeto de negocio. Finalmente, el enlace establece el ID de contenido del adjunto en el objeto de negocio.

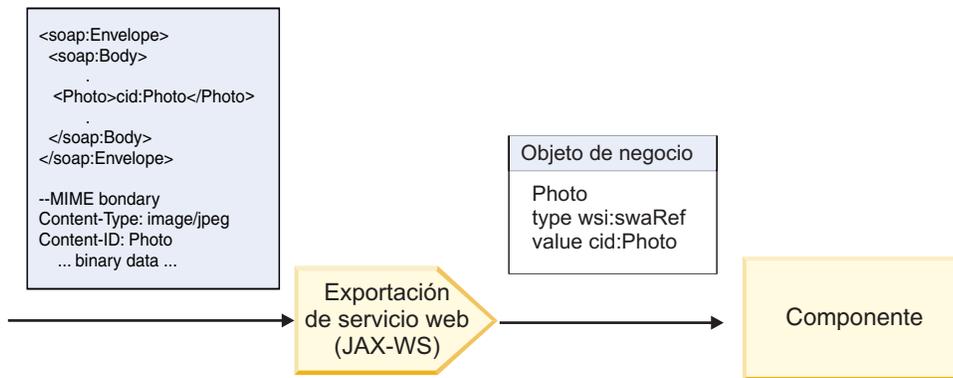


Figura 14. Cómo el enlace de exportación de servicio web (JAX-WS) procesa un mensaje SOAP con un adjunto swaRef

### Acceso a los metadatos de adjunto en un componente de flujo de mediación

Tal como se indica en Figura 15, cuando los componentes acceden a los adjuntos swaRef, el identificador del contenido del adjunto aparece como un elemento del tipo swaRef.

Cada adjunto de un mensaje SOAP también tiene un elemento **attachments** correspondiente en SMO. Cuando se utiliza el tipo WS-I swaRef, el elemento **attachments** incluye el tipo de contenido de adjunto y el ID de contenido, así como los datos binarios reales del adjunto.

Para obtener el valor de un adjunto swaRefm, por lo tanto, es necesario obtener el valor del elemento del tipo swaRef y, a continuación, localizar el elemento **attachments** con el valor **contentID** correspondiente. Tenga en cuenta que el valor **contentID** normalmente tiene el prefijo **cid:** eliminado del valor swaRef.

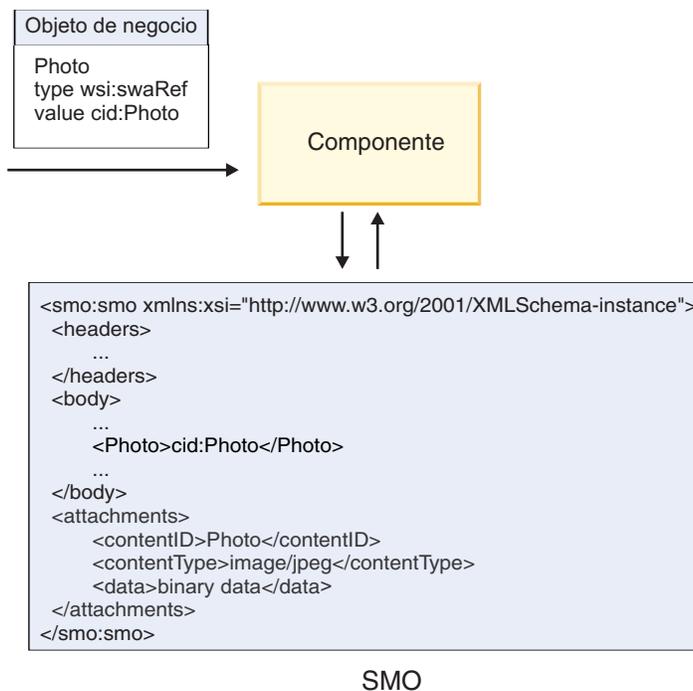


Figura 15. Cómo aparecen los adjuntos swaRef en SMO

## Proceso de salida

Utilice Integration Designer para configurar un enlace de importación de servicio web (JAX-WS) para invocar un servicio web externo. El enlace de importación se configura con un documento WSDL que describe el servicio web que se debe invocar y define el adjunto que se pasará al servicio web.

Cuando un enlace de importación de servicio web (JAX-WS) recibe un mensaje SCA, los elementos de tipo `swaRef` se envían como adjuntos, si la importación se conecta a un componente de flujo de mediación y el elemento de tipo `swaRef` tiene un elemento **attachments** correspondiente.

Para el proceso de salida, los elementos de tipo `swaRef` siempre se envían con sus valores de ID de contenido; sin embargo, el módulo de mediación debe asegurarse de que hay un elemento **attachments** correspondiente con un valor **contentID** coincidente.

**Nota:** Para ajustarse a WS-I Attachments Profile, el valor **content ID** deberá seguir la "codificación de la parte de id de contenido", tal como se describe en la sección 3.8 de *WS-I Attachments Profile 1.0*.

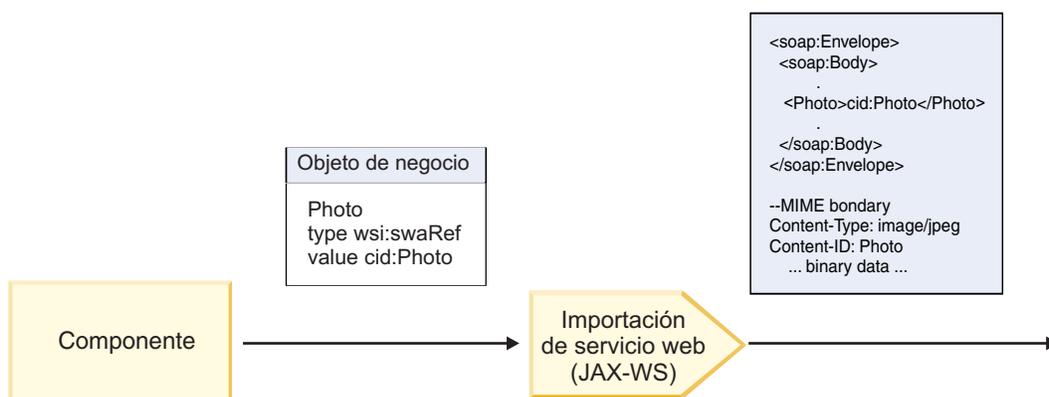


Figura 16. Cómo genera el enlace de importación de servicio web (JAX-WS) un mensaje SOAP con un adjunto `swaRef`

## Configuración de los metadatos de adjuntos en un componente de flujo de mediación

Si, en SMO, existe un valor del elemento del tipo `swaRef` y un elemento **attachments**, el enlace prepara el mensaje SOAP (con el adjunto) y lo envía a un destinatario.

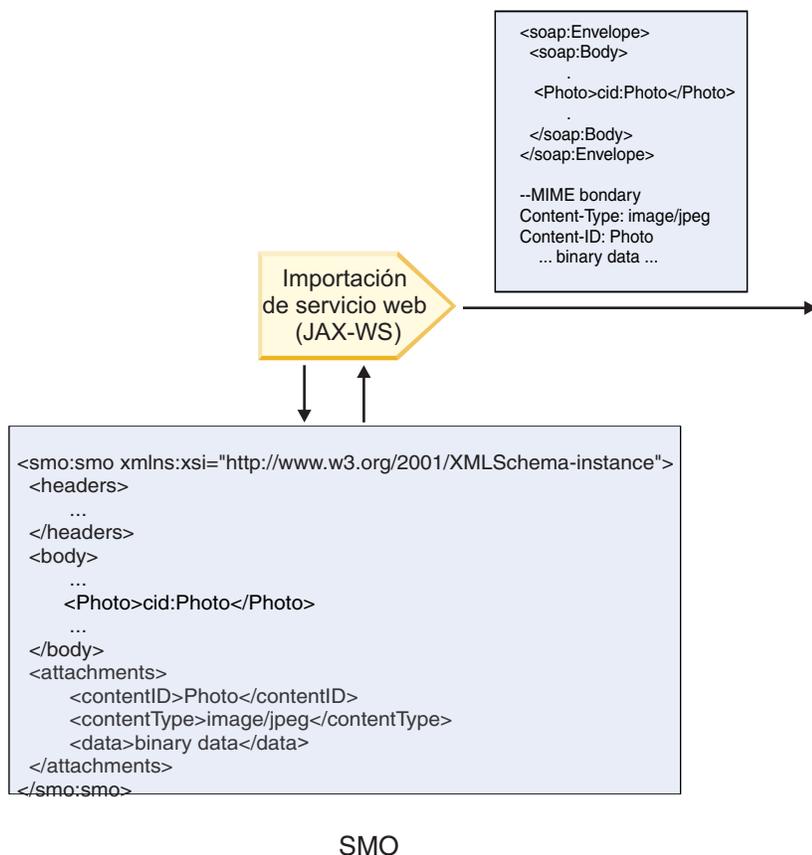


Figura 17. Cómo se accede a un adjunto swaRef en SMO para crear el mensaje SOAP

El elemento **attachments** está presente en SMO sólo si un componente de flujo de mediación está conectado directamente a la importación o exportación; no pasa entre otros tipos de componente. Si los valores son necesarios en un módulo que contiene otros tipos de componente, se debe utilizar un componente de flujo de mediación para copiar los valores en una ubicación en el módulo en la que se pueda acceder a éstos y otro componente de flujo de mediación utilizado para establecer los valores correctos antes de una invocación de salida mediante una importación de servicio web.

**Importante:** Tal y como se describe en “Representación XML de SMO,” la primitiva de mediación Transformación XSL transforma mensajes mediante una transformación XSLT 1.0. La transformación opera en una serialización XML del SMO. La primitiva de mediación Transformación XSL permite especificar la raíz de la serialización, y el elemento raíz del documento XML refleja esta raíz.

Cuando se envían mensajes SOAP con archivos adjuntos, el elemento raíz que elija determina qué archivos adjuntos se propagan.

- Si se utiliza “/body” como la raíz de la correlación XML, se propagan de forma predeterminada todos los archivos adjuntos por la correlación.
- Si se utiliza “/” como la raíz de la correlación, se puede controlar la propagación de los archivos adjuntos.

*Adjuntos referenciados: partes de mensaje de nivel superior:*

Puede enviar y recibir mensajes SOAP que incluyan adjuntos binarios que estén declarados como partes de la interfaz de servicio.

En un mensaje SOAP de varias partes MIME, el cuerpo SOAP es la primera parte del mensaje y el archivo adjunto o los archivos adjuntos se encuentran en las partes subsiguientes.

¿Qué ventajas tiene enviar o recibir un adjunto referenciado en un mensaje SOAP? Los datos binarios que componen el adjunto (que generalmente es muy grande) se mantienen de forma separada al cuerpo del mensaje SOAP, de modo que no es necesario que se analicen como XML. Esto da como resultado un proceso más eficaz que si los datos binarios se mantienen dentro de un elemento XML.

### Tipos de mensajes SOAP con archivos adjuntos referenciados

A partir de la Versión 7.0.0.3 de IBM Business Process Manager, tiene la opción de elegir cómo se genera el mensaje SOAP:

- **Mensajes compatibles con WS-I**

En tiempo de ejecución se pueden generar mensajes SOAP que cumplan con *WS-I Attachments Profile versión 1.0* y con *WS-I Basic Profile versión 1.1*. En un mensaje SOAP en conformidad con estos perfiles, solamente una parte del mensaje está enlazada al cuerpo SOAP; para los que estén enlazados como adjuntos, la codificación de la parte de id de contenido (como se describe en *WS-I Attachments Profile Version 1.0*) se utiliza para relacionar el archivo adjunto con la parte del mensaje.

- **Mensajes no compatibles con WS-I**

En tiempo de ejecución se pueden generar mensajes SOAP que no sean compatibles con perfiles WS-I pero que sí sean compatibles con los mensajes generados en la versión 7.0 o 7.0.0.2 de IBM Business Process Manager. Los mensajes SOAP utilizan elementos de nivel superior denominados según la parte del mensaje con un atributo **href** que contiene el adjunto **content-id**, pero la codificación de la parte de id de contenido (tal como se describe en *WS-I Attachments Profile Version 1.0*) no se utiliza.

### Selección de conformidad con WS-I para exportaciones de servicio web

Puede utilizar Integration Designer para configurar un enlace de exportación. Cree un módulo y sus operaciones e interfaz asociadas. Cree un enlace de servicio web (JAX-WS). La página Adjuntos referenciados muestra todas las partes binarias de la operación creada y puede seleccionar qué partes serán adjuntos. A continuación, especificará en la página Especificar la conformidad con WS-I AP 1.0 de Integration Designer, una de las siguientes opciones:

- **Utilizar mensaje SOAP en conformidad con WS-I AP 1.0**

Si selecciona esta opción, también especificará qué parte del mensaje deberá enlazarse al cuerpo SOAP.

**Nota:** Esta opción sólo puede utilizarse cuando el archivo WSDL correspondiente también es compatible con WS-I.

Un archivo WSDL generado por Integration Designer Versión 7.0.0.3 está en conformidad con WS-I. No obstante, si importa un archivo WSDL que no esté en conformidad con WS-I, no podrá seleccionar esta opción.

- **Utilizar mensaje SOAP no en conformidad con WS-I AP 1.0**

Si selecciona esta opción, que es la predeterminada, la primera parte del mensaje se enlaza al cuerpo SOAP.

**Nota:** Sólo las partes del mensaje de nivel superior (es decir, los elementos definidos en `portType` de WSDL como partes del mensaje de entrada o salida) que tienen un tipo binario (ya sea `base64Binary` o `hexBinary`) se pueden enviar o recibir como adjuntos referenciados.

Consulte el tema “Trabajo con adjuntos” en el Information Center de Integration Designer para obtener información más detallada.

Para los mensajes en conformidad con WS-I, el ID de contenido que se genera en el mensaje SOAP es una concatenación de los siguientes elementos:

- El valor del atributo **name** del elemento **wsdl:part** referenciado por **mime:content**
- El carácter =
- Un valor exclusivo globalmente, por ejemplo un UUID

- El carácter @
- Un nombre de dominio válido

### Proceso de entrada de adjuntos referenciados

Cuando un cliente pasa un mensaje SOAP con un adjunto a un componente SCA (Service Component Architecture), el enlace de exportación de servicios Web (JAX-WS) en primer lugar elimina el adjunto. A continuación, analiza la parte SOAP del mensaje y crea un objeto de negocio. Finalmente, el enlace establece el binario del adjunto en el objeto de negocio.

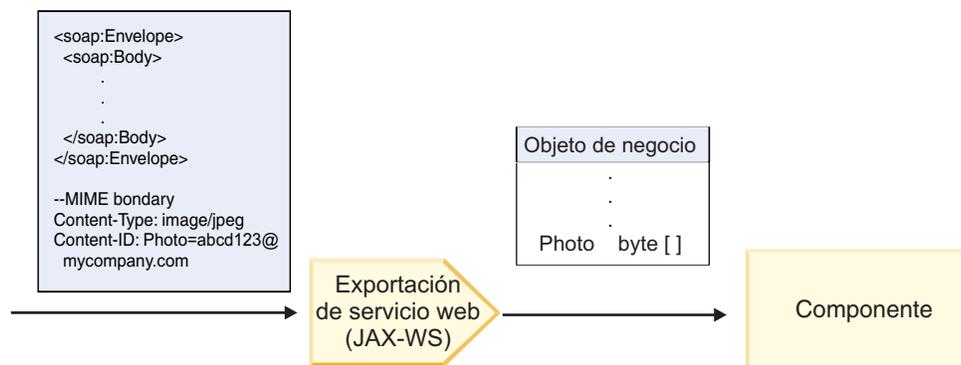


Figura 18. Cómo el enlace de exportación de servicio web (JAX-WS) procesa un mensaje SOAP en conformidad con WS-I con un adjunto referenciado

### Acceso a los metadatos de adjunto en un componente de flujo de mediación

Como se muestra en la Figura 18, cuando los componentes acceden a los adjuntos referenciados, los datos del adjunto aparecen como una matriz de bytes.

Cada adjunto referenciado de un mensaje SOAP también tiene un elemento **attachments** en el SMO. El elemento **attachments** incluye el tipo de contenido del adjunto y la vía de acceso al elemento del cuerpo del mensaje donde está contenido el adjunto.

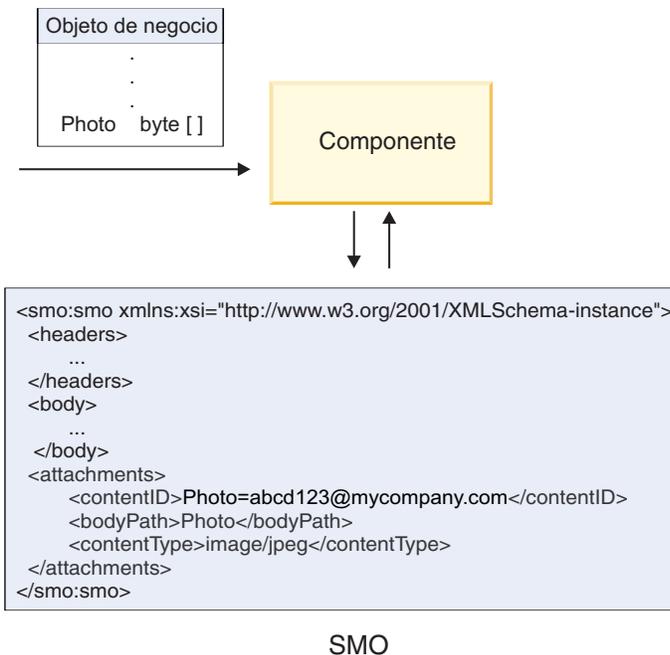


Figura 19. Cómo aparecen los adjuntos referenciados en SMO

**Importante:** La vía de acceso al elemento del cuerpo del mensaje no se actualiza automáticamente si el mensaje se transforma y se mueve el adjunto. Puede utilizar un flujo de mediación para actualizar el elemento **attachments** con la nueva vía de acceso (por ejemplo, como parte de la transformación o utilizando un setter del elemento de mensaje separado).

### Cómo se construyen los mensajes SOAP salientes

Utilice Integration Designer para configurar un enlace de importación de servicio web (JAX-WS) para invocar un servicio web externo. El enlace de importación se configura con un documento WSDL que describe el servicio web que se debe invocar y define qué partes del mensaje se deben pasar como adjuntos. También puede indicar, en la página Especificar la conformidad con WS-I AP 1.0 de Integration Designer, una de las siguientes opciones:

- **Utilizar mensaje SOAP en conformidad con WS-I AP 1.0**  
Si selecciona esta opción, también especificará qué parte del mensaje deberá enlazarse al cuerpo SOAP; todas las demás se enlazan a adjuntos o cabeceras. Los mensajes enviados por el enlace no incluyen elementos en el cuerpo SOAP que hagan referencia a los adjuntos; la relación se expresa por medio del ID de contenido del adjunto que incluye el nombre de la parte del mensaje.
- **Utilizar mensaje SOAP no en conformidad con WS-I AP 1.0**  
Si selecciona esta opción, que es la predeterminada, la primera parte del mensaje se enlaza al cuerpo SOAP; todas las demás se enlazan a adjuntos o cabeceras. Los mensajes enviados por el enlace incluyen uno o varios elementos en el cuerpo SOAP que hacen referencia a los adjuntos por medio de un atributo **href**.

**Nota:** La parte que representa un adjunto, tal como se define en el WSDL, debe ser un tipo sencillo (ya sea base64Binary o hexBinary). Si una parte se ha definido mediante complexType, dicha parte no puede enlazarse como un adjunto.

## Proceso de salida de adjuntos referenciados

El enlace de importación utiliza la información en el SMO para determinar cómo se envían las partes binarias del mensaje de nivel superior como adjuntos.

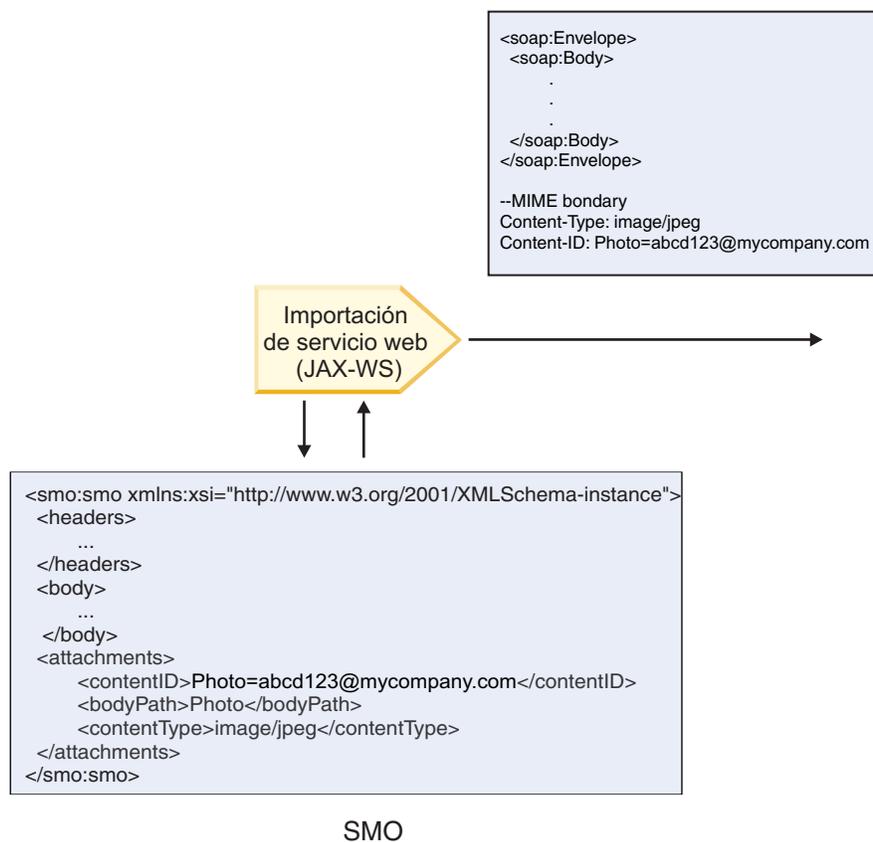


Figura 20. Cómo se accede al adjunto referenciado en SMO para crear el mensaje SOAP

El elemento **attachments** está presente en el SMO únicamente si un componente del flujo de mediación está conectado directamente a la importación o exportación. No se pasa a través de otros tipos de componentes. Si los valores son necesarios en un módulo que contiene otros tipos de componentes, se debe utilizar un componente de flujo de mediación para copiar los valores en una ubicación del módulo a la que se pueda acceder, y se debe utilizar otro componente de flujo de mediación para establecer los valores correctos antes de realizar una invocación de salida mediante una importación de servicios Web.

El enlace utiliza una combinación de las siguientes condiciones para determinar cómo (o si) se envía el mensaje.

- Si existe un enlace MIME de WSDL para la parte binaria de nivel superior del mensaje y, si existey, cómo se define el tipo de contenido
- Si existe un elemento **attachments** en el SMO cuyo valor **bodyPath** hace referencia a una parte binaria de nivel superior

## Cómo se crean los adjuntos cuando existe un elemento attachment en el SMO

La tabla siguiente muestra cómo se crea y se envía un adjunto si el SMO contiene un elemento **attachment** con un **bodyPath** que coincide con una parte de nombre de mensaje:

Tabla 20. Cómo se genera el adjunto

Estado del enlace MIME de WSDL para la parte binaria de nivel superior del mensaje	Cómo se crea y se envía el mensaje
Presente con una de las acciones siguientes: <ul style="list-style-type: none"> <li>• No hay tipo de contenido definido para la parte del mensaje</li> <li>• Hay definidos varios tipos de contenido</li> <li>• Tipo de contenido comodín definido</li> </ul>	La parte de mensaje se envía como un adjunto.  <b>Content-Id</b> se establece en el valor contenido en el elemento attachments si existe; de lo contrario, se genera uno.  <b>Content-Type</b> se establece en el valor contenido en el elemento attachments si existe; de lo contrario, se establece en <b>application/octet-stream</b> .
Presente con contenido único, sin caracteres comodín, para la parte del mensaje	La parte del mensaje se envía como adjunto.  <b>Content-Id</b> se establece en el valor contenido en el elemento attachments si existe; de lo contrario, se genera uno.  <b>Content-Type</b> se establece en el valor contenido en el elemento attachments si existe; de lo contrario, se establece en el tipo definido en el elemento de contenido MIME de WSDL.
No presente	La parte de mensaje se envía como adjunto.  <b>Content-Id</b> se establece en el valor contenido en el elemento attachments si existe; de lo contrario, se genera uno.  <b>Content-Type</b> se establece en el valor contenido en el elemento attachments si existe; de lo contrario, se establece en <b>application/octet-stream</b> . <b>Nota:</b> Enviar partes de mensaje como adjuntos cuando no se han definido como tales en WSDL puede romper la conformidad con WS-I Attachments Profile 1.0, por lo que debe evitarse si es posible.

### Cómo se crean adjuntos cuando no existe un elemento attachment en el SMO

La tabla siguiente muestra cómo se crea y envía un adjunto si el SMO no contiene un elemento **attachment** con un **bodyPath** que coincide con una parte de nombre de mensaje:

Tabla 21. Cómo se genera el adjunto

Estado del enlace MIME de WSDL para la parte binaria de nivel superior del mensaje	Cómo se crea y se envía el mensaje
Presente con una de las acciones siguientes: <ul style="list-style-type: none"> <li>• No hay tipo de contenido definido para la parte del mensaje</li> <li>• Hay definidos varios tipos de contenido</li> <li>• Tipo de contenido comodín definido</li> </ul>	La parte de mensaje se envía como un adjunto.  Se genera <b>Content-Id</b> .  <b>Content-Type</b> se establece en <b>application/octet-stream</b> .
Presente con contenido único, sin caracteres comodín, para la parte del mensaje	La parte de mensaje se envía como un adjunto.  Se genera <b>Content-Id</b> .  <b>Content-Type</b> se establece en el tipo definido en el elemento de contenido MIME de WSDL.
No presente	La parte del mensaje no se envía como un adjunto.

**Importante:** Tal y como se describe en “Representación XML de SMO,” la primitiva de mediación Transformación XSL transforma mensajes mediante una transformación XSLT 1.0. La transformación opera en una serialización XML del SMO. La primitiva de mediación Transformación XSL permite especificar la raíz de la serialización, y el elemento raíz del documento XML refleja esta raíz.

Cuando se envían mensajes SOAP con archivos adjuntos, el elemento raíz que elija determina qué archivos adjuntos se propagan.

- Si se utiliza “/body” como la raíz de la correlación XML, se propagan de forma predeterminada todos los archivos adjuntos por la correlación.
- Si se utiliza “/” como la raíz de la correlación, se puede controlar la propagación de los archivos adjuntos.

*Adjuntos no referenciados:*

Puede enviar y recibir archivos adjuntos *no referenciados* que no estén declarados en la interfaz de servicio.

En un mensaje SOAP de varias partes MIME, el cuerpo SOAP es la primera parte del mensaje y los archivos adjuntos se encuentran en las partes subsiguientes. No se incluye ninguna referencia al archivo adjunto en el cuerpo SOAP.

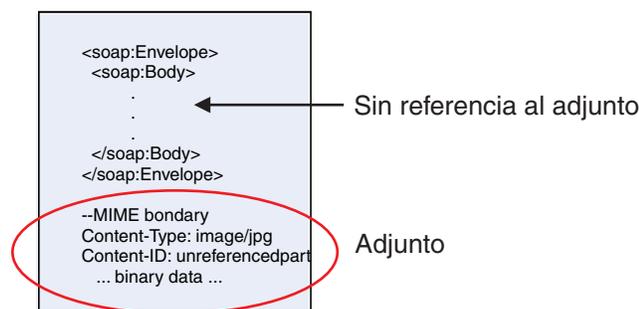


Figura 21. Un mensaje SOAP con un archivo adjunto no referenciado

Puede enviar un mensaje SOAP con un archivo adjunto no referenciado mediante una exportación de servicio Web a una importación de servicio Web. El mensaje de salida, que se envía al servicio Web de destino, contiene el archivo adjunto.

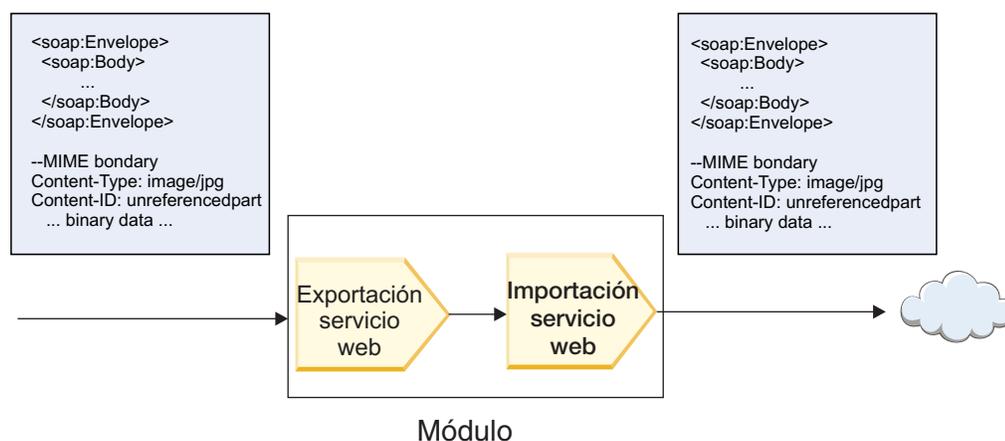


Figura 22. Un archivo adjunto que pasa a través de un módulo SCA

En la Figura 22, el mensaje SOAP, con el archivo adjunto, pasa sin modificaciones.

También puede modificar el mensaje SOAP utilizando un componente de flujo de mediación. Por ejemplo, puede utilizar el componente de flujo de mediación para extraer datos del mensaje SOAP (datos binarios en el cuerpo del mensaje, en este caso) y crear un mensaje SOAP con archivos adjuntos. Los datos se procesan como parte del elemento de archivos adjuntos de un objeto de mensaje de servicio (SMO).

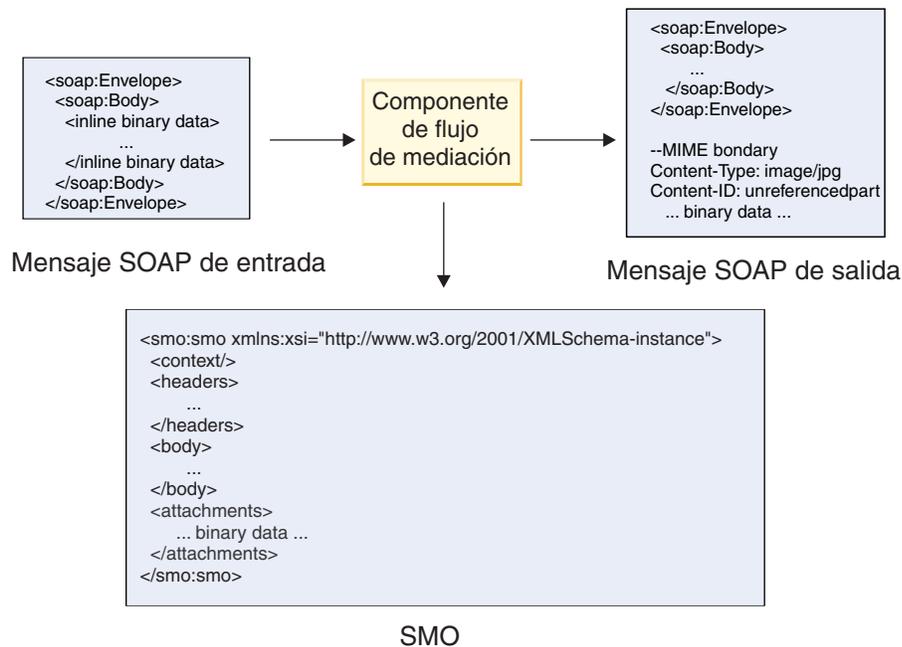


Figura 23. Un mensaje procesado por un componente de flujo de mediación

En sentido inverso, el componente de flujo de mediación puede transformar el mensaje entrante extrayendo y codificando el archivo adjunto y, a continuación, transmitiendo el mensaje sin archivos adjuntos.

En lugar de extraer los datos de un mensaje SOAP entrante para formar un mensaje SOAP con archivos adjuntos, puede obtener los datos de archivo adjunto de un origen externo, como por ejemplo una base de datos.

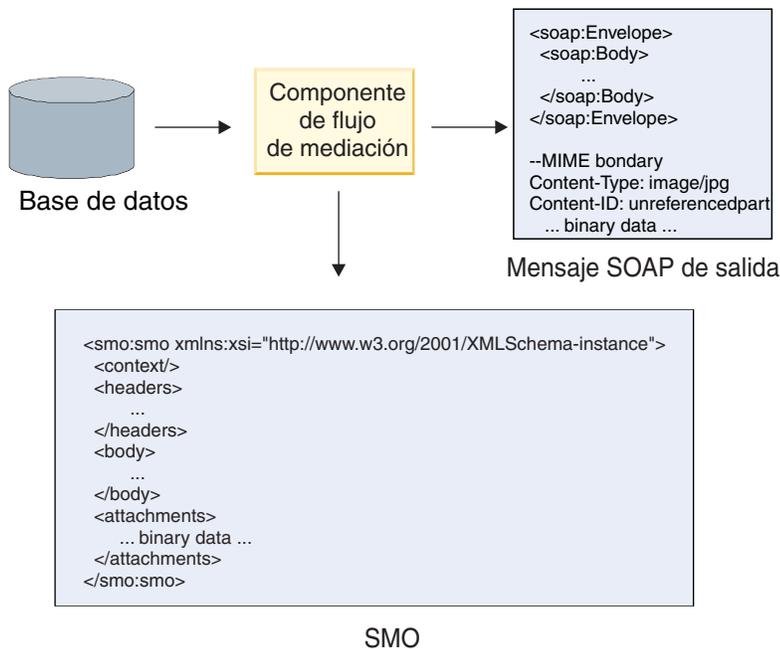


Figura 24. Un archivo adjunto obtenido de una base de datos y añadido al mensaje SOAP

En sentido inverso, el componente de flujo de mediación puede extraer el archivo adjunto de un mensaje SOAP entrante y procesar el mensaje (por ejemplo, almacenar el archivo adjunto en una base de datos).

Los archivos adjuntos no referenciados se pueden propagar sólo entre componentes de flujo de mediación. Si otro tipo de componente debe acceder a un archivo adjunto, o el archivo adjunto se debe propagar a éste, utilice un componente de flujo de mediación para mover el archivo adjunto a una ubicación a la que pueda acceder ese componente.

**Importante:** Tal y como se describe en “Representación XML de SMO,” la primitiva de mediación Transformación XSL transforma mensajes mediante una transformación XSLT 1.0. La transformación opera en una serialización XML del SMO. La primitiva de mediación Transformación XSL permite especificar la raíz de la serialización, y el elemento raíz del documento XML refleja esta raíz.

Cuando se envían mensajes SOAP con archivos adjuntos, el elemento raíz que elija determina qué archivos adjuntos se propagan.

- Si se utiliza “/body” como la raíz de la correlación XML, se propagan de forma predeterminada todos los archivos adjuntos por la correlación.
- Si se utiliza “/” como la raíz de la correlación, se puede controlar la propagación de los archivos adjuntos.

#### Uso del enlace de estilo de documento WSDL con mensajes de varias partes:

La organización WS-I (Web Services Interoperability - Interoperatividad de servicios web) tiene definido un conjunto de normas en relación a la manera en que se deben describir los servicios Web por medio de un WSDL y a la manera en se deben formar los mensajes SOAP correspondientes, a fin de asegurar la interoperatividad.

Estas normas se especifican en *Basic Profile Versión 1.1* de WS-I (<http://www.ws-i.org/Profiles/BasicProfile-1.1.html>). Concretamente, WS-I Basic Profile 1.1 R2712 indica: “Un enlace documento-literal DEBE estar serializado como ENVELOPE con un soap:Body cuyo elemento hijo sea una instancia de la declaración de elemento global a la que hace referencia la wsdl:message part correspondiente.”

Esto significa que, cuando se utiliza un enlace SOAP de estilo de documento para una operación con mensajes (de entrada, salida o error) que se definen con varias partes, sólo se deberá enlazar una de esas partes al cuerpo de SOAP a fin de ajustarse a los estándares de WS-I Basic Profile 1.1.

Además, WS-I Attachments Profile 1.0 R2941 indica: “Un wsdl:binding en una DESCRIPTION DEBERÁ enlazar cada wsdl:part de un wsdl:message en el wsdl:portType al que hace referencia a uno de los siguientes: soapbind:body, soapbind:header, soapbind:fault , soapbind:headerfault, o mime:content.”.

Esto significa que, cuando se utiliza un enlace SOAP de estilo de documento para una operación con mensajes (de entrada, salida o error) que se definen con varias partes, todas las partes que no sean la seleccionada para enlazar al cuerpo SOAP deben enlazarse como adjuntos o cabeceras.

El planteamiento siguiente se utiliza cuando se generan descripciones WSDL para exportaciones con enlaces de servicio Web (JAX-WS y JAX-RPC) en este caso:

- Puede elegir qué parte de mensaje se enlaza al cuerpo SOAP si hay más de un elemento de tipo no binario. Si hay un único elemento de tipo no binario, ese elemento se enlaza automáticamente al cuerpo SOAP.
- Para el enlace JAX-WS, todas las demás partes de mensaje de tipo "hexBinary" o "base64Binary" se enlazan como adjuntos referenciados. Consulte “Adjuntos referenciados: partes de mensaje de nivel superior” en la página 70.
- Todas las demás partes de mensaje se enlaza como cabeceras SOAP.

Los enlaces de importación JAX-RPC y JAX-WS se ajustan a los estándares del enlace SOAP en un documento WSDL existente con mensajes de estilo de documento de varias partes incluso si éste enlaza varias partes al cuerpo SOAP; sin embargo, no puede generar clientes de servicio Web para dichos documentos WSDL en Rational Application Developer.

**Nota:** El enlace JAX-RPC no soporta adjuntos.

Por consiguiente el patrón recomendado cuando se utilizan mensajes de varias partes con una operación que tiene enlace SOAP de estilo de documento es:

1. Utilice el estilo incluido de documento/literal. En este caso, los mensajes siempre tienen una sola parte; sin embargo, los adjuntos no deben tener referencia (como se describe en “Adjuntos no referenciados” en la página 76) o pueden ser de tipo swaRef (como se describe en “Adjuntos referenciados: elementos de tipo swaRef” en la página 66) en este caso.
2. Utilice el estilo de RPC/literal. En este caso, no hay restricciones en el enlace WSDL en términos de número de partes enlazadas al cuerpo SOAP; el mensaje SOAP que se produce tiene siempre un solo hijo que representa la operación que se está invocando, donde las partes de mensaje son hijos de ese elemento.
3. Para el enlace JAX-WS, tenga como máximo una parte de mensaje no sea de tipo "hexBinary" o "base64Binary", a menos que sea aceptable enlazar las demás partes no binarias a cabeceras SOAP.
4. Cualquier otro caso está sujeto al comportamiento descrito.

**Nota:** Existen restricciones adicionales al utilizar mensajes SOAP que no se ajusten a *WS-I Basic Profile Version 1.1*.

- La primera parte del mensaje no debe ser binaria.
- Al recibir mensajes SOAP de estilo de documento de varias partes con adjuntos referenciados, el enlace JAX-WS espera que cada adjunto referenciado esté representado por un elemento hijo de cuerpo SOAP con un valor de atributo href que identifica al adjunto por su ID de contenido. El enlace JAX-WS envía adjuntos referenciados para dichos mensajes del mismo modo. Este comportamiento no se ajusta a los estándares de WS-I Basic Profile.

Para asegurar que los mensajes se ajustan al Basic Profile, siga el planteamiento 1 en la página 79 o 2 en la página 79 de la lista anterior o evite el uso de adjuntos referenciados para dichos mensajes y utilice en su lugar adjuntos no referenciados o de tipo swaRef.

## Enlaces HTTP

El enlace HTTP está diseñado para proporcionar conectividad SCA (Service Component Architecture) con HTTP. En consecuencia, las aplicaciones HTTP existentes o recién desarrolladas pueden participar en entornos de arquitectura orientada a servicios (SOA).

HTTP (protocolo de transporte de hipertexto) es un protocolo ampliamente utilizado para transportar información en la Web. Cuando trabaje con una aplicación externa que utilice el protocolo HTTP, necesitará un enlace HTTP. El enlace HTTP maneja la transformación de los datos pasados, como un mensaje en un formato nativo, a un objeto de negocio de una aplicación SCA. El enlace HTTP también puede transformar los datos extraídos como un objeto de negocio, al formato nativo que espera la aplicación externa, para un mensaje de entrada.

**Nota:** Si desea interactuar con clientes y servicios que utilizan el protocolo SOAP/HTTP de servicios Web, tenga en cuenta la posibilidad de utilizar uno de los enlaces de servicio Web, que proporcionan funcionalidad adicional respecto al manejo de calidades de servicio estándares de servicios Web.

En la lista siguiente se describen algunos escenarios comunes para utilizar el enlace HTTP:

- Los servicios alojados en SCA pueden invocar aplicaciones HTTP utilizando una importación HTTP.
- Los servicios alojados en SCA pueden exponerse a sí mismos como aplicaciones habilitadas para HTTP, de modo que los clientes HTTP pueden utilizarlos, mediante una exportación HTTP.
- IBM Business Process Manager y Process Server pueden comunicarse entre sí a través de una infraestructura HTTP y, por lo tanto, los usuarios pueden gestionar sus comunicaciones según los estándares corporativos.
- IBM Business Process Manager y Process Server pueden actuar como mediadores de comunicaciones HTTP, transformando y direccionando los mensajes, que mejora la integración de aplicaciones que utilicen una red HTTP.
- IBM Business Process Manager y Process Server se pueden utilizar para hacer de puente entre el protocolo HTTP y otros protocolos como, por ejemplo, SOAP/servicios Web HTTP, adaptadores de recursos basados en JCA (Java Connector Architecture), JMS, etc.

Puede encontrar información detallada sobre la creación de enlaces de importación y exportación HTTP en el Centro de información de Integration Designer. Consulte los temas **Desarrollo de aplicaciones de integración > Acceso a servicios externos con HTTP**.

### Visión general de enlaces HTTP:

El enlace HTTP proporciona conectividad a las aplicaciones alojadas bajo un protocolo HTTP. Su función es arbitrar la comunicación entre las aplicaciones HTTP y permite que, desde un módulo, se pueda llamar a las aplicaciones alojadas bajo un protocolo HTTP.

### Enlaces de importación HTTP

El enlace de importación HTTP proporciona conectividad de salida desde aplicaciones SCA (Service Component Architecture) a un servidor o aplicaciones HTTP.

La importación invoca un URL de punto final HTTP. El URL se puede especificar de una de tres maneras:

- El URL se puede establecer dinámicamente en las cabeceras HTTP por medio del URL de alteración dinámico.
- El URL se puede establecer dinámicamente en el elemento de dirección de destino SMO.
- El URL se puede especificar como una propiedad de configuración en la importación.

Esta invocación, por naturaleza, siempre es síncrona.

Aunque las invocaciones HTTP son siempre del tipo petición-respuesta, la importación HTTP da soporte tanto a las operaciones unidireccionales como bidireccionales, y no tiene en cuenta la respuesta en el caso de que se trate de una operación unidireccional.

### Enlaces de exportación HTTP

El enlace de exportación HTTP proporciona conectividad de entrada de aplicaciones HTTP a una aplicación SCA.

Un URL se define en la exportación HTTP. Las aplicaciones HTTP que quieren enviar mensajes de petición a la exportación, utilizan este URL para invocar la exportación.

La exportación HTTP también da soporte a los mandatos ping.

### Enlaces HTTP en el tiempo de ejecución

Una importación con un enlace HTTP en tiempo de ejecución envía una petición, con o sin datos, en el cuerpo del mensaje desde la aplicación SCA al servicio Web externo. La petición se efectúa desde la aplicación SCA al servicio Web externo, tal como se muestra en la Figura 25.

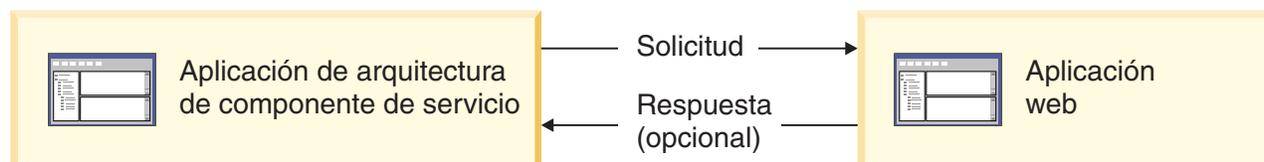


Figura 25. Flujo de una petición de la aplicación SCA a la aplicación Web.

Opcionalmente, la importación con el enlace HTTP puede recibir datos procedentes de la aplicación Web, en respuesta a la petición.

Con una exportación, una aplicación cliente efectúa la petición a un servicio Web, tal como se muestra en la Figura 26.

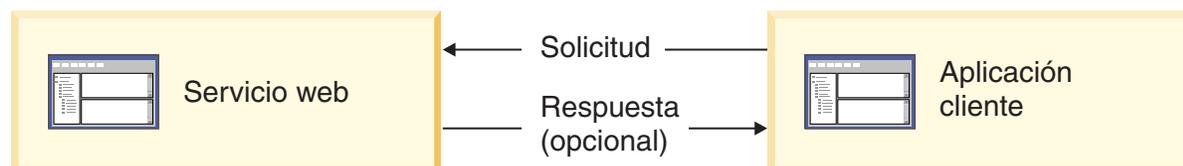


Figura 26. Flujo de una petición del servicio Web a la aplicación cliente.

El servicio Web es una aplicación Web que se ejecuta en el servidor. La exportación se implementa en dicha aplicación Web como un servlet, de forma que el cliente envía su petición a una dirección URL. El servlet pasa la petición a la aplicación SCA en el tiempo de ejecución.

Opcionalmente, la exportación puede enviar datos a la aplicación cliente en respuesta a la petición.

### Cabeceras HTTP:

Los enlaces de importación y exportación HTTP permiten la configuración de cabeceras HTTP y sus valores para ser utilizados para los mensajes de salida. La importación HTTP utiliza estas cabeceras para peticiones y la exportación HTTP las utiliza para las respuestas.

Las cabeceras configuradas estáticamente y la información de control tienen preferencia sobre los valores establecidos de forma dinámica durante la ejecución. Sin embargo, los valores de control del URL de alteración dinámico, la versión y el método alteran temporalmente los valores estáticos, que, de lo contrario, se consideran los valores por omisión.

El enlace soporta la naturaleza dinámica del URL de importación HTTP determinando el valor del URL de destino HTTP, la versión y el método en el tiempo de ejecución. Estos valores se determinan mediante la extracción del valor de referencia de punto final, URL de alteración dinámico, versión y método.

- En el caso de Referencia de punto final, utilice las API `com.ibm.websphere.sca.addressing.EndpointReference` o establezca el campo `/headers/SMOHeader/Target/address` en la cabecera SMO.
- Para el URL de alteración dinámico, la versión y el método, utilice la sección de parámetros de control HTTP del mensaje SCA (Service Component Architecture). Tenga en cuenta que el URL de alteración dinámico tiene prioridad sobre la referencia de punto final de destino; sin embargo, dado que la referencia de punto final se aplica a través de los enlaces es el planteamiento preferido y se deberá utilizar donde sea posible.

La información de control y cabecera para los mensajes de salida bajo los enlaces de importación y exportación HTTP se procesa en el orden siguiente:

1. La información de cabecera y control excluyendo el URL de alteración dinámico de HTTP, la versión y el método del mensaje SCA (prioridad más baja)
2. Modificaciones desde la consola administrativa en el nivel de exportación/importación
3. Modificaciones desde la consola administrativa en el nivel de método de la exportación o la importación
4. Dirección de destino especificada por medio de la referencia de punto final o la cabecera SMO
5. URL de alteración dinámico, versión y método del mensaje SCA
6. La información de cabeceras y control del manejador de datos o del enlace de datos (prioridad mayor)

La exportación e importación HTTP sólo llenará los parámetros de control y las cabeceras de dirección de entrada con los datos procedentes del mensaje entrante (`HTTPExportRequest` y `HTTPImportResponse`) si la propagación de cabecera de protocolo se ha establecido en **True**. Y a la inversa, la exportación e importación HTTP sólo leerá y procesará las cabeceras de salida y los parámetros de control (`HTTPExportResponse` y `HTTPImportRequest`) si la propagación de cabecera de protocolo se ha establecido en **True**.

**Nota:** Los cambios del manejador de datos o del enlace de datos efectuados en las cabeceras, o los parámetros de control, de la respuesta de importación o petición de exportación, no alterarán las instrucciones de proceso del mensaje dentro del enlace de importación o exportación, y deben utilizarse sólo para propagar los valores modificados a los componentes SCA en sentido descendente.

El servicio de contexto es responsable de propagar el contexto (incluidas las cabeceras de protocolo, por ejemplo la cabecera HTTP, y el contexto de usuario, por ejemplo el ID de cuenta) en una vía de acceso de invocación SCA. Durante el desarrollo en IBM Integration Designer, puede controlar la propagación de contexto mediante las propiedades de importación y exportación. Para obtener más detalles, consulte la información sobre enlaces de importación y exportación en el Information Center de IBM Integration Designer.

### Estructuras de cabecera HTTP proporcionadas y soporte

Tabla 22 en la página 83 detalla los parámetros de petición/respuesta para las peticiones y respuestas de importación HTTP y exportación HTTP.

Tabla 22. Información de cabecera HTTP proporcionada

Nombre de control	Petición de importación HTTP	Respuesta de importación HTTP	Petición de exportación HTTP	Respuesta de exportación HTTP
URL	Se omite	No establecida	Se lee en el mensaje de petición <b>Nota:</b> La serie de consulta también forma parte del parámetro de control URL.	Se omite
Versión (valores posibles: 1.0, 1.1; el valor por omisión es 1.1)	Se omite	No establecida	Se lee en el mensaje de petición	Se omite
Método	Se omite	No establecida	Se lee en el mensaje de petición	Se omite
URL de alteración dinámico	Si se establece en el manejador de datos o en el enlace de datos, altera temporalmente el URL de importación HTTP. Se escribe en el mensaje en la línea de petición. <b>Nota:</b> La serie de consulta también forma parte del parámetro de control URL.	No establecida	No establecida	Se omite
Versión de alteración dinámica	Si se establece, altera temporalmente la versión de importación HTTP. Se escribe en el mensaje en la línea de petición.	No establecida	No establecida	Se omite
Método de alteración dinámico	Si se establece, altera temporalmente el método de importación HTTP. Se escribe en el mensaje en la línea de petición.	No establecida	No establecida	Se omite
Tipo de soporte (este parámetro de control lleva parte del valor de la cabecera HTTP Content-Type).	Si está presente, se escribe en el mensaje como parte de la cabecera Content-Type. <b>Nota:</b> Este valor del elemento de control debe ser proporcionado por el manejador de datos o el enlace de datos.	Se lee en el mensaje de respuesta, cabecera Content-Type	Se lee en el mensaje de petición, cabecera Content-Type	Si está presente, se escribe en el mensaje como parte de la cabecera Content-Type. <b>Nota:</b> Este valor del elemento de control debe ser proporcionado por el manejador de datos o el enlace de datos.

Tabla 22. Información de cabecera HTTP proporcionada (continuación)

Nombre de control	Petición de importación HTTP	Respuesta de importación HTTP	Petición de exportación HTTP	Respuesta de exportación HTTP
Conjunto de caracteres (el valor por omisión es UTF-8)	Si está presente, se escribe en el mensaje como parte de la cabecera Content-Type. <b>Nota:</b> Este valor del elemento de control debe ser proporcionado por el enlace de datos.	Se lee en el mensaje de respuesta, cabecera Content-Type	Se lee en el mensaje de petición, cabecera Content-Type	Soportada; se escribe en el mensaje como parte de la cabecera Content-Type. <b>Nota:</b> Este valor del elemento de control debe ser proporcionado por el enlace de datos.
Codificación de transferencia (valores posibles: chunked, identity; el valor por omisión es identity)	Si está presente, se escribe en el mensaje como parte de una cabecera y controla cómo se codifica la transformación del mensaje.	Se lee en el mensaje de respuesta	Se lee en el mensaje de petición	Si está presente, se escribe en el mensaje como parte de una cabecera y controla cómo se codifica la transformación del mensaje.
Codificación de contenido (valores posibles: gzip, x-gzip, deflate, identity; el valor por omisión es identity)	Si está presente, se escribe en el mensaje como una cabecera y controla cómo se codifica la carga útil.	Se lee en el mensaje de respuesta	Se lee en el mensaje de petición	Si está presente, se escribe en el mensaje como una cabecera y controla cómo se codifica la carga útil.
Content-Length	Se omite	Se lee en el mensaje de respuesta	Se lee en el mensaje de petición	Se omite
StatusCode (valor por omisión: 200)	No soportado	Se lee en el mensaje de respuesta	No soportado	Si está presente, se escribe en el mensaje en la línea de respuesta
ReasonPhrase (valor por omisión: OK)	No soportado	Se lee en el mensaje de respuesta	No soportado	Se ignora el valor de control. El valor de la línea de respuesta del mensaje se genera a partir de StatusCode.
Autenticación (contiene varias propiedades)	Si está presente, se utiliza para construir la cabecera de autenticación básica. <b>Nota:</b> El valor de esta cabecera se codificará sólo en el protocolo HTTP. En la SCA, se descodificará y se pasará como texto legible.	No aplicable	Se lee en la cabecera de autenticación básica del mensaje de petición. La presencia de esta cabecera no indica que el usuario se haya autenticado. La autenticación se debe controlar en la configuración del servlet. <b>Nota:</b> El valor de esta cabecera se codificará sólo en el protocolo HTTP. En la SCA, se descodificará y se pasará como texto legible.	No aplicable

Tabla 22. Información de cabecera HTTP proporcionada (continuación)

Nombre de control	Petición de importación HTTP	Respuesta de importación HTTP	Petición de exportación HTTP	Respuesta de exportación HTTP
Proxy (contiene varias propiedades: Host, Port, Authentication)	Si está presente, se utiliza para establecer una conexión a través del proxy.	No aplicable	No aplicable	No aplicable
SSL (contiene varias propiedades: Keystore, Keystore Password, Trustore, Trustore Password, ClientAuth)	Si se ha rellenado y el URL de destino es HTTPS, se utiliza para establecer una conexión a través de SSL.	No aplicable	No aplicable	No aplicable

### Enlaces de datos HTTP:

Para cada correlación diferente de datos entre un mensaje de la arquitectura SCA (Arquitectura de componente de servicio) y un mensaje del protocolo HTTP, debe configurarse un manejador de datos o un enlace de datos HTTP. Los manejadores de datos proporcionan una interfaz de enlaces neutros que permite volver a utilizar los enlaces de transporte y representar el enfoque recomendado. Los enlaces de datos son específicos de un enlace de transporte concreto. Se proporcionan clases de enlaces de datos específicos de HTTP; también pueden escribir manejadores de datos o enlaces de datos personalizados.

**Nota:** Las tres clases de enlaces de datos HTTP descritas en este tema (HTTPStreamDataBindingSOAP, HTTPStreamDataBindingXML y HTTPServiceGatewayDataBinding) están en desuso desde IBM Business Process Manager Versión 7.0. En lugar de utilizar los enlaces de datos descritos en este tema, considere los siguientes manejadores de datos:

- Utilice SOAPDataHandler en lugar de HTTPStreamDataBindingSOAP.
- Utilice UTF8XMLDataHandler en lugar de HTTPStreamDataBindingXML
- Utilice GatewayTextDataHandler en lugar de HTTPServiceGatewayDataBinding

Los enlaces de datos se proporciona para ser utilizados con las importaciones y exportaciones HTTP: enlace de datos binarios, enlace de datos XML y enlace de datos SOAP. En las operaciones unidireccionales no se necesitan enlaces de datos de respuesta. Un enlace de datos se representa mediante el nombre de una clase Java cuyas instancias pueden convertirse de HTTP a ServiceDataObject y viceversa. Se debe utilizar un selector de función en una exportación que, junto con los enlaces de método, puede determina qué enlace de datos se utiliza y qué operación se invoca. Los enlaces de datos proporcionados son:

- Enlaces de datos binarios, los cuales tratan el cuerpo como datos binarios sin estructura. La implementación del esquema XSD del enlace de datos binarios es la siguiente:

```
<xsd:schema elementFormDefault="qualified"
  targetNamespace="http://com.ibm.websphere.http.data.bindings/schema"
  xmlns:tns="http://com.ibm.websphere.http.data.bindings/schema"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:complexType name="HTTPBaseBody">
    <xsd:sequence/>
  </xsd:complexType>

  <xsd:complexType name="HTTPBytesBody">
    <xsd:complexContent>
      <xsd:extension base="tns:HTTPBaseBody">
        <xsd:sequence>
          <xsd:element name="value" type="xsd:hexBinary"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

```
        </xsd:sequence>
    </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
```

- Enlaces de datos XML, los cuales soportan el cuerpo como datos XML. La implementación del enlace de datos XML es similar al enlace de datos JMS XML y no tiene limitaciones en el esquema de la interfaz.
- Enlaces de datos SOAP, los cuales soportan el cuerpo como datos SOAP. La implementación del enlace de datos SOAP no tiene limitaciones en el esquema de la interfaz.

### Implementación de enlaces de datos HTTP personalizado

Esta sección describe cómo implementar un enlace de datos HTTP personalizado.

**Nota:** El enfoque recomendado es la implementación de un manejador de datos personalizado, ya que puede volver a utilizarse con enlaces de transporte.

HTTPStreamDataBinding es la interfaz principal para manejar los mensajes HTTP personalizados. La interfaz se ha diseñado para permitir el manejo de grandes cargas útiles. Sin embargo, para que dicha implementación funcione, este enlace de datos debe devolver la información de control y las cabeceras antes de escribir el mensaje en la corriente de datos.

Los métodos y su orden de ejecución, listado a continuación, deben ser implementados por el enlace de datos personalizado.

Para personalizar un enlace de datos, escriba una clase que implemente HTTPStreamDataBinding. El enlace de datos debe tener cuatro propiedades privadas:

- private DataObject pDataObject
- private HTTPControl pCtrl
- private HTTPHeaders pHeaders
- private yourNativeDataType nativeData

El enlace HTTP invoca el enlace de datos personalizado en el orden siguiente:

- Proceso saliente (DataObject a formato nativo):
  1. setDataObject(...)
  2. setHeaders(...)
  3. setControlParameters(...)
  4. setBusinessException(...)
  5. convertToNativeData()
  6. getControlParameters()
  7. getHeaders()
  8. write(...)
- Proceso entrante (formato nativo a DataObject):
  1. setControlParameters(...)
  2. setHeaders(...)
  3. convertFromNativeData(...)
  4. isBusinessException()
  5. getDataObject()
  6. getControlParameters()
  7. getHeaders()

Debe invocar setDataObject(...) en convertFromNativeData(...) para establecer el valor de dataObject, el cual pasa de ser dato nativo a propiedad privada de "pDataObject".

```
public void setDataObject(DataObject dataObject)
    throws DataBindingException {
    pDataObject = dataObject;
}

public void setControlParameters(HTTPControl arg0) {
    this.pCtrl = arg0;
}

public void setHeaders(HTTPHeaders arg0) {
    this.pHeaders = arg0;
}

/*
 * Add http header "IsBusinessException" in pHeaders.
 * Two steps:
 * 1.Remove all the header with name IsBusinessException (case-insensitive) first.
 * This is to make sure only one header is present.
 * 2.Add the new header "IsBusinessException"
 */
public void setBusinessException(boolean isBusinessException) {
    //remove all the header with name IsBusinessException (case-insensitive) first.
    //This is to make sure only one header is present.
    //add the new header "IsBusinessException", code example:
    HTTPHeader header=HeadersFactory.eINSTANCE.createHTTPHeader();
    header.setName("IsBusinessException");
    header.setValue(Boolean.toString(isBusinessException));
    this.pHeaders.getHeader().add(header);
}

public HTTPControl getControlParameters() {
    return pCtrl;
}

public HTTPHeaders getHeaders() {
    return pHeaders;
}

public DataObject getDataObject() throws DataBindingException {
    return pDataObject;
}

/*
 * Get header "IsBusinessException" from pHeaders, return its boolean value
 */
public boolean isBusinessException() {
    String headerValue = getHeaderValue(pHeaders,"IsBusinessException");
    boolean result=Boolean.parseBoolean(headerValue);
    return result;
}

public void convertToNativeData() throws DataBindingException {
    DataObject dataObject = getDataObject();
    this.nativeData=realConvertWorkFromSDOToNativeData(dataObject);
}

public void convertFromNativeData(HTTPInputStream arg0){
    //Customer-developed method to
    //Read data from HTTPInputStream
    //Convert it to DataObject
    DataObject dataobject=realConvertWorkFromNativeDataToSDO(arg0);
    setDataObject(dataobject);
}

public void write(HTTPOutputStream output) throws IOException {
    if (nativeData != null)
        output.write(nativeData);
}
}
```

## Enlaces EJB

Los enlaces de importación EJB (Enterprise JavaBeans) permiten a los componentes SCA (Service Component Architecture) invocar servicios proporcionados por la lógica empresarial Java EE que se ejecuta en un servidor Java EE. Los enlaces de exportación EJB permite exponer a los componentes SCA como Enterprise JavaBeans, de forma que la lógica empresarial Java EE puede invocar componentes SCA que, de lo contrario, no están disponibles.

### Enlaces de importación EJB:

Los enlaces de importación EJB permiten a un módulo SCA llamar a implementaciones EJB especificando la forma a través de la que se enlaza el módulo que consume con el EJB externo. La importación de servicios de una implementación EJB externa permite a los usuarios conectar su lógica empresarial en el entorno de IBM Business Process Manager y participar en un proceso de negocio.

Utilice Integration Designer para crear enlaces de importación EJB. Puede utilizar cualquiera de estos procedimientos para generar los enlaces:

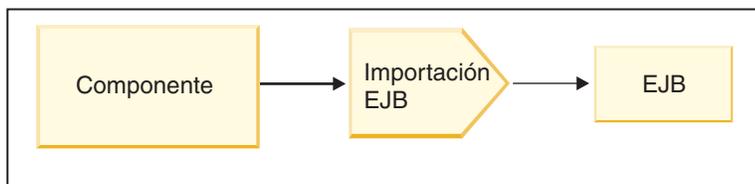
- Crear la importación EJB utilizando el asistente de servicios externos  
Puede utilizar el asistente de servicios externos en Integration Designer para crear una importación EJB basada en una implementación existente. El asistente de servicio externo crea servicios basados en los criterios que proporcione. Genera objetos de negocio, interfaces y archivos de importación basándose en los servicios descubiertos.
- Crear una importación EJB utilizando el editor de ensamblaje.  
Puede crear una importación EJB dentro de un diagrama de ensamblaje utilizando el editor de ensamblaje de Integration Designer. Desde la paleta, puede utilizar una importación o utilizar una clase Java para crear el enlace EJB.

La importación generada tiene enlaces de datos que realizan la conexión Java-WSDL, en lugar de necesitar un componente de puente Java. Puede conectar directamente un componente con una referencia de lenguaje de descripción de servicios web (WSDL) a la importación EJB que se comunica con un servicio basado en EJB utilizando una interfaz Java.

La importación EJB puede interactuar con la lógica empresarial Java EE utilizando el modelo de programación EJB 2.1 o el modelo de programación EJB 3.0.

La invocación de la lógica empresarial Java EE puede ser local (sólo para EJB 3.0) o remota.

- La invocación local se utiliza cuando se desea llamar a la lógica empresarial Java EE que reside en el mismo servidor que la importación.



Servidor A

Figura 27. Invocación local de un EJB (sólo EJB 3.0)

- La invocación remota se utiliza cuando se desea llamar a la lógica empresarial Java EE que no reside en el mismo servidor que la importación.  
Por ejemplo, en la siguiente figura, una importación EJB utiliza la invocación de método remoto sobre protocolo Internet InterORB (RMI/IIOP) para invocar un método EJB en otro servidor.

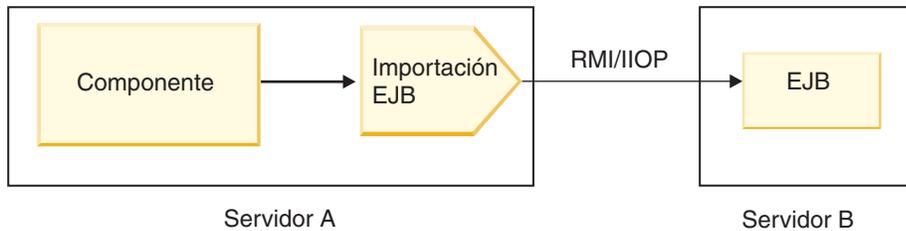


Figura 28. Invocación remota de un EJB

Cuando configura el enlace EJB, Integration Designer utiliza el nombre JNDI para determinar el nivel del modelo de programación EJB y el tipo de invocación (local o remota).

Los enlaces de importación EJB contienen los siguientes componentes principales:

- Manejador de datos JAX-WS
- Selector de error de EJB
- Selector de función de importación EJB

Si el escenario del usuario no se basa en la correlación JAX-WS, es posible que necesite un manejador de datos personalizados, un selector de función y un selector de errores para realizar las tareas que, de lo contrario, completarían los componentes que forman parte de los enlaces de importación EJB. Esto incluye la correlación realizada normalmente por el algoritmo de correlación personalizado.

#### Enlaces de exportación EJB:

Las aplicaciones Java EE externas pueden invocar un componente SCA a través de un enlace de exportación EJB. La utilización de una exportación EJB le permite exponer componentes SCA de forma que las aplicaciones Java EE externas pueden invocar estos componentes utilizando el modelo de programación EJB.

**Nota:** La exportación EJB es un bean sin estado.

Utilice Integration Designer para crear enlaces EJB. Puede utilizar cualquiera de estos procedimientos para generar los enlaces:

- Creación de enlaces de exportación EJB utilizando el asistente de servicios externos  
Puede utilizar el asistente de servicios externos en Integration Designer para crear un servicio de exportación EJB basado en una implementación existente. El asistente de servicio externo crea los servicios en función del criterio que proporcione. A continuación, genera los objetos de negocio, interfaces, y exporta los archivos en función de los servicios descubiertos.
- Creación de enlaces de exportación EJB utilizando el editor de ensamblaje  
Puede crear una exportación EJB utilizando el editor de ensamblaje de Integration Designer.

**Importante:** Un cliente Java 2 Platform, Standard Edition (J2SE) no puede invocar al cliente de exportación EJB que generó en Integration Designer.

Puede generar el enlace a partir de un componente SCA existente, o puede generar una exportación con un enlace EJB para una interfaz Java.

- Cuando genere una exportación para un componente SCA existente que tiene una interfaz WSDL existente, se asigna una interfaz Java a la exportación.
- Cuando genere una exportación para una interfaz Java, puede seleccionar una interfaz WSDL o Java para la exportación.

**Nota:** Una interfaz Java utilizada para crear una exportación EJB tiene las siguientes limitaciones respecto a los objetos (parámetros y excepciones de entrada y salida) pasadas como parámetros en una llamada remota:

- Deben ser de un tipo concreto (en lugar de un tipo abstracto o de interfaz).
- Se deben ajustar a la especificación Enterprise JavaBean. Deben ser serializables y tener el constructor no-argument predeterminado y se debe poder acceder a todas las propiedades mediante los métodos getter y setter.

Consulte en el sitio a Sun Microsystems, Inc., en el sitio Web <http://java.sun.com> la información relacionada con la especificación Enterprise JavaBean.

Asimismo, la excepción debe ser una excepción comprobada, heredada de `java.lang.Exception`, y debe ser singular (esto es, no debe dar soporte a la generación de varios tipos de excepción comprobados).

También debe tener en cuenta que la interfaz empresarial de un EnterpriseBean Java es una interfaz Java sencilla y no debe ampliar `javax.ejb.EJBObject` o `javax.ejb.EJBLocalObject`. Los métodos de la interfaz empresarial no deben generar `java.rmi.Remote.Exception`.

Los enlaces de exportación EJB pueden interactuar con la lógica empresarial Java EE utilizando el modelo de programación EJB 2.1 o el modelo de programación EJB 3.0.

La invocación puede ser local (sólo para EJB 3.0) o remota.

- La invocación local se utiliza cuando la lógica empresarial Java EE llama a un componente SCA que reside en el mismo servidor que la exportación.
- La invocación remota se utiliza cuando la lógica empresarial Java EE no reside en el mismo servidor que la exportación.

Por ejemplo, en la figura siguiente, un EJB utiliza RMI/IIOP para llamar a un componente SCA en un servidor diferente.

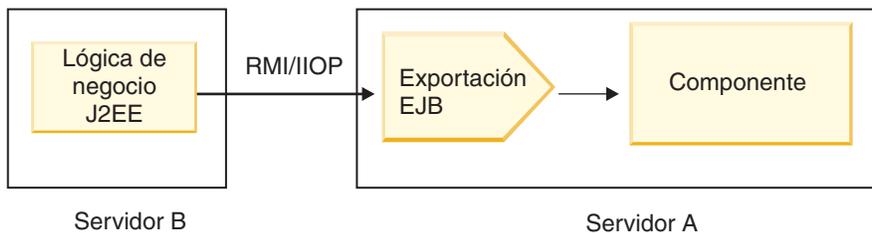


Figura 29. Llamada remota de un cliente a un componente SCA a través de una exportación EJB

Cuando configura el enlace EJB, Integration Designer utiliza el nombre JNDI para determinar el nivel del modelo de programación EJB y el tipo de invocación (local o remota).

Los enlaces de exportación EJB contienen los siguientes componentes principales:

- Manejador de datos JAX-WS
- Selector de función de exportación EJB

Si su escenario de usuario no está basado en la correlación JAX-WS, es posible que necesite un manejador de datos personalizado y un selector de funciones para realizar las tareas que, de otro modo, complementan los componentes que forman parte de los enlaces de exportación EJB. Esto incluye la correlación realizada normalmente por el algoritmo de correlación personalizado.

### Propiedades de enlace EJB:

Los enlaces de importación EJB utilizan los nombres JNDI configurados para determinar el nivel de modelo de programación EJB y el tipo de invocación (local o remota). Los enlaces de importación y

exportación EJB utilizan el manejador de datos JAX-WS para la transformación de datos. El enlace de importación EJB utiliza un selector de función de importación EJB y un selector de errores EJB y el enlace de exportación EJB utiliza un selector de función de exportación EJB.

*Nombres JNDI y enlaces de importación EJB:*

Cuando configura el enlace EJB en una importación, Integration Designer utiliza el nombre JNDI para determinar el nivel de modelo de programación EJB y el tipo de invocación (local o remota).

Si no se ha especificado ningún nombre JNDI, se utiliza el enlace de interfaz EJB predeterminado. Los nombres predeterminados que se han creado dependen de si se invoca un JavaBean de EJB 2.1 o un JavaBean de EJB 3.0.

**Nota:** Consulte el centro de información de WebSphere Application Server si desea información más detallada sobre los convenios de denominación: Visión general de enlaces de aplicación EJB 3.0.

- **JavaBean EJB 2.1**

El nombre JNDI predeterminado seleccionado previamente por Integration Designer es el enlace de EJB 2.1 predeterminado, que adopta el formato **ejb/** más la interfaz de inicio, separados por barras inclinadas.

Por ejemplo, para la interfaz inicial de un JavaBean EJB 2.1 para `com.mycompany.myremotebusinesshome`, el enlace predeterminado es:

```
ejb/com/mycompany/myremotebusinesshome
```

Para EJB 2.1, sólo está soportada la invocación EJB remota.

- **JavaBean EJB 3.0**

El nombre JNDI predeterminado seleccionado previamente por Integration Designer para el JNDI local en el nombre de clase plenamente cualificado de la interfaz local precedido por **ejblocal:**. Por ejemplo, para la interfaz plenamente cualificada de la interfaz local `com.mycompany.mylocalbusiness`, la JNDI EJB 3.0 preseleccionada es:

```
ejblocal:com.mycompany.mylocalbusiness
```

Para la interfaz remota `com.mycompany.myremotebusiness`, la JNDI EJB 3.0 preseleccionada es la interfaz plenamente cualificada:

```
com.mycompany.myremotebusiness
```

Los enlaces de aplicación predeterminados de EJB 3.0 se describen en la siguiente ubicación: Visión general de enlaces de aplicación EJB 3.0.

Integration Designer utilizará el nombre "corto" como la ubicación de JNDI predeterminada para los EJB que utilizan el modelo de programación de versión 3.0.

**Nota:** Si la referencia JNDI desplegada del EJB de destino es diferente de la ubicación de enlace JNDI predeterminado porque se ha utilizado o configurado una correlación personalizada, el nombre JNDI de destino se debe especificar correctamente. Puede especificar el nombre Integration Designer antes del despliegue, o , para el enlace de importación, puede cambiar el nombre en la consola administrativa (después del despliegue) para coincidir con el nombre JNDI del EJB de destino.

Si desea más información sobre cómo crear enlaces EJB, consulte la sección dedicada a Trabajar con enlaces EJB en el centro de información de Integration Designer.

*Manejador de datos JAX-WS:*

El enlace de importación EJB (Enterprise JavaBeans) utiliza el manejador de datos JAX-WS para convertir los objetos de negocio de petición en parámetros de objeto Java y para convertir el valor de retorno de objeto Java en el objeto de negocio de respuesta. El enlace de exportación EJB utiliza el manejador de datos JAX-WS para convertir los EJB de petición en objetos de negocio de petición y para convertir el objeto de negocio de respuesta en un valor de retorno.

Este manejador de datos correlaciona los datos de la interfaz WSDL especificada por SCA con la interfaz Java de EJB de destino (y viceversa) utilizando la especificación de la API de Java para los servicios web de XML (JAX-WS) y la especificación de arquitectura Java para enlaces XML (JAXB).

**Nota:** El soporte actual está limitado a las especificaciones JAX-WS 2.1.1 y JAXB 2.1.3.

El manejador de datos especificado en el nivel de enlace EJB se utiliza para realizar el proceso de peticiones, errores y excepciones de tiempo de ejecución.

**Nota:** Para los errores, se puede establecer un manejador de datos determinado para cada error, especificando la propiedad de configuración `aultBindingType`. Esto altera temporalmente el valor especificado en el nivel del enlace EJB.

El manejador de datos JAX-WS se utiliza, por omisión, cuando el enlace EJB tiene una interfaz WSDL. Este manejador de datos no puede utilizarse para transformar un mensaje SOAP que representa una invocación JAX-WS de un objeto de datos.

El enlace de importación EJB utiliza un manejador de datos para transformar un objeto de datos en una matriz de objetos Java (`Object[]`). Durante las comunicaciones de salida, tiene lugar el siguiente proceso:

1. El enlace EJB establece el tipo esperado, el elemento esperado y el nombre de método de destino en el `BindingContext` para que coincidan con los especificados en el WSDL.
2. El enlace EJB invoca el método de transformación para el objeto de datos que requiere transformación de datos.
3. El manejador de datos devuelve un `Object[]` que representa los parámetros del método (en el orden de su definición en el método).
4. El enlace EJB utiliza el `Object[]` para invocar el método en la interfaz EJB de destino.

El enlace también prepara un `Object[]` para procesar la respuesta de la invocación EJB.

- El primer elemento del `Object[]` es el valor de retorno de la invocación de método Java.
- Los valores siguientes representan los parámetros de entrada para el método.

Esto es necesario para soportar los tipos de entrada/salida y salida de parámetros.

Para los parámetros del tipo salida, los valores se deben devolver en el objeto de datos de respuesta.

El manejador de datos procesa y transforma los valores encontrados en el `Object[]` y, a continuación, devuelve una respuesta al objeto de datos.

El manejador de datos da soporte a `xs:AnyType`, `xs:AnySimpleType` y `xs:Any`, junto con otros tipos de datos XSD. Para habilitar el soporte para `xs:Any`, utilice `@XmlElement (lax=true)` para la propiedad `JavaBean` en el código Java, tal como se muestra en el siguiente ejemplo:

```
public class TestType {
    private Object[] object;

    @XmlElement (lax=true)
    public Object[] getObject() {
        return object;
    }

    public void setObject (Object[] object) {
        this.object=object;
    }
}
```

Esto convierte el objeto de propiedad en TestType en un campo xs:any. El valor de la clase Java utilizado en el campo xs:any debe tener la anotación **@XmlAnyElement**. Por ejemplo, si Dirección es la clase Java que se utiliza para rellenar la matriz de objetos, la clase Dirección debe tener la anotación **@XmlRootElement**.

**Nota:** Para personalizar la correlación del tipo XSD con los tipos Java definidos por la especificación JAX-WS, cambie las anotaciones JAXB para que se adapten a sus necesidades empresariales. El manejador de datos JAX-WS soporta xs:any, xs:anyType y xs:anySimpleType.

Se aplican las siguientes restricciones para el manejador de datos JAX-WS:

- El manejador de datos no incluye el soporte para la anotación **@WebParam** del atributo de cabecera.
- El espacio de nombres para los archivos de esquema de objetos de negocio (archivos XSD) no incluye la correlación predeterminada del nombre de paquete Java. La anotación **@XMLSchema** de package-info.java tampoco funciona. La única forma de crear un XSD con un espacio de nombres es utilizar las anotaciones **@XmlType** y **@XmlRootElement**. **@XmlRootElement** define el espacio de nombres de destino de destino para el elemento global en los tipos JavaBean.
- El asistente de importación EJB no crea los archivos XSD para las clases no relacionadas. La versión 2.0 no soporta la anotación **@XmlSeeAlso**, así pues, si no se hace referencia directamente a la clase hijo desde la clase padre, no se crea un XSD. La solución a este problema es ejecutar SchemaGen para este tipo de clases hijo.

SchemaGen es un programa de utilidad de línea de comandos de línea de mandatos (situado en el directorio *WPS\_Install\_Home/bin*) proporcionado para crear archivos XSD para un bean de Java determinado. Estos XSD se deben copiar manualmente en el módulo para que funcione la solución.

*Selector de errores de EJB:*

Se determina el selector de errores de EJB, si una invocación EJB ha generado un error, una excepción de tiempo de ejecución o una respuesta satisfactoria.

Si se detecta un error, el selector de errores de EJB devuelve el nombre del error nativo al tiempo de ejecución del enlace, de forma que el manejador de datos JAX-WS pueda convertir el objeto de excepción en un objeto de negocio con anomalía.

En una respuesta satisfactoria (sin errores), el enlace de importación EJB ensambla una matriz de objetos Java (Object[]) para devolver los valores.

- El primer elemento del Object[] es el valor de retorno de la invocación de método Java.
- Los valores siguientes representan los parámetros de entrada para el método.

Esto es necesario para soportar los tipos de entrada/salida y salida de parámetros.

En los escenarios de excepción, el enlace ensambla un Object[] y el primer elemento representa la excepción generada por el método.

El selector de errores puede devuelve los valores siguientes:

*Tabla 23. Valores de retorno*

Tipo	Valor de retorno	Descripción
Error	<b>ResponseType.FAULT</b>	Se devuelve cuando el Object[] pasado contiene un objeto de excepción.
Excepción de tiempo de ejecución	<b>ResponseType.RUNTIME</b>	Se devuelve si el objeto de excepción no coincide con ninguno de los tipos de excepción declarados en el método.
Respuesta normal	<b>ResponseType.RESPONSE</b>	Se devuelve en todos los demás casos.

Si el selector de errores devuelve un valor de **ResponseType.FAULT**, se devuelve el nombre de error nativo. El enlace utiliza este nombre de error nativo para determinar el nombre de anomalía WSDL correspondiente del modelo e invocar el manejador de datos de error correcto.

*Selector de función EJB:*

Los enlaces EJB utilizan un selector de función de importación (para el proceso de salida) o un selector de función de exportación (para el proceso de entrada) para determinar el método EJB para llamar.

### **Selector de función de importación**

Para el proceso de salida, el selector de función de importación deriva el tipo de método EJB basado en el nombre de la operación invocada por el componente SCA que está conectado a la importación EJB. El selector de función consulta la anotación `@WebMethod` en la clase Java anotada JAX-WS generada por Integration Designer para determinar el nombre de la operación de destino asociada.

- Si está presente la anotación `@WebMethod`, el selector de función utiliza la anotación `@WebMethod` para determinar la correlación correcta del método Java para el método WSDL.
- Si falta la anotación `@WebMethod`, el selector de función da por supuesto que el nombre del método Java es el mismo que el nombre de la operación invocada.

**Nota:** Este selector de función sólo es válido para una interfaz de tipo WSDL en una importación EJB, no para una interfaz de tipo Java en una importación EJB.

El selector de función devuelve un objeto `java.lang.reflect.Method` que representa el método de la interfaz EJB.

La selector de función utiliza una matriz de objetos Java (`Object[]`) para contener la respuesta del método de destino. El primer elemento del `Object[]` es un método Java con el nombre del WSDL, y el segundo elemento en el `Object[]` es el objeto de negocio de entrada.

### **Selector de función de exportación**

Para el proceso de entrada, el selector de función de exportación deriva el método de destino que se debe invocar desde el método Java.

El selector de función de exportación correlaciona el nombre de la operación Java invocada por el cliente EJB con el nombre de la operación de la interfaz del componente de destino. El nombre de método se devuelve como una serie y lo resuelve el tiempo de ejecución de SCA, en función del tipo de interfaz del componente de destino.

## **Enlaces EIS**

Los enlaces EIS (Sistema de información de empresa) proporcionan conectividad entre los componentes SCA y un EIS externo. Esta comunicación se consigue a través de las exportaciones e importaciones EIS que dan soporte a los adaptadores de recursos JCA 1.5 y a WebSphere Adapters.

Los componentes SCA pueden necesitar la transferencia de datos desde o hacia un EIS externo. Cuando se crea un módulo SCA que requiere este tipo de conectividad, debe incluir (además del componente SCA) una importación o exportación con un enlace EIS para comunicarse con un EIS externo específico.

Los adaptadores de recursos de WebSphere Integration Developer se utilizan en el contexto de una importación o una exportación. Primero desarrolla una importación o una exportación con el asistente de servicio externo y, al desarrollarla, incluye el adaptador de recursos. Una importación EIS, que permite a la aplicación invocar un servicio en un sistema EIS, o una exportación EIS, que permite a una aplicación en un sistema EIS invocar un servicio desarrollado en WebSphere Integration Developer, se crean con un adaptador de recurso particular. Por ejemplo, crearía una importación con el adaptador JD Edwards para invocar un servicio en el sistema JD Edwards.

Cuando utilice el asistente del servicio externo, la información del enlace EIS se creará de forma automática. También puede utilizar otra herramienta, el editor de ensamblaje, para añadir o modificar la información del enlace. Consulte el Centro de información de Integration Designer para obtener más información.

Tras desplegarse en el servidor el módulo SCA que contiene el enlace EIS, podrá utilizar la consola administrativa para ver información sobre el enlace, o para configurar éste último.

### Visión general de los enlaces EIS:

El enlace EIS (sistema de información de empresa), cuando se utiliza con un adaptador de recursos JCA, permite acceder a los servicios en un sistema de información de empresa o hacer que los servicios estén disponibles para el EIS.

En el ejemplo siguiente se muestra un módulo SCA típico con el nombre ContactSyncModule sincroniza la información de contacto entre un sistema Siebel y un sistema SAP.

1. El componente SCA denominado ContactSync escucha (mediante una exportación de aplicación EIS denominada Contacto Siebel) los cambios en los contactos Siebel.
2. El propio componente SCA ContactSync utiliza una aplicación SAP (mediante una importación de aplicación EIS) para actualizar la información de contacto SAP según corresponda.

Como las estructuras de datos utilizadas para almacenar contactos son distintas en los sistemas Siebel y SAP, el componente SCA ContactSync debe proporcionar una correlación.

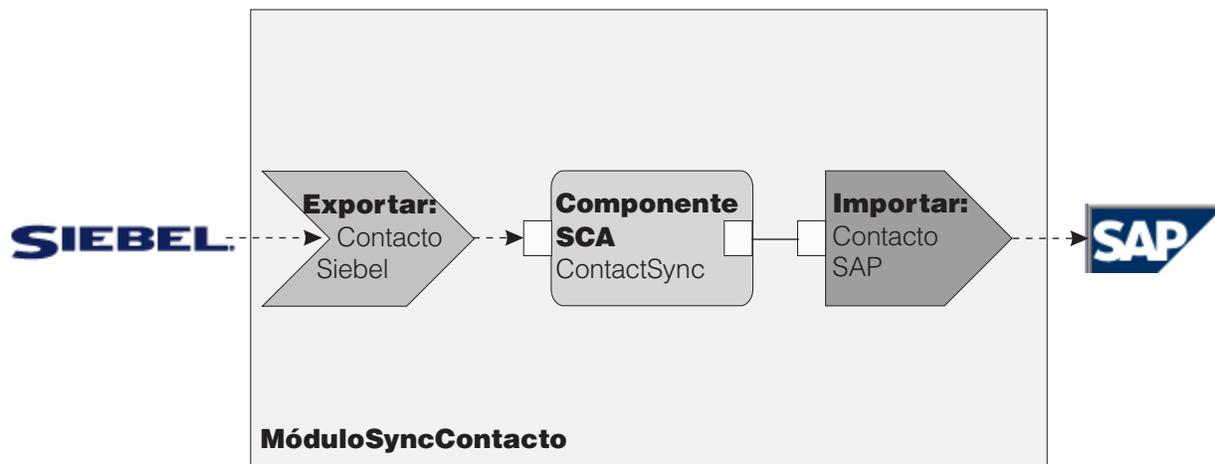


Figura 30. Flujo desde un sistema Siebel a uno SAP

La importación Contacto Siebel y la importación Contacto SAP tienen configurados los adaptadores de recursos adecuados.

### Características clave de los enlaces EIS:

Una importación EIS es una importación SCA (Service Component Architecture) que permite a los componentes del módulo SCA utilizar los servicios EIS definidos fuera del módulo SCA. Una importación EIS se utiliza para transferir datos del componente SCA a un EIS externo; una exportación EIS se utiliza para transferir datos del EIS externo a un componente SCA.

## Importaciones

El rol de la importación EIS es hacer de puente entre los componentes SCA y los sistemas EIS externos. Las aplicaciones externas se pueden tratar como una importación EIS. En este caso, la importación EIS envía datos al EIS externo y, opcionalmente, recibe datos de respuesta.

La importación EIS proporciona a los componentes SCA una vista uniforme de las aplicaciones externas al módulo. Esto permite que los componentes se comuniquen con un EIS externo como, por ejemplo, SAP, Siebel o PeopleSoft, utilizando un modelo SCA coherente.

En el lado del cliente de la importación existe una interfaz, expuesta por la aplicación de importación EIS, con uno o más métodos, cada uno de los cuales toma objetos de datos como argumentos y valores de retorno. En el lado de la implementación, existe una CCI (Common Client Interface) implementada por un adaptador de recursos.

La implementación en tiempo de ejecución de una importación EIS conecta la interfaz del lado del cliente y la CCI. La importación correlaciona la invocación del método en la interfaz de la invocación en la CCI.

Los enlaces se crean a tres niveles: el enlace de interfaz, que utiliza los enlaces de método contenidos, que a su vez utilizan enlaces de datos.

El enlace de interfaz relaciona la interfaz de la importación con la conexión con el sistema EIS que proporciona la aplicación. Esto refleja el hecho de que la instancia específica del EIS proporciona el conjunto de aplicaciones, representado por la interfaz, y la conexión proporciona acceso a esta instancia. El elemento de enlace contiene propiedades con suficiente información para crear la conexión (estas propiedades forman parte de la instancia `javax.resource.spi.ManagedConnectionFactory`).

El enlace de método asocia el método con la interacción específica con el sistema EIS. Para JCA, la interacción está caracterizada por el conjunto de propiedades de la implementación de interfaz `javax.resource.cci.InteractionSpec`. El elemento de interacción del enlace de método contiene estas propiedades, junto con el nombre de la clase, lo que proporciona información suficiente para realizar la interacción. El enlace de método utiliza enlaces de datos que describen la correlación del argumento y el resultado del método de interfaz con la representación EIS.

El escenario de ejecución de una importación EIS es el siguiente:

1. El método en la interfaz de importación se invoca utilizando el modelo de programación SCA.
2. La petición, que alcanza la importación EIS, contiene el nombre del método y los argumentos.
3. La importación crea primero una implementación de enlace de interfaz y, a continuación, utilizando los datos del enlace de importación, una `ConnectionFactory`, y asocia ambas. Es decir, la importación invoca `setConnectionFactory` en el enlace de interfaz.
4. Se crea la implementación de enlace de método que coincide con el método invocado.
5. La instancia `javax.resource.cci.InteractionSpec` se crea y se rellena y, a continuación, se utilizan enlaces de datos para enlazar los argumentos de método con un formato conocido para el adaptador de recursos.
6. La interfaz CCI se utiliza para realizar la interacción.
7. Cuando se devuelve la llamada, se utilizan los enlaces de datos para crear el resultado de la invocación y el resultado se devuelve al llamante.

## Exportaciones

El rol de la exportación EIS es hacer de puente entre un componente SCA y un EIS externo. Las aplicaciones externas se pueden tratar como una exportación EIS. En este caso, la aplicación externa envía

los datos en forma de notificaciones periódicas. Una exportación EIS se puede considerar una aplicación de suscripción que escucha una solicitud externa de un EIS. El componente SCA que utiliza la exportación EIS la ve como una aplicación local.

La exportación EIS proporciona a los componentes SCA una vista uniforme de las aplicaciones externas al módulo. Esto permite que los componentes se comuniquen con un EIS como, por ejemplo, SAP, Siebel o PeopleSoft, utilizando un modelo SCA coherente.

La exportación presenta una implementación de receptor que recibe solicitudes del EIS. El receptor implementa una interfaz receptora específica de adaptador de recursos. La exportación también contiene un componente que implementa la interfaz, expuesto al EIS a través de la exportación.

La implementación en tiempo de ejecución de una exportación EIS conecta el receptor con el componente que implementa la interfaz. La exportación correlaciona la solicitud EIS con la invocación de la operación correspondiente en el componente. Los enlaces se crean a tres niveles: un enlace de receptor, que utiliza el enlace de método nativo contenido, que a su vez utiliza un enlace de datos.

El enlace de receptor relaciona el receptor que recibe las solicitudes con el componente expuesto mediante la exportación. La definición de exportación contiene el nombre del componente; el tiempo de ejecución lo localiza y le envía las solicitudes.

El enlace de método nativo asocia el método nativo o el tipo de suceso recibido por el receptor con la operación implementada por el componente expuesto a través de la exportación. No existe ninguna relación entre el método invocado en el receptor y el tipo de suceso; todos los sucesos llegan a través de uno o varios métodos del receptor. El enlace de método nativo utiliza el selector de funciones definido en la exportación para extraer el nombre de método nativo de los datos de entrada y enlaces de datos para enlazar el formato de datos del EIS con un formato conocido para el componente.

El escenario en tiempo de ejecución de una exportación EIS es el siguiente:

1. La solicitud EIS activa la invocación del método en la implementación del receptor.
2. El receptor localiza e invoca la exportación y le pasa todos los argumentos de invocación.
3. La exportación crea la implementación del enlace de receptor.
4. La exportación crea una instancia del selector de funciones y lo establece en el enlace de receptor.
5. La exportación inicializa enlaces de método nativo y los añade al enlace de receptor. Para cada enlace de método nativo, también se inicializan los enlaces de datos.
6. La exportación invoca el enlace del receptor.
7. El enlace de receptor localiza los componentes exportados y utiliza el selector de funciones para recuperar el nombre de método nativo.
8. Este nombre se utiliza para localizar el enlace de método nativo que, a continuación, invoca el componente de destino.

El estilo de interacción del adaptador permite al enlace de exportación EIS invocar el componente de destino de forma asíncrona (valor por omisión) o síncrona.

### **Adaptadores de recurso**

Primero desarrolla una importación o una exportación con el asistente de servicio externo y, al desarrollarla, incluye un adaptador de recursos. Los adaptadores proporcionados con IBM Integration Designer que se utiliza para acceder a sistemas CICS, IMS, JD Edwards, PeopleSoft, SAP y Siebel están pensados con fines de desarrollo y prueba solamente. Esto significa que puede usarlos para desarrollar y probar sus aplicaciones.

Una vez desplegada la aplicación, necesitará adaptadores de tiempo de ejecución con licencia para ejecutar la aplicación. Sin embargo, cuando cree el servicio, puede incluir el adaptador con su servicio. Es

posible que la licencia de su adaptador le permita usar el adaptador incluido en tiempo de ejecución. Estos adaptadores se ajustan a los estándares de Java EE Connector Architecture (JCA 1.5). JCA es un estándar abierto utilizado en Java EE para la conectividad EIS. JCA incluye una infraestructura gestionada; es decir, el servidor de aplicaciones proporciona Calidad del servicio (QoS), lo que ofrece gestión de ciclo de vida y seguridad a las transacciones. También se ajustan a la especificación Enterprise Metadata Discovery con excepción del adaptador de recursos IBM CICS ECI y del conector IBM IMS para Java.

El asistente también admite los adaptadores de WebSphere Business Integration, un conjunto antiguo de adaptadores.

## Recursos Java EE

El módulo EIS, un módulo SCA que sigue el patrón del módulo EIS, puede desplegarse en la plataforma Java EE.

El despliegue del módulo EIS en la plataforma Java EE genera una aplicación, que está preparada para su ejecución, comprimida como un archivo EAR y desplegada en el servidor. Se crean todos los artefactos y recursos Java EE; se configura la aplicación y está preparada para ejecutarse.

### Propiedades dinámicas de la especificación de conexión y la especificación de interacción JCA:

El enlace EIS puede aceptar entrada para la InteractionSpec y la ConnectionSpec especificadas utilizando un objeto de datos hijo bien definido que acompaña a la carga útil. Esto permite interacciones dinámicas de solicitud-respuesta con un adaptador de recursos a través de la InteractionSpec y la autenticación de componentes a través de la ConnectionSpec.

`javax.cci.InteractionSpec` transporta información sobre cómo se debe manejar la petición de interacción con el adaptador de recursos. También puede incluir información sobre cómo se ha conseguido la interacción después de la petición. Estas comunicaciones de dos direcciones a través de las interacciones se conocen también como *conversaciones*.

El enlace EIS espera que la carga útil que será el argumento en el adaptador de recursos contenga un objeto de datos hijo denominado **properties**. Este objeto de datos de propiedad contendrá pares de nombre/valor, donde el nombre de las propiedades de especificación de interacción tendrá un formato específico. Las normas del formato son las siguientes:

- Los nombres deben empezar con el prefijo **IS**, seguido del nombre de propiedad. Por ejemplo, una especificación de interacción con una propiedad JavaBeans llamada **InteractionId** especificará el nombre de propiedad como **ISInteractionId**.
- El par de nombre/valor representa el nombre y el valor del tipo simple de la propiedad de especificación de interacción.

En este ejemplo, una interfaz especifica que la entrada de una operación es un objeto de datos **Account**. Esta interfaz invoca un servicio de enlace de importación EIS con el objetivo de enviar y recibir una propiedad dinámica InteractionSpec denominada **workingSet** con el valor **xyz**.

El gráfico de negocio o los objetos de negocio del servidor contienen un objeto de negocio **properties** subyacente que permite el envío de datos específicos del protocolo con la carga útil. Este objeto de negocio **properties** está incorporado y no se debe especificar en el esquema XML cuando se construye un objeto de negocio. Sólo se debe crear y utilizar. Si tiene sus propios tipos de datos definidos basados en un esquema XML, deberá especificar un elemento **properties** que contenga los pares de nombre/valor esperados.

```

BOFactory dataFactory = (BOFactory) \
serviceManager.locateService("com/ibm/websphere/bo/BOFactory");
//Envoltura para interfaces de estilo acomodado de literal de documento,
//saltar a la carga útil si no es literal de documento
DataObject docLitWrapper = dataFactory.createElement /
("http://mytest/eis/Account", "AccountWrapper");

```

Cree la carga útil.

```

DataObject account = docLitWrapper.createDataObject(0);
DataObject accountInfo = account.createDataObject("AccountInfo");
//Llevar a cabo la configuración de la carga útil

```

```

//Construir los datos de propiedades para la interacción dinámica

```

```

DataObject properties = account.createDataObject("properties");

```

Para el nombre `workingSet`, establezca el valor esperado (**xyz**).

```

properties.setString("ISworkingSet", "xyz");

```

```

//Invocar el servicio con argumento

```

```

Service accountImport = (Service) \
serviceManager.locateService("AccountOutbound");
DataObject result = accountImport.invoke("createAccount", docLitWrapper);

```

```

//Obtener la propiedad devuelta
DataObject retProperties = result.getDataObject("properties");

```

```

String workingset = retProperties.getString("ISworkingSet");

```

Puede utilizar las propiedades `ConnectionSpec` en la autenticación de componentes dinámica. Se aplican las mismas normas que antes, excepto que el prefijo de nombre de la propiedad debe ser **CS** (en lugar de **IS**). Las propiedades `ConnectionSpec` no son de dos direcciones. El mismo objeto de datos **properties** puede contener propiedades **IS** y **CS**.

Para utilizar las propiedades `ConnectionSpec`, establezca la propiedad **resAuth** especificada en el enlace de importación en **Application**. Asimismo, asegúrese de que el adaptador de recursos dé soporte a la autorización de componentes. Consulte el capítulo 8 de *J2EE Connector Architecture Specification* para obtener más información.

### Clientes externos con enlaces EIS:

El servidor puede enviar mensajes o recibir mensajes de clientes externos utilizando enlaces EIS.

Un cliente externo, por ejemplo, un portal Web o un EIS, necesita enviar un mensaje a un módulo SCA en el servidor o necesita que lo invoque un componente desde el servidor.

El cliente invoca la importación EIS del mismo modo que invoca cualquier otra aplicación, utilizando la interfaz DII (Dynamic Invocation Interface) o la interfaz Java.

1. El cliente externo crea una instancia del `ServiceManager` y busca la importación EIS utilizando su nombre de referencia. El resultado de la búsqueda es una implementación de la interfaz de servicio.
2. El cliente crea un argumento de entrada, un objeto de datos genérico, que se crea dinámicamente utilizando el esquema de objeto de datos. Este paso se realiza utilizando la implementación de la interfaz `DataFactory` del objeto de datos de servicio.
3. El cliente externo invoca el EIS y obtiene los resultados necesarios.

De forma alternativa, el cliente puede invocar la importación EIS utilizando la interfaz Java.

1. El cliente crea una instancia del ServiceManager y busca la importación EIS utilizando su nombre de referencia. El resultado de la búsqueda es una interfaz Java de la importación del EIS.
2. El cliente crea un argumento de entrada y un objeto de datos escrito.
3. El cliente invoca el EIS y obtiene los resultados necesarios.

La interfaz de exportación EIS define la interfaz del componente SCA exportado que está disponible para los servicios EIS externos. Esta interfaz se puede considerar la interfaz que invocará una aplicación externa (por ejemplo, SAP o PeopleSoft) a través de la implementación del tiempo de ejecución del servicio de exportación EIS.

La exportación utiliza EISExportBinding para enlazar los servicios exportados a la aplicación EIS externa. Permite suscribirse a una aplicación contenida en el módulo SCA para que escuche las solicitudes de servicio EIS. El enlace de exportación de EIS especifica la correlación entre la definición de sucesos de entrada, tal como la entiende el adaptador de recursos (utilizando las interfaces Java EE Connector Architecture) y la invocación de las operaciones SCA.

EISExportBinding requiere que los servicios EIS externos se basen en contratos de entrada JCA (Java EE Connector Architecture 1.5). EISExportBinding requiere que se especifique un manejador de datos o un enlace de datos a nivel de enlace o a nivel de método.

## Enlaces JMS

Un proveedor de JMS (Java Message Service) permite la mensajería basada en la API y el modelo de programación JMS (Java Messaging Service). Proporciona fábricas de conexiones Java EE para crear conexiones para destinos JMS y para enviar y recibir mensajes.

Se proporcionan estos enlaces JMS:

- Enlace de proveedor SIB (Service Integration Bus) compatible con JMS JCA 1.5 (*enlace JMS*).
- Enlaces JMS genéricos no de JCA compatibles con JMS 1.1 (*enlace JMS genérico*).
- Enlace JMS de WebSphere MQ, que proporciona soporte de proveedor JMS para WebSphere MQ y permite la interoperatividad de la aplicación Java EE (*enlace JMS de WebSphere MQ*)

Los enlaces de exportación e importación JMS permiten que un módulo SCA (Service Component Architecture) realice llamadas y reciba mensajes de sistemas JMS externos.

También están soportados los enlaces de WebSphere MQ (*enlace de WebSphere MQ*) que permiten a los usuarios MQ nativos manejar los mensajes arbitrarios de mensajes de entrada y de salida (WebSphere MQ obligatorio).

Los enlaces de importación y exportación JMS proporcionan integración con las aplicaciones JMS que utilizan el proveedor JMS SIB basado en JCA 1.5, que forma parte de WebSphere Application Server. Los demás adaptadores de recursos JMS basados en JCA 1.5 no están soportados.

## Visión general de enlaces JMS:

Los enlaces JMS proporcionan conectividad entre el entorno SCA (Service Component Architecture) y los sistemas JMS.

## Enlaces JMS

Los principales componentes de los enlaces de importación JMS y exportación JMS son:

- Adaptador de recursos: permite la conectividad gestionada y bidireccional entre un módulo SCA y sistemas JMS externos.
- Conexiones: encapsulan una conexión virtual entre un cliente y una aplicación de proveedor.
- Destinos: utilizados por un cliente para especificar el destino de los mensajes que produce o el origen de los mensajes que consume.

- Datos de autenticación: utilizados para proteger el acceso al enlace.

### Enlaces de importación JMS

Los enlaces de importación JMS permiten importar una aplicación JMS externa para utilizarla dentro del módulo SCA. Los enlaces de importación JMS permiten a los componentes del módulo SCA comunicarse con los servicios proporcionados por las aplicaciones JMS externas.

Las conexiones del proveedor JMS asociado de los destinos JMS se crean mediante una fábrica de conexiones JMS. Utilice los objetos administrativos de la fábrica de conexiones para gestionar fábricas de conexiones de JMS para el proveedor de mensajería por omisión.

La interacción con los sistemas JMS externos incluye la utilización de destinos para enviar peticiones y recibir respuestas.

Se da soporte a dos tipos de escenarios de uso para los enlaces de importación JMS, dependiendo del tipo de operación que se invoque:

- Unidireccional: la importación JMS coloca un mensaje en el destino de envío configurado en el enlace de importación. No se envía nada al campo replyTo de la cabecera JMS.
- Bidireccional (petición y respuesta): la importación JMS coloca un mensaje en el destino de envío y persiste la respuesta que recibe del componente SCA.

El enlace de importación se puede configurar (utilizando el campo **Esquema de correlación de respuesta** en Integration Designer) para esperar el ID de correlación de mensajes de respuesta que se ha copiado del ID de mensaje de petición (el predeterminado), o desde el ID de correlación de mensajes de petición. El enlace de importación también se puede configurar para utilizar un destino de respuesta dinámico temporal para correlacionar respuestas con peticiones. Se crea un destino temporal para cada petición y la importación utiliza este destino para recibir la respuesta.

El destino indicado en receive se establece en la propiedad de la cabecera replyTo del mensaje de salida. Se despliega un receptor de mensajes para escuchar en el destino de recepción y cuando se recibe una respuesta, el receptor de mensajes vuelve a pasar la respuesta al componente.

Para los escenarios unidireccionales y bidireccionales, se pueden especificar propiedades dinámicas y estáticas. Las propiedades estáticas se pueden establecer desde el enlace de método de importación JMS. Algunas de estas propiedades tienen significados especiales para el tiempo de ejecución JMS de SCA.

Resulta importante tener en cuenta que JMS es un enlace asíncrono. Si un componente llamante invoca una importación JMS de forma síncrona (para una operación bidireccional), el componente llamante se bloquea hasta que el servicio JMS devuelve la respuesta.

Figura 31 en la página 102 ilustra cómo se enlaza la importación con el servicio externo.

## Importación JMS

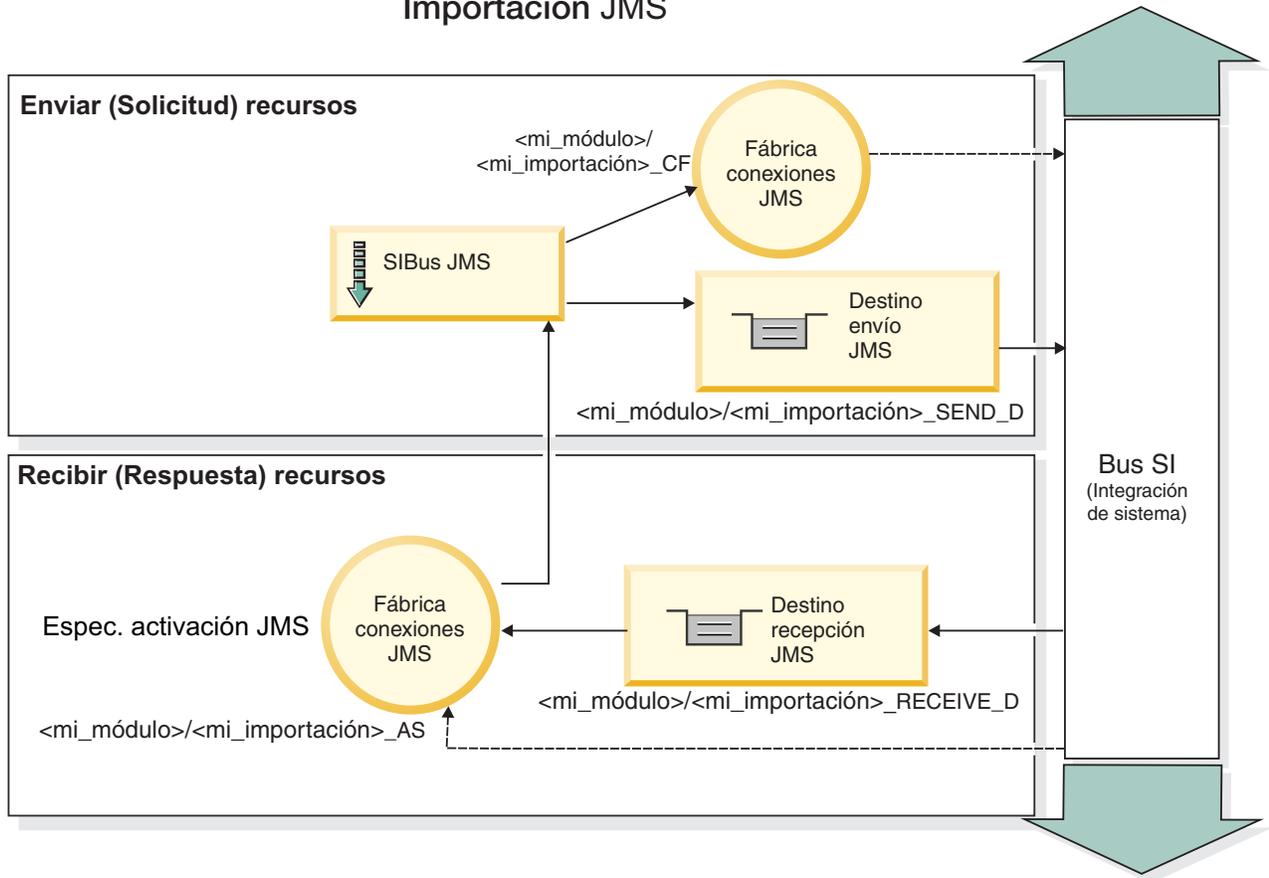


Figura 31. Recursos de enlace de importación JMS

### Enlaces de exportación JMS

Los enlaces de exportación JMS proporcionan el medio mediante el cual los módulos SCA pueden ofrecer servicios a aplicaciones JMS externas.

La conexión que forma parte de una exportación JMS es una especificación de activación configurable.

Una exportación JMS ha enviado y recibido destinos.

- El destino indicado en receive es el lugar donde debe colocarse el mensaje de entrada del componente de destino.
- El destino indicado en send es el lugar al que se enviará la respuesta, a menos que el mensaje de entrada lo altere temporalmente mediante la propiedad de cabecera replyTo.

Se despliega un receptor de mensajes para escuchar las peticiones de entrada del destino receive especificado en el enlace de exportación. El destino especificado en el campo send se utiliza para enviar la respuesta a la petición de entrada si el componente invocado proporciona una respuesta. El destino especificado en el campo replyTo del mensaje de entrada altera temporalmente el destino especificado en el campo send.

Figura 32 en la página 103 ilustra cómo se enlaza el solicitante externo con la exportación.

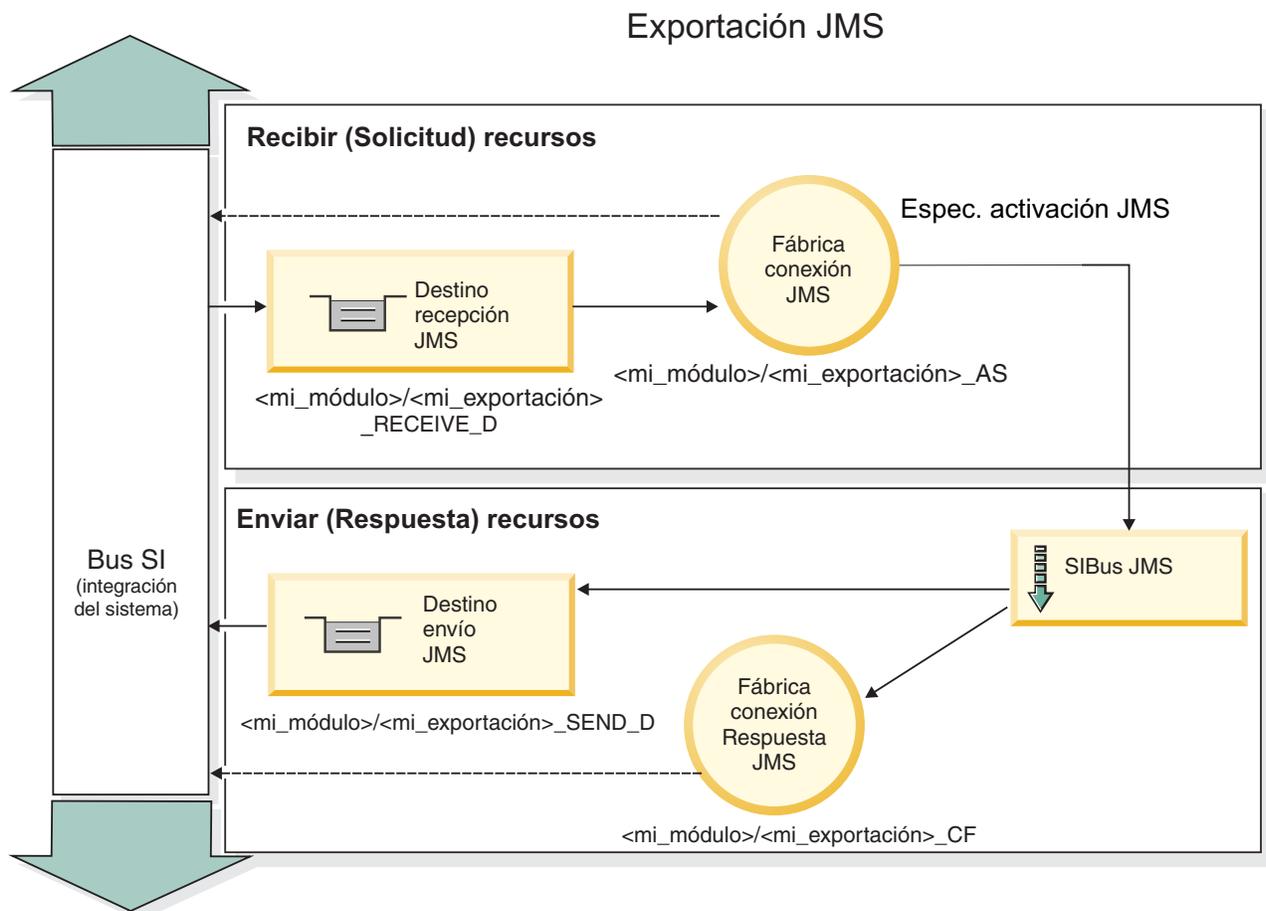


Figura 32. Recursos de enlace de exportación JMS

#### Integración JMS y adaptadores de recursos:

El servicio JMS (Java Message Service) proporciona integración mediante un adaptador de recursos basado en JMS JCA 1.5 disponible. Se proporciona soporte completo para la integración JMS para el adaptador de recursos JMS SIB (Service Integration Bus).

Utilice un proveedor de JMS para el adaptador de recursos JCA 1.5 cuando desee integrarse con un sistema JMX externo compatible con JCA 1.5. Los servicios externos compatibles con JCA 1.5 pueden recibir y enviar mensajes para la integración con los componentes SCA (Service Component Architecture) utilizando el adaptador de recursos JMS SIB.

El uso de adaptadores de recursos JCA 1.5 específicos de otros proveedores no está soportado.

Los módulos JMS no se pueden desplegar en un entorno Java SE. Dichos módulos sólo se despliegan en un entorno Java EE.

#### Características clave de los enlaces JMS:

Las características claves de los enlaces de importación y exportación JMS incluyen cabeceras y recursos Java EE creados.

## Cabeceras especiales

Se utilizan las propiedades de cabecera especiales en las importaciones y exportaciones JMS para indicar al destino cómo debe manejar el mensaje.

Por ejemplo, TargetFunctionName se correlaciona desde el método nativo al método de operación.

## Recursos Java EE

Se crean varios recursos Java EE cuando se despliegan las importaciones y exportaciones JMS en un entorno Java EE.

### ConnectionFactory

Utilizado por los clientes para crear una conexión con el proveedor de JMS.

### ActivationSpec

Las importaciones la utilizan para recibir las respuestas a una petición. Las exportaciones la utilizan cuando configuran los puntos finales de mensaje que representan los receptores de mensajes en sus interacciones con el sistema de mensajería.

## Destinos

- Destino de envío: en una importación, es el lugar donde se envía la solicitud o el mensaje saliente; en una exportación, es el lugar donde se enviará el mensaje de respuesta, si no se reemplaza por el campo de cabecera JMSReplyTo en el mensaje de entrada.
- Destino de recepción: donde se debe colocar el mensaje de entrada; con las importaciones, se trata de una respuesta; con las exportaciones, se trata de una solicitud.
- Destino de devolución de llamada: el destino del sistema JMS SCA utilizado para almacenar la información de correlación. Ni lea ni escriba en este destino.

La tarea de instalación crea la ConnectionFactory y tres destinos. Además, crea la ActivationSpec para permitir que el receptor de mensajes en tiempo de ejecución escuche las respuestas en el destino de recepción. Las propiedades de estos recursos se especifican en el archivo de exportación o de importación.

## Cabeceras JMS:

Un mensaje JMS contiene dos tipos de cabeceras: la cabecera del sistema JMS y varias propiedades JMS. Se puede acceder a ambos tipos de cabeceras, bien en un módulo de mediación del objeto de mensajes de servicios (SMO), o bien mediante la API ContextService.

## Cabecera del sistema JMS

La cabecera del sistema JMS se representa en el SMO mediante el elemento JMSHeader, que contiene todos los campos que se encuentran, normalmente, en una cabecera JMS. Aunque éstos pueden modificarse en la mediación (o en ContextService), algunos campos de cabecera del sistema JMS establecidos en el SMO no se propagarán en el mensaje JMS de salida, ya que el sistema, o los valores estáticos, los alteran temporalmente.

Los campos clave de la cabecera del sistema JMS que se pueden actualizar en la mediación (o en ContextService) son:

- **JMSType** y **JMSCorrelationID**: valores de las propiedades de la cabecera del mensaje predefinidas específicas.
- **JMSDeliveryMode**: valores de la modalidad de entrega (persistentes o no persistentes). El valor por omisión es persistente).
- **JMSPriority**: valor de prioridad (0 a 9). El valor por omisión es JMS\_Default\_Priority).

## Propiedades JMS

Las propiedades JMS se representan en el SMO como entradas de la lista Propiedades. Las propiedades se pueden añadir, actualizar o suprimir dentro de una mediación, o bien utilizando la API ContextService.

Las propiedades también pueden establecerse estáticamente en el enlace JMS. Las propiedades que se establecen estáticamente alteran temporalmente los valores (que tengan el mismo nombre) que se establecen dinámicamente.

Las propiedades de usuario propagadas desde otros enlaces (por ejemplo, un enlace HTTP) generarán una salida en el enlace JMS en forma de propiedades JMS.

## Valores de propagación de cabecera

La propagación de las propiedades y la cabecera del sistema JMS bien sea a desde el mensaje JMS de entrada a los componentes ubicados en sentido descendente, o bien sea desde los componentes ubicados en sentido ascendente al mensaje JMS de salida, se pueden controlar mediante el distintivo Propagate Protocol Header del enlace.

Cuando se establece Propagate Protocol Header, la información de cabecera puede fluir al mensaje o al componente de destino, tal como se describe en la lista siguiente:

- **Petición de exportación JMS**  
La cabecera JMS recibida en el mensaje se propagará a los componentes de destino por medio del servicio de contexto. Las propiedades JMS recibidas en el mensaje se propagará a los componentes de destino por medio del servicio de contexto.
- **Respuesta de exportación JMS**  
Cualquiera de los campos de cabecera JMS establecidos en el servicio de contexto se utilizará en el mensaje de salida, si no lo han alterado temporalmente las propiedades establecidas en el enlace de exportación de JMS. Cualquiera de las propiedades establecidas en el servicio de contexto se utilizará en el mensaje de salida, si no la han alterado temporalmente las propiedades establecidas en el enlace de exportación de JMS.
- **Petición de importación JMS**  
Cualquiera de los campos de cabecera JMS establecidos en el servicio de contexto se utilizará en el mensaje de salida, si no lo han alterado temporalmente las propiedades establecidas en el enlace de importación de JMS. Cualquiera de las propiedades establecidas en el servicio de contexto se utilizará en el mensaje de salida, si no la han alterado temporalmente las propiedades establecidas en el enlace de importación de JMS.
- **Respuesta de importación JMS**  
La cabecera JMS recibida en el mensaje se propagará a los componentes de destino por medio del servicio de contexto. Las propiedades JMS recibidas en el mensaje se propagará a los componentes de destino por medio del servicio de contexto.

## Esquema de correlación de destinos de respuesta dinámicos temporales de JMS:

El esquema de correlación de destinos de respuesta dinámicos temporales provoca que se cree una cola o tema dinámica exclusiva para cada petición enviada.

El destino de respuesta estático especificado en la importación se utiliza para derivar la naturaleza de la cola o el tema de destino dinámico temporal. Esto se establece en el campo **ReplyTo** de la petición y la importación JMS escucha respuestas en dicho destino. Cuando se recibe la respuesta, se vuelve a poner en la cola en el destino de respuesta estático para el proceso asíncrono. El campo **CorrelationID** de la respuesta no se utiliza y no se debe establecer.

## Problemas transaccionales

Cuando se utiliza un destino dinámico temporal, la respuesta se debe consumir en la misma hebra que la respuesta enviada. La petición se debe enviar fuera de la transacción global y se debe confirmar antes de ser recibida por el servicio de programa de fondo y se devuelve una respuesta.

## Persistencia

Las colas dinámicas temporales son entidades de vida corta y no garantizan el mismo nivel de persistencia asociada a una cola o un tema estático. Una cola o un tema dinámico temporal no sobrevivirá un reinicio de servidor y no generará mensajes. Después de que el mensaje se haya vuelto a poner en la cola del destino de respuesta estático, conserva la persistencia definida en el mensaje.

## Tiempo de espera

La importación espera a recibir la respuesta en el destino de respuesta dinámico temporal para una cantidad de tiempo fijo. Este intervalo de tiempo se tomará del calificador de tiempo Caducidad de respuesta SCA, si está establecido, de lo contrario, el tiempo toma el valor predeterminado de 60 segundos. Si se excede el tiempo de espera, la importación lanza una excepción `ServiceTimeoutRuntimeException`.

## Clientes externos:

El servidor puede enviar mensajes o recibir mensajes de clientes externos utilizando enlaces JMS.

Un cliente externo (como un portal Web o un sistema de empresa) puede enviar un mensaje a un módulo SCA del servidor o lo puede invocar un componente desde el servidor.

Los componentes de exportación JMS despliegan los receptores de mensajes para escuchar las peticiones que entran en el destino de recepción especificado en el enlace de exportación. El destino especificado en el campo de envío se utiliza para enviar la respuesta a la petición de entrada si la aplicación invocada proporciona una respuesta. De esta forma, un cliente externo puede invocar aplicaciones con el enlace de exportación.

Las importaciones JMS se enlazan y pueden entregar mensajes a los clientes externos. Este mensaje puede necesitar o no una respuesta del cliente externo.

### *Trabajo con clientes externos:*

Es posible que un cliente externo (es decir, un cliente que está fuera del servidor) necesite interactuar con una aplicación instalada en el servidor.

Considere un escenario muy sencillo en el cual un cliente externo desea interactuar con una aplicación en el servidor. En la figura se describe un escenario típico sencillo.

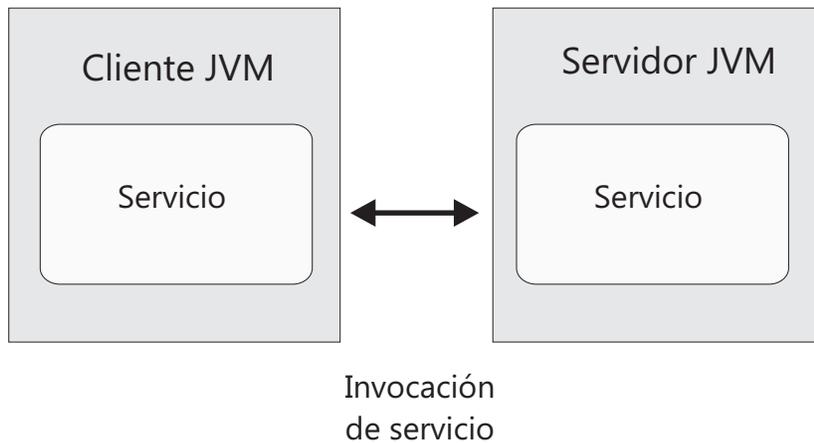


Figura 33. Escenario de caso de uso sencillo: el cliente externo interactúa con la aplicación de servidor

La aplicación SCA incluye una exportación con un enlace JMS; que hace que la aplicación esté disponible para los clientes externos.

Cuando tenga un cliente externo en una máquina virtual Java (JVM) independiente del servidor, debe seguir varios pasos para crear una conexión e interactuar con una exportación JMS. El cliente obtiene un InitialContext con los valores correctos y busca los recursos a través de JNDI. A continuación, el cliente utiliza la especificación JMS 1.1 para acceder a los destinos y enviar y recibir mensajes en los destinos.

Los nombres JNDI por omisión de los recursos creados automáticamente por el tiempo de ejecución aparecen en el tema de configuración de esta sección. No obstante, si tiene recursos creados previamente, utilice esos nombres JNDI.

1. Configure los destinos JMS y la fábrica de conexiones para enviar el mensaje.
2. Asegúrese de que el contexto JNDI, el puerto para el adaptador de recursos SIB y el puerto de la rutina de arranque de mensajería sean correctos.

El servidor utiliza algunos puertos por omisión, pero si existen más servidores instalados en ese sistema, se crean puertos alternativos durante la instalación para evitar conflictos con otras instancias de servidor. También puede utilizar la consola administrativa para determinar los puertos que el servidor está utilizando. Vaya a **Servidores > Servidores de aplicaciones > nombre\_su\_servidor > Configuración** y pulse **Puertos en Comunicación**. A continuación, puede editar el puerto que se está utilizando.

3. El cliente obtiene un contexto inicial con los valores correctos y busca los recursos a través de JNDI.
4. Utilizando las especificaciones JMS 1.1, el cliente accede a los destinos y a los mensajes de envío y recepción en los destinos.

### Resolución de problemas de enlaces JMS:

Puede diagnosticar y solucionar problemas con los enlaces JMS.

### Excepciones de implementación

Como respuesta a las distintas condiciones de error, la implementación de la importación y exportación JMS puede devolver uno de estos dos tipos de excepciones:

- Excepción empresarial de servicio: esta excepción se devuelve si se ha producido la excepción especificada de error en la interfaz empresarial de servicio (tipo de puerto WSDL).
- Excepción de tiempo de ejecución de servicio: se produce en todos los demás casos. En la mayoría de los casos, la excepción cause contendrá la excepción (JMSEException) original.

Por ejemplo, la importación espera sólo un mensaje de respuesta para cada mensaje de solicitud. Si llega más de una respuesta o si llega una respuesta con demora (una para la que se ha pasado el tiempo de caducidad de respuesta de SCA), se genera una excepción de tiempo de ejecución de servicio. Se retrotrae la transacción y el mensaje de respuesta se vuelve a poner fuera de la cola o lo gestiona el Gestor de sucesos anómalos.

### Condiciones de error principales

Las condiciones de error principales de los enlaces JMS quedan determinadas por la semántica de transacciones, por la configuración del proveedor JMS o por la referencia a un comportamiento existente de otros componentes. Las primeras condiciones de error son:

- No se puede conectar al proveedor JMS o al destino.

Si no se puede conectar al proveedor JMS para recibir mensajes, no se podrá iniciar el receptor de mensajes. Esta condición se registrará en el archivo de registro de WebSphere Application Server. Los mensajes permanentes permanecerán en el destino hasta que se recuperen satisfactoriamente (o caduquen).

Si no se puede conectar al proveedor JMS para enviar mensajes de salida, provocará la retrotracción de la transacción que controla el envío.

- No se puede analizar un mensaje de entrada o construir un mensaje de salida.

Una anomalía en el enlace de datos o en el manejador de datos provoca la retrotracción de la transacción que controla el trabajo.

- No se puede enviar el mensaje de salida.

Una anomalía al enviar un mensaje provoca una retrotracción de la transacción correspondiente.

- Varios mensajes de respuesta con demora o mensajes inesperados.

La importación espera sólo un mensaje de respuesta para cada mensaje de solicitud. Además, el periodo de tiempo válido en el que se puede recibir una respuesta viene determinado por el calificador de caducidad de respuesta de SCA en la solicitud. Cuando llega una respuesta o se supera el tiempo de espera, se suprime el registro de correlaciones. Si llegan mensajes de respuesta de forma inesperada o llegan con demora, se genera una excepción de tiempo de espera de servicio.

- Excepción en tiempo de ejecución de tiempo de espera excedido de servicio provocado por una respuesta con demora al utilizar el esquema de correlación de destino de respuesta dinámica temporal.

La importación de JMS provocará un tiempo de espera excedido después de un periodo de tiempo determinado por el calificador de caducidad de respuesta de SCA, o si esto no es establece utilizará como valor por omisión 60 segundos.

### Mensajes SCA basados en JMS que no aparecen en el gestor de sucesos anómalos

Si los mensajes SCA originados en una interacción JMS fallan, los mensajes deberán aparecer en el gestor de sucesos anómalos. Si los mensajes no aparecen en el gestor de sucesos anómalos, asegúrese de que el destino SIB subyacente del destino JMS tenga un valor máximo de entregas con error mayor que **1**. Si se establece este valor en **2** o más, se habilita la interacción con el gestor de sucesos anómalos durante las invocaciones de SCA para los enlaces JMS.

### Manejo de excepciones:

El modo en que está configurado el enlace de datos determina cómo se tratan las excepciones generadas por los manejadores de datos o los enlaces de datos. Además, la naturaleza del flujo de mediación dicta el comportamiento del sistema cuando se genera una excepción de este estilo.

Pueden surgir varios problemas cuando su enlace llama a un manejador de datos o enlace de datos. Por ejemplo, un manejador de datos puede recibir un mensaje que tiene una carga útil dañada, o puede tratar de leer un mensaje que tenga un formato incorrecto.

El modo en que el enlace gestiona una excepción de este estilo viene determinado por el modo de implementar el manejador de datos o el enlace de datos. El comportamiento recomendado es diseñar su propio enlace de datos para producir una excepción **DataBindingException**.

Cuando se produce cualquier excepción en tiempo de ejecución, incluida **DataBindingException**:

- Si el flujo de mediación está configurado para ser transaccional, el mensaje JMS , por omisión, se almacena en el Gestor de sucesos anómalos para la repetición o supresión manual.

**Nota:** Puede cambiar la modalidad de recuperación en el enlace, de modo que el mensaje se retrotraiga en lugar de almacenarse en el Gestor de sucesos anómalos.

- Si el flujo de mediación no es transaccional, el flujo de mediación se anota y se pierde el mensaje.

La situación es similar para un manejador de datos. Dado que el enlace de datos invoca el manejador de datos, cualquier excepción del manejador de datos se incluye en una excepción de enlace de datos. Por lo tanto, se le notifica una excepción **DataHandlerException** como una **DataBindingException**.

## Enlaces JMS genéricos

El enlace JMS genérico proporciona conectividad con proveedores externos compatibles con JMS 1.1. El funcionamiento de los enlaces JMS genéricos es similar a los enlaces JMS.

El servicios proporcionado mediante un enlace JMS permite a un módulo SCA (Service Component Architecture) realizar llamadas o recibir mensajes de sistemas externos. El sistema puede ser un sistema JMS externo.

El enlace JMS genérico proporciona integración con los proveedores JMS no de JCA compatibles con la versión 1.5 que soportan JMS 1.1 e implementa el recurso del servidor de aplicaciones JMS. El enlace JMS genérico soporta estos proveedores de JMS (incluidos Oracle AQ, TIBCO, SonicMQ, WebMethods, BEA WebLogic y WebSphere MQ) que no dan soporte a JCA 1.5, pero sí que dan soporte a Application Server Facility de la especificación JMS 1.1. Este enlace no soporta el proveedor JMS incorporado de WebSphere (SIBJMS), que es un proveedor JMS de JCA 1.5; al utilizar el proveedor, utilice los “Enlaces JMS” en la página 100.

Utilice este enlace genérico cuando integre con un sistema basado en JMS, no de JCA, compatible con la versión 1.5, dentro de un entorno SCA. Las aplicaciones externas de destino pueden posteriormente recibir mensajes y enviar mensajes para la integración con un componente SCA.

### Visión general de enlaces de JMS genérico:

Los enlaces JMS genéricos son enlaces JMS no de JCA que proporcionan conectividad entre el entorno de SCA (Service Component Architecture) y los sistemas JMS que cumplen con JMS 1.1 y que implementan el recurso del servidor de aplicaciones JMS.

### Enlaces JMS genéricos

Los aspectos más relevantes de los enlaces de importación y exportación JMS genéricos son:

- Puerto receptor: permite que los proveedores JMS no basados en JCA recibir mensajes y asignarlos a un Bean controlado por mensajes (MDB)
- Conexiones: encapsulan una conexión virtual entre un cliente y una aplicación de proveedor.
- Destinos: utilizados por un cliente para especificar el destino de los mensajes que produce o el origen de los mensajes que consume.
- Datos de autenticación: utilizados para proteger el acceso al enlace.

## Enlaces de importación JMS genéricos

Los enlaces de importación JMS genéricos permiten a los componentes del módulo SCA comunicarse con los servicios proporcionados por proveedores de JMS compatibles con sistemas no JCA 1.5.

La parte de la conexión de una importación JMS es una fábrica de conexiones. Una fábrica de conexiones es el objeto que utiliza un cliente para crear una conexión con un proveedor, encapsula un conjunto de parámetros de configuración de conexión definidos por un administrador. Cada fábrica de conexiones es una instancia de la interfaz `ConnectionFactory`, `QueueConnectionFactory` o `TopicConnectionFactory`.

La interacción con los sistemas JMS externos incluye la utilización de destinos para enviar peticiones y recibir respuestas.

Se da soporte a dos tipos de escenarios de uso para los enlaces de importación JMS genéricos, dependiendo del tipo de operación que se invoque:

- Unidireccional: la importación JMS genérica coloca un mensaje en el destino de envío configurado en el enlace de importación. No se envía nada al campo `replyTo` de la cabecera JMS.
- Bidireccional (petición y respuesta): la importación JMS genérica coloca un mensaje en el destino de envío y persiste la respuesta que recibe del componente SCA.

El destino indicado en `receive` se establece en la propiedad de la cabecera `replyTo` del mensaje de salida. Se despliega un bean MDB (Message Driven Bean) para escuchar en el destino de recepción y cuando se recibe una respuesta, el MDB vuelve a pasar la respuesta al componente.

El enlace de importación se puede configurar (mediante el campo **Esquema de correlación de respuesta** de Integration Designer) para que espere que el ID de correlación de mensaje de respuesta se haya copiado del ID mensaje de petición (el valor por omisión) o del ID de correlación de mensaje de petición.

Para los escenarios unidireccionales y bidireccionales, se pueden especificar propiedades dinámicas y estáticas. Las propiedades estáticas se pueden establecer desde el enlace de método de importación JMS genérica. Algunas de estas propiedades tienen significados especiales para el tiempo de ejecución JMS de SCA.

Resulta importante tener en cuenta que el JMS genérico es un enlace asíncrono. Si un componente llamante invoca una importación JMS genérica de forma síncrona (para una operación bidireccional), el componente llamante se bloquea hasta que el servicio JMS devuelve la respuesta.

Figura 34 en la página 111 ilustra cómo se enlaza la importación con el servicio externo.

## (Genérico) Importación JMS

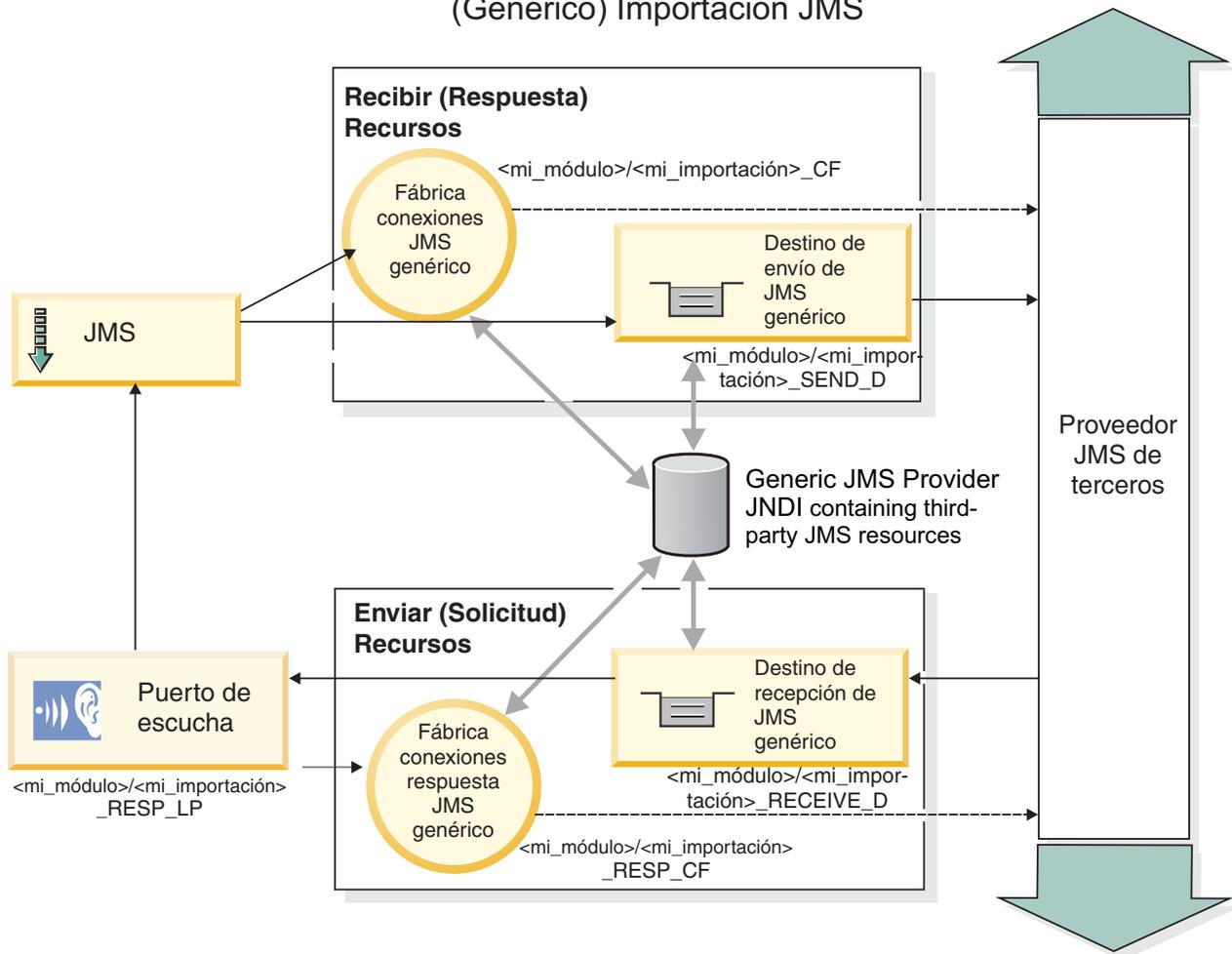


Figura 34. Recursos de enlace de importación de JMS genérico

### Enlaces de exportación de JMS genéricos

Los enlaces de exportación JMS genéricos proporcionan el medio mediante el cual los módulos SCA pueden ofrecer servicios a aplicaciones JMS externas.

La parte de conexión de una exportación JMS está formada por una `ConnectionFactory` y un `ListenerPort`.

Una exportación JMS genérica ha enviado y recibido destinos.

- El destino indicado en `receive` es el lugar donde debe colocarse el mensaje de entrada del componente de destino.
- El destino indicado en `send` es el lugar al que se enviará la respuesta, a menos que el mensaje de entrada lo altere temporalmente mediante la propiedad de cabecera `replyTo`.

Se despliega un MDB para que escuche las peticiones de entrada en el destino indicado en `receive` especificado en el enlace de exportación.

- El destino especificado en el campo `send` se utiliza para enviar la respuesta a la petición de entrada si el componente invocado proporciona una respuesta.
- El destino especificado en el campo `replyTo` del mensaje de entrada altera temporalmente el destino especificado en el campo `send`.

- En los escenarios de petición y respuesta, se puede configurar el enlace de importación (mediante el campo **Esquema de correlación de respuesta** de Integration Designer) para que espere que la respuesta copie el ID de mensaje de petición en el campo ID de correlación del mensaje de respuesta (valor por omisión), o que la respuesta pueda copiar el ID de correlación de petición en el campo ID de correlación del mensaje de respuesta.

Figura 35 ilustra cómo se enlaza el solicitante externo con la exportación.

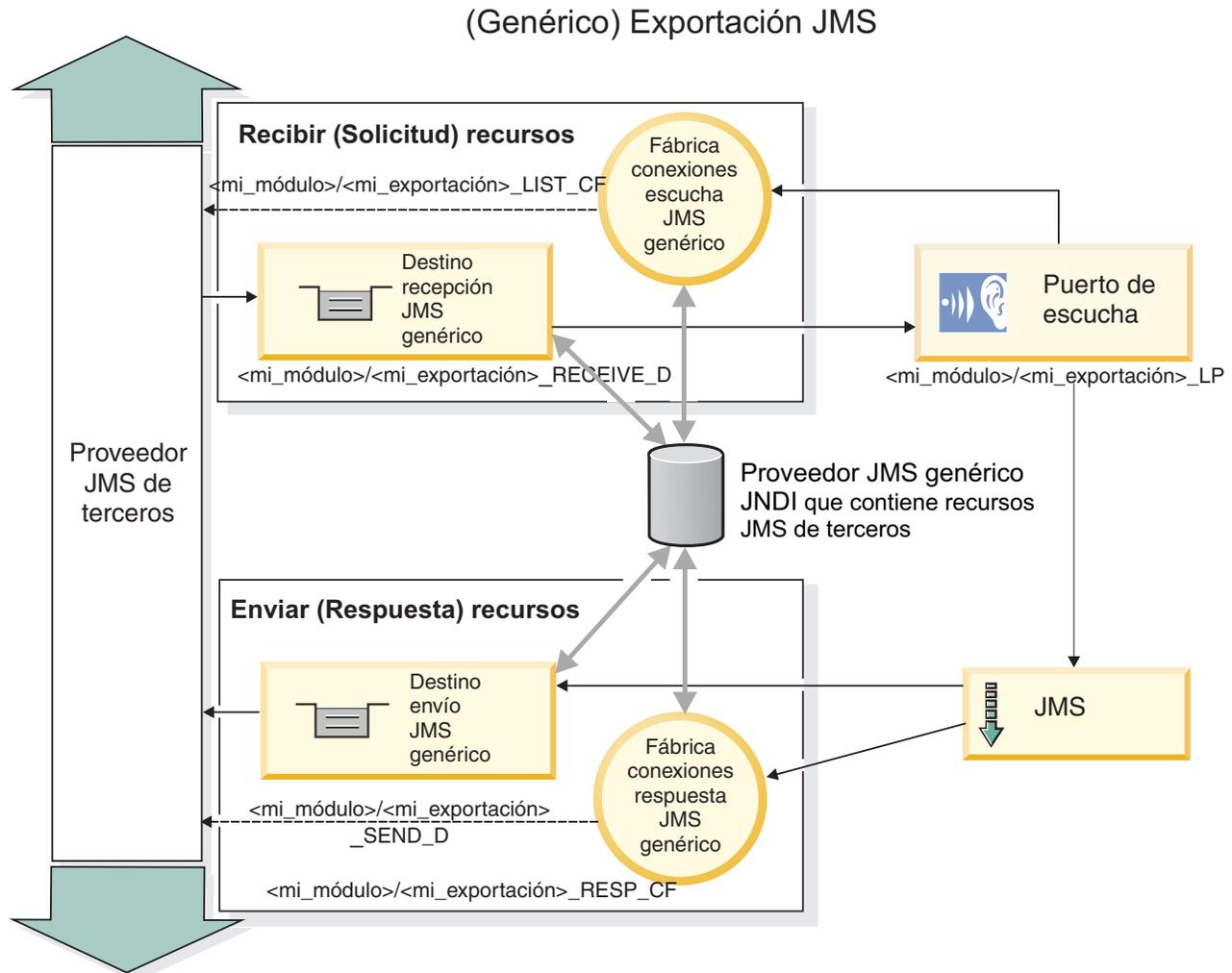


Figura 35. Recursos de enlace de exportación de JMS genérico

#### Características clave de los enlaces JMS genéricos:

Las características del enlace de exportación e importación JMS genérico son coherentes con las de los enlaces de importación JMS incorporados de WebSphere y JMS de MQ. Las características clave incluyen definiciones de cabecera y el acceso a los recursos Java EE existentes. No obstante, debido a su naturaleza genérica, no hay opciones de conectividad específicas del proveedor JMS y este enlace tiene una posibilidad limitada de generar recursos durante el despliegue y la instalación.

#### Importaciones genéricas

Al igual que la aplicación de importación JMS de MQ, la implementación JMS genérica es asíncrona y soporta tres invocaciones: unidireccional, bidireccional (también conocida como de petición-respuesta) y devolución de llamada.

Cuando se despliega la importación JMS, se despliega un bean controlado por mensajes (MDB) proporcionado por el entorno de ejecución. El MDB escucha las respuestas al mensaje de petición. El MDB se asocia (escucha) al destino enviado con la petición en el campo de cabecera replyTo del mensaje JMS.

### Exportaciones genéricas

Los enlaces de exportación JMS genéricos son distintos de los enlaces de exportación EIS en el manejo de la devolución de los resultados. Una exportación JMS genérica envía explícitamente la respuesta al destino replyTo especificado en el mensaje de entrada. Si no se especifica ninguno, se utiliza el destino de envío.

Cuando se despliega la exportación JMS genérica, se despliega un bean controlado por mensajes (un MDB distinto del utilizado para las importaciones JMS genéricas). Este escucha las peticiones de entrada en el destino de recepción y envía las peticiones para que las procese el tiempo de ejecución de SCA.

### Cabeceras especiales

Se utilizan las propiedades de cabecera especiales en las importaciones y exportaciones de JMS genéricos para indicar al enlace de destino cómo debe manejar el mensaje.

Por ejemplo, el selector de función por omisión utiliza la propiedad TargetFunctionName para identificar el nombre de la operación en la interfaz de exportación que se está invocando.

**Nota:** El enlace de importación se puede configurar para establecer la cabecera TargetFunctionName en el nombre de cada operación.

### Recursos Java EE

Se crea una serie de recursos Java EE cuando se despliega un enlace JMS en un entorno Java EE.

- El puerto de receptor para escuchar en el destino (respuesta) de recepción (sólo bidireccional) las importaciones y en el destino de recepción (petición) las exportaciones
- La fábrica de conexiones JMS genérica para outboundConnection (importación) e inboundConnection (exportación)
- El destino JMS genérico para los destinos de envío (importación) y recepción (exportación; sólo bidireccional)
- La fábrica de conexiones JMS genérica para responseConnection (sólo bidireccional y opcional; de lo contrario, se utiliza outboundConnection para las importaciones y se utiliza inboundConnection para las exportaciones)
- El destino JMS genérico para el destino de recepción (importación) y envío (exportación) (sólo bidireccional)
- El destino JMS de devolución de llamada del proveedor de mensajería por omisión se utiliza para acceder al destino de cola de devolución de llamada SIB (sólo bidireccional)
- La fábrica de conexiones JMS de devolución de llamada del proveedor de mensajería por omisión se utiliza para acceder al destino JMS de devolución de llamada (sólo bidireccional)
- El destino de cola de devolución de llamada SIB se utiliza para almacenar información acerca del mensaje de petición para utilizarlo durante el proceso de la respuesta (sólo bidireccional)

La tarea de instalación crea ConnectionFactory, los tres destinos y ActivationSpec a partir de la información de los archivos de importación y exportación.

### Cabeceras JMS genéricas:

Las cabeceras JMS genéricas son objetos SDO (Service Data Objects) que contienen todas las propiedades de los mensajes JMS genéricos. Estas propiedades pueden ser del mensaje de entrada o pueden ser las propiedades que se aplicarán al mensaje de salida.

Un mensaje JMS contiene dos tipos de cabeceras: la cabecera del sistema JMS y varias propiedades JMS. Se puede acceder a ambos tipos de cabeceras, bien en un módulo de mediación del objeto de mensajes de servicios (SMO), o bien mediante la API ContextService.

Las propiedades siguientes se establecen estáticamente en `methodBinding`:

- `JMSType`
- `JMSCorrelationID`
- `JMSDeliveryMode`
- `JMSPriority`

El enlace JMS genérico también da soporte a la modificación dinámica de las cabeceras JMS y de las propiedades del mismo modo que los enlaces JMS y los enlaces JMS de MQ.

Algunos proveedores JMS genéricos imponen restricciones sobre las propiedades que puede establecer la aplicación y sus combinaciones. Debe consultar la documentación del producto de terceros para obtener más información. No obstante, se ha añadido una propiedad adicional a `methodBinding`, `ignoreInvalidOutboundJMSProperties`, que permite propagar cualquier excepción.

Las cabeceras JMS genéricas y las propiedades de los mensajes se utilizan únicamente cuando se ha activado el conmutador del enlace SCDL de SCA (Service Component Architecture). Cuando se activa el conmutador, se propaga la información de contexto. Por omisión, este conmutador está encendido. Para impedir que se propague la información de contexto, cambie el valor a **false**.

Cuando la propagación de contexto está habilitada, la información de cabecera puede fluir al mensaje o al componente de destino. Para activar y desactivar la propagación del contexto, especifique **true** o **false** para el atributo `contextPropagationEnabled` de los enlaces de importación y exportación. Por ejemplo:

```
<esbBinding xsi:type="eis:JMSImportBinding" contextProgagationEnabled="true">
```

El valor por omisión es **true**.

### **Resolución de problemas de enlaces JMS genéricos:**

Puede diagnosticar y corregir problemas con enlaces JMS genéricos.

### **Excepciones de implementación**

Como respuesta a las distintas condiciones de error, la implementación de la importación y exportación JMS genérica puede devolver uno de estos dos tipos de excepciones:

- Excepción empresarial de servicio: esta excepción se devuelve si se ha producido la excepción especificada de error en la interfaz empresarial de servicio (tipo de puerto WSDL).
- Excepción de tiempo de ejecución de servicio: se produce en todos los demás casos. En la mayoría de los casos, la excepción cause contendrá la excepción (`JMSException`) original.

### **Resolución de problemas de caducidad de mensajes JMS genéricos**

Los mensajes de petición de un proveedor de JMS caducan.

La *caducidad de petición* hace referencia a la caducidad de un mensaje de petición de un proveedor JMS cuando se alcanza la hora definida por `JMSExpiration` en el mensaje de petición. Tal como sucede con otros enlaces JMS, el enlace JMS genérico gestiona la caducidad estableciendo la del mensaje de

devolución de llamada efectuado por la importación en el mismo valor que para la solicitud de salida. La notificación de la caducidad del mensaje de devolución de llamada indicará que el mensaje de petición ha caducado y que se debe notificar al cliente mediante una excepción empresarial.

No obstante, si el destino de la devolución de llamada pasa al proveedor de terceros, este tipo de caducidad de petición no está soportada.

La *caducidad de respuesta* hace referencia a la caducidad de un mensaje de respuesta de un proveedor JMS cuando se alcanza la hora definida por `JMSExpiration` en el mensaje de respuesta.

La caducidad de respuesta para el enlace JMS genérico no está soportada porque el comportamiento exacto de la caducidad de un proveedor JMS de terceros no está definido. Sin embargo, puede comprobar que la respuesta no haya caducado cuando se reciba.

Para los mensajes de petición de salida, el valor `JMSExpiration` se calculará a partir del tiempo que se haya esperado y de los valores `requestExpiration` contenidos en `asyncHeader`, si se han establecido.

### **Resolución de problemas de errores de fábrica de conexiones JMS genéricos**

Cuando define determinados tipos de fábricas de conexiones en su proveedor JMS genérico, es posible que reciba un mensaje de error cuando intenta iniciar una aplicación. Puede modificar la fábrica de conexiones externa para evitar este problema.

Al iniciar una aplicación, puede recibir el siguiente mensaje de error:

```
El tipo JMSConnectionFactory  
del puerto de escucha MDB no coincide con el tipo JMSDestination
```

Este problema puede surgir cuando define las fábricas de conexiones externas. Específicamente la excepción se puede generar cuando crea una fábrica de conexiones de temas JMS 1.0.2, en lugar de una fábrica de conexiones JMS 1.1 (unificada) (es decir, una que pueda dar soporte a las comunicaciones de punto a punto y de publicación/suscripción).

Para solucionar este problema, lleve a cabo los pasos siguientes:

1. Acceda al proveedor JMS genérico que está utilizando.
2. Sustituya la fábrica de conexiones de temas JMS 1.0.2 que ha definido por una fábrica de conexiones JMS 1.1 (unificada).

Cuando inicie la aplicación con la fábrica de conexiones JMS 1.1 que acaba de definir, ya no recibirá ningún mensaje de error.

### **Mensajes SCA basados en JMS genéricas que no aparecen en el gestor de sucesos anómalos**

Si los mensajes SCA originados en una interacción JMS genérica fallan, los mensajes deberán aparecer en el gestor de sucesos anómalos. Si los mensajes no aparecen en el gestor de sucesos anómalos, asegúrese de que el valor de la propiedad de máximo de reintentos en el puerto receptor subyacente sea mayor o igual que 1. Si establece este valor en 1 o más, se habilita la interacción con el gestor de sucesos anómalos durante las invocaciones de SCA para los enlaces JMS genéricos.

### **Manejo de excepciones:**

El modo en que está configurado el enlace de datos determina cómo se tratan las excepciones generadas por los manejadores de datos o los enlaces de datos. Además, la naturaleza del flujo de mediación dicta el comportamiento del sistema cuando se genera una excepción de este estilo.

Pueden surgir varios problemas cuando su enlace llama a un manejador de datos o enlace de datos. Por ejemplo, un manejador de datos puede recibir un mensaje que tiene una carga útil dañada, o puede tratar de leer un mensaje que tenga un formato incorrecto.

El modo en que el enlace gestiona una excepción de este estilo viene determinado por el modo de implementar el manejador de datos o el enlace de datos. El comportamiento recomendado es diseñar su propio enlace de datos para producir una excepción **DataBindingException**.

La situación es similar para un manejador de datos. Dado que el enlace de datos invoca el manejador de datos, cualquier excepción del manejador de datos se incluye en una excepción de enlace de datos. Por lo tanto, se le notifica una excepción **DataHandlerException** como una **DataBindingException**.

Cuando se produce cualquier excepción de tiempo de ejecución, incluida la excepción **DataBindingException**:

- Si el flujo de mediación se ha configurado para ser transaccional, el mensaje JMS se almacena en el gestor de sucesos anómalos de forma predeterminada para una respuesta o supresión manual.

**Nota:** Puede cambiar la modalidad de recuperación en el enlace, de forma que el mensaje se retrotrae, en lugar de almacenarse en el gestor de sucesos anómalos.

- Si el flujo de mediación no es transaccional, el flujo de mediación se anota y se pierde el mensaje.

La situación es similar para un manejador de datos. Puesto que el enlace de datos llama al manejador de datos, se produce una excepción de manejador de datos dentro de una excepción de enlace de datos. Por lo tanto, se informa de una excepción **DataHandlerException** como una **DataBindingException**.

## Enlaces JMS de WebSphere MQ

El enlace JMS de WebSphere MQ proporciona integración con aplicaciones externas que utilizan un proveedor basado en JMS de WebSphere MQ.

Utilice los enlaces de exportación e importación de WebSphere MQ para integrarse directamente con los sistemas JMS o JMS MQ externos desde el entorno de servidor. Esto elimina la necesidad de utilizar las características de Enlace MQ o Enlace de cliente de Service Integration Bus.

Cuando un componente interactúa con un servicio basado en JMS de WebSphere MQ, a través de una importación, el enlace de importación JMS de WebSphere MQ utiliza un destino al que se enviarán los datos y un destino donde se puede recibir la respuesta. La conversión de los datos a y desde un mensaje JMS se realiza mediante el manejador de datos JMS o el componente Data Binding Edge.

Cuando un módulo SCA proporciona un servicio a clientes JMS de WebSphere MQ, el enlace de exportación JMS de WebSphere MQ utiliza un destino donde se puede recibir la petición y donde se puede enviar la respuesta. La conversión de los datos a y desde un mensaje JMS se realiza a través del manejador de datos o el enlace de datos JMS.

El selector de función proporciona una correlación con la operación del componente de destino que se va a invocar.

### Visión general de enlaces JMS de WebSphere MQ:

El enlace JMS de WebSphere MQ proporciona integración con las aplicaciones externas que utilizan el proveedor de JMS de WebSphere MQ.

### Tareas administrativas de WebSphere MQ

Se espera que el administrador del sistema WebSphere MQ cree el gestor de colas WebSphere MQ subyacente, que utilizará los enlaces JMS de WebSphere MQ, antes de ejecutar una aplicación que contiene estos enlaces.

## Enlaces de importación JMS de WebSphere MQ

La importación JMS de WebSphere MQ permite a los componentes del módulo SCA comunicarse con los servicios proporcionados por los proveedores basados en JMS de WebSphere MQ. Debe utilizar una versión soportada de WebSphere MQ. Puede encontrar los requisitos detallados de hardware y software en las páginas de soporte de IBM.

Se da soporte a dos tipos de situaciones de uso para los enlaces de importación JMS de WebSphere MQ, dependiendo del tipo de operación que se invoque:

- Unidireccional: la importación JMS de WebSphere MQ coloca un mensaje en el destino de envío configurado en el enlace de importación. No se envía nada al campo `replyTo` de la cabecera JMS.
- Bidireccional (petición y respuesta): la importación JMS de WebSphere MQ coloca un mensaje en el destino de envío.

El destino indicado en `receive` se establece en el campo de la cabecera `replyTo`. Se despliega un bean MDB (Message Driven Bean) para escuchar en el destino de recepción y cuando se recibe una respuesta, el MDB vuelve a pasar la respuesta al componente.

El enlace de importación se puede configurar (mediante el campo **Esquema de correlación de respuesta** de Integration Designer) para que espere que el ID de correlación de mensaje de respuesta se haya copiado del ID mensaje de petición (el valor por omisión) o del ID de correlación de mensaje de petición.

Para los escenarios de uso unidireccionales y bidireccionales, se pueden especificar propiedades de cabecera dinámicas y estáticas. Las propiedades estáticas se pueden establecer desde el enlace de método de importación JMS. Algunas de estas propiedades tienen significados especiales para el tiempo de ejecución JMS de SCA.

Resulta importante tener en cuenta que JMS de WebSphere MQ es un enlace asíncrono. Si un componente llamante invoca una importación JMS de WebSphere MQ de forma síncrona (para una operación bidireccional), el componente llamante se bloquea hasta que el servicio JMS devuelve la respuesta.

Figura 36 en la página 118 ilustra cómo se enlaza la importación con el servicio externo.

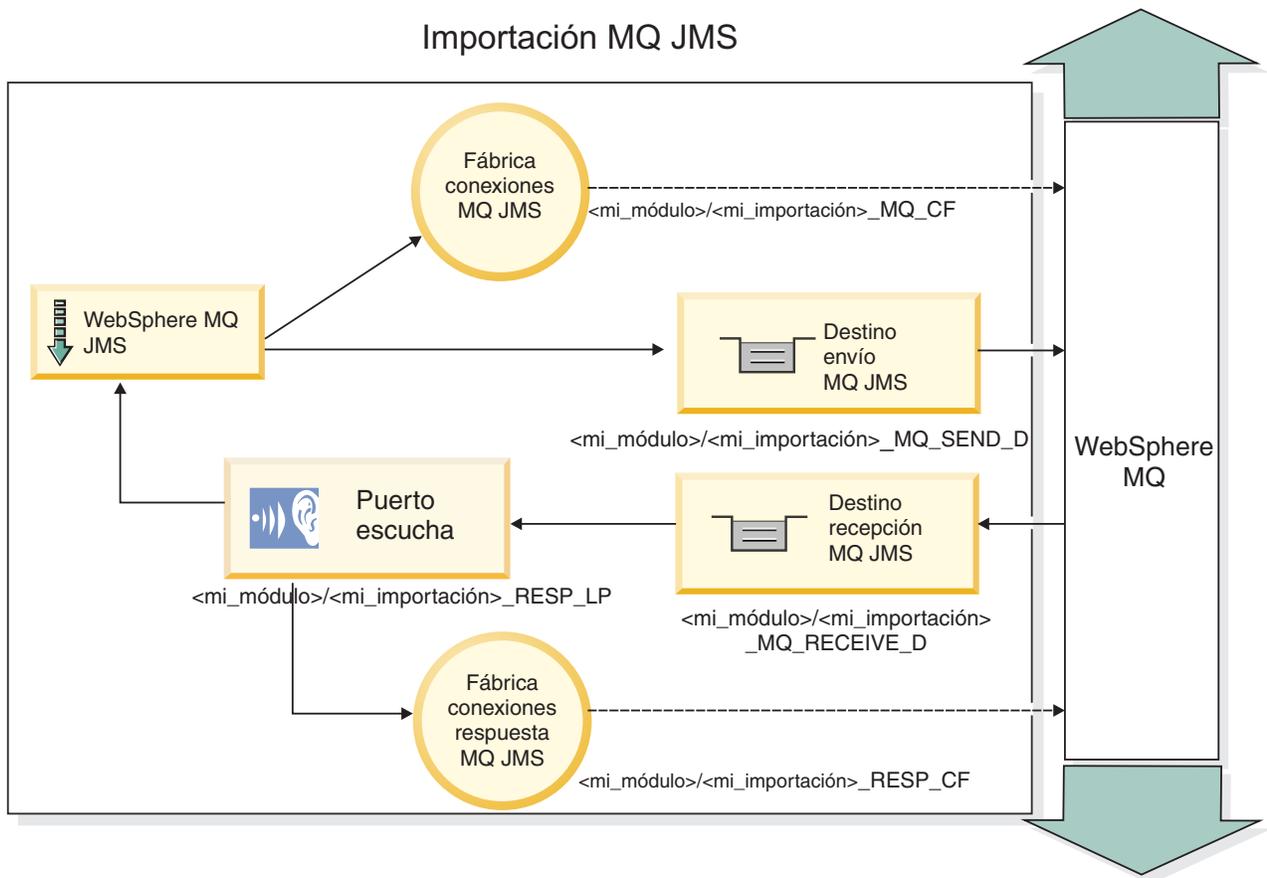


Figura 36. Recursos de enlace de importación de JMS de WebSphere MQ

### Enlaces de exportación JMS de WebSphere MQ

El enlace de exportación JMS de WebSphere MQ proporciona el medio mediante el cual los módulos SCA pueden ofrecer servicios a aplicaciones JMS externas en el proveedor de JMS basado en WebSphere MQ.

Se despliega un MDB para que escuche las peticiones de entrada en el destino indicado en receive especificado en el enlace de exportación. El destino especificado en el campo send se utiliza para enviar la respuesta a la petición de entrada si el componente invocado proporciona una respuesta. El destino especificado en el campo replyTo del mensaje de respuesta altera temporalmente el destino especificado en el campo send.

Figura 37 en la página 119 ilustra cómo se enlaza el solicitante externo con la exportación.

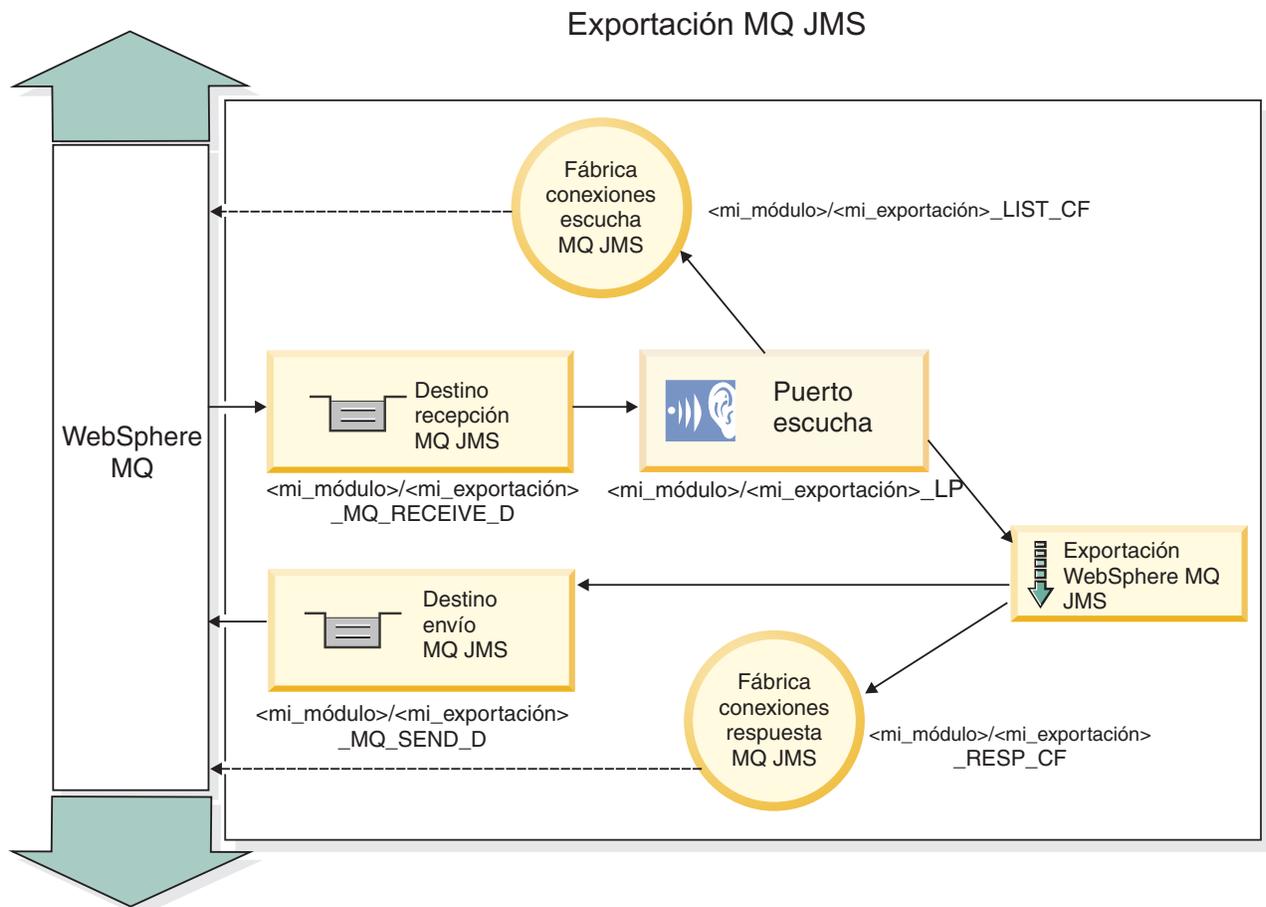


Figura 37. Recursos de enlace de exportación de JMS de WebSphere MQ

**Nota:** La Figura 36 en la página 118 y la Figura 37 ilustran cómo una aplicación de una versión anterior de IBM Business Process Manager se enlaza a un servicio externo. Para aplicaciones desarrolladas para IBM Business Process Manager Versión 7.0, se utiliza la Especificación de activación en lugar del Puerto receptor y de la Fábrica de conexiones.

### Características clave de los enlaces JMS de WebSphere MQ:

Entre las características clave de un enlace JMS de WebSphere MQ se incluyen las cabeceras, los artefactos Java EE y los recursos Java EE creados.

#### Cabeceras

Una cabecera de mensaje JMS contiene varios campos predefinidos que incluyen valores que utilizan los clientes y los proveedores para identificar y direccionar mensajes. Pueden usarse propiedades de enlace para configurar estas cabeceras con valores fijos o se pueden especificar dinámicamente en tiempo de ejecución.

#### JMSCorrelationID

Enlaza con un mensaje relacionado. Normalmente, este campo se establece en la serie del identificador del mensaje al que se está respondiendo.

#### TargetFunctionName

Esta cabecera la utiliza uno de los selectores de función para identificar la operación que se está invocando. Establecer la propiedad de cabecera JMS TargetFunctionName en un mensaje enviado a una exportación JMS permite utilizar este selector de función. La propiedad se puede establecer

directamente en aplicaciones cliente de JMS o al conectar una importación con un JMS enlazando a esa exportación. En este caso, el enlace de importación de JMS debe configurarse para establecer la cabecera TargetFunctionName de cada operación de la interfaz en el nombre de la operación.

## Esquemas de correlación

Los enlaces JMS de WebSphere MQ proporcionan varios esquemas de correlación que se utilizan para determinar cómo correlacionar los mensajes de petición con los mensajes de respuesta.

### RequestMsgIDToCorrelID

El JMSMessageID se copia en el campo JMSCorrelationID. Éste es el valor por omisión.

### RequestCorrelIDToCorrelID

El JMSCorrelationID se copia en el campo JMSCorrelationID.

## Recursos Java EE

Se crea una serie de recursos Java EE cuando se despliega una importación de JMS de MQ en un entorno Java EE.

## Parámetros

### Fábrica de conexiones MQ

Utilizada por los clientes para crear una conexión con el proveedor de JMS de MQ.

### Fábrica de conexiones de respuesta

Utilizada por el tiempo de ejecución JMS MQ SCA cuando el destino de envío está en un gestor de colas diferente que el destino de recepción.

### Especificación de activación

Una especificación de activación JMS se asocia con uno o más beans controlados por mensajes y proporciona la configuración necesaria para que ellos reciban mensajes.

## Destinos

- Destino de envío:
  - Importaciones: donde se envía la petición o el mensaje de salida.
  - Exportaciones: donde se enviará el mensaje de respuesta, si no se reemplaza por el campo de cabecera JMSReplyTo en el mensaje de entrada.
- Destino de recepción:
  - Importaciones: donde se debe colocar la respuesta o el mensaje de entrada.
  - En las exportaciones, donde se debe colocar el mensaje de entrada o de petición.

## Cabeceras JMS:

Un mensaje JMS contiene dos tipos de cabeceras: la cabecera del sistema JMS y varias propiedades JMS. Se puede acceder a ambos tipos de cabeceras, bien en un módulo de mediación del objeto de mensajes de servicios (SMO), o bien mediante la API ContextService.

### Cabecera del sistema JMS

La cabecera del sistema JMS se representa en el SMO mediante el elemento JMSHeader, que contiene todos los campos que se encuentran, normalmente, en una cabecera JMS. Aunque éstos pueden modificarse en la mediación (o en ContextService), algunos campos de cabecera del sistema JMS establecidos en el SMO no se propagarán en el mensaje JMS de salida, ya que el sistema, o los valores estáticos, los alteran temporalmente.

Los campos clave de la cabecera del sistema JMS que se pueden actualizar en la mediación (o en ContextService) son:

- **JMSType** y **JMSCorrelationID**: valores de las propiedades de la cabecera del mensaje predefinidas específicas.
- **JMSDeliveryMode**: valores de la modalidad de entrega (persistentes o no persistentes). El valor por omisión es persistente).
- **JMSPriority**: valor de prioridad (0 a 9). El valor por omisión es JMS\_Default\_Priority).

### Propiedades JMS

Las propiedades JMS se representan en el SMO como entradas de la lista Propiedades. Las propiedades se pueden añadir, actualizar o suprimir dentro de una mediación, o bien utilizando la API ContextService.

Las propiedades también pueden establecerse estáticamente en el enlace JMS. Las propiedades que se establecen estáticamente alteran temporalmente los valores (que tengan el mismo nombre) que se establecen dinámicamente.

Las propiedades de usuario propagadas desde otros enlaces (por ejemplo, un enlace HTTP) generarán una salida en el enlace JMS en forma de propiedades JMS.

### Valores de propagación de cabecera

La propagación de las propiedades y la cabecera del sistema JMS bien sea a desde el mensaje JMS de entrada a los componentes ubicados en sentido descendente, o bien sea desde los componentes ubicados en sentido ascendente al mensaje JMS de salida, se pueden controlar mediante el distintivo Propagate Protocol Header del enlace.

Cuando se establece Propagate Protocol Header, la información de cabecera puede fluir al mensaje o al componente de destino, tal como se describe en la lista siguiente:

- **Petición de exportación JMS**  
La cabecera JMS recibida en el mensaje se propagará a los componentes de destino por medio del servicio de contexto. Las propiedades JMS recibidas en el mensaje se propagará a los componentes de destino por medio del servicio de contexto.
- **Respuesta de exportación JMS**  
Cualquiera de los campos de cabecera JMS establecidos en el servicio de contexto se utilizará en el mensaje de salida, si no lo han alterado temporalmente las propiedades establecidas en el enlace de exportación de JMS. Cualquiera de las propiedades establecidas en el servicio de contexto se utilizará en el mensaje de salida, si no la han alterado temporalmente las propiedades establecidas en el enlace de exportación de JMS.
- **Petición de importación JMS**  
Cualquiera de los campos de cabecera JMS establecidos en el servicio de contexto se utilizará en el mensaje de salida, si no lo han alterado temporalmente las propiedades establecidas en el enlace de importación de JMS. Cualquiera de las propiedades establecidas en el servicio de contexto se utilizará en el mensaje de salida, si no la han alterado temporalmente las propiedades establecidas en el enlace de importación de JMS.
- **Respuesta de importación JMS**  
La cabecera JMS recibida en el mensaje se propagará a los componentes de destino por medio del servicio de contexto. Las propiedades JMS recibidas en el mensaje se propagará a los componentes de destino por medio del servicio de contexto.

### Clientes externos:

El servidor puede enviar mensajes o recibirlos de clientes externos utilizando los enlaces JMS de WebSphere MQ.

Un cliente externo (como un portal Web o un sistema de empresa) puede enviar un mensaje a un componente SCA de la aplicación mediante una exportación, o lo puede invocar un componente SCA de la aplicación mediante una importación.

El enlace de exportación JMS de WebSphere MQ despliega los beans controlados por mensajes (MDB) para escuchar las peticiones procedentes del destino especificado en receive especificado en el enlace de exportación. El destino especificado en el campo send se utiliza para enviar la respuesta a la petición de entrada si la aplicación invocada proporciona una respuesta. De esta forma, un cliente externo puede invocar aplicaciones a través del enlace de exportación.

Las importaciones de JMS de WebSphere MQ se enlazan con clientes externos y pueden entregar mensajes a éstos. Este mensaje puede necesitar o no una respuesta del cliente externo.

Puede encontrar más información sobre cómo interactuar con clientes externos utilizando WebSphere MQ en el Centro de información de WebSphere MQ.

### **Resolución de problemas de enlaces JMS de WebSphere MQ:**

Puede diagnosticar y solucionar problemas con los enlaces JMS de WebSphere MQ.

### **Excepciones de implementación**

Como respuesta a las distintas condiciones de error, la implementación de la importación y exportación de JMS puede devolver uno de estos dos tipos de excepciones:

- Excepción empresarial de servicio: esta excepción se devuelve si se ha producido la excepción especificada de error en la interfaz empresarial de servicio (tipo de puerto WSDL).
- Excepción de tiempo de ejecución de servicio: se produce en todos los demás casos. En la mayoría de los casos, la excepción cause contendrá la excepción (JMSEException) original.

Por ejemplo, la importación espera sólo un mensaje de respuesta para cada mensaje de solicitud. Si llega más de una respuesta o si llega una respuesta con demora (una para la que se ha pasado el tiempo de caducidad de respuesta de SCA), se genera una excepción de tiempo de ejecución de servicio. Se retrotrae la transacción y el mensaje de respuesta se vuelve a poner fuera de la cola o lo gestiona el Gestor de sucesos anómalos.

### **Mensajes SCA basados en JMS de WebSphere MQ que no aparecen en el gestor de sucesos anómalos**

Si los mensajes SCA originados en una interacción JMS de WebSphere MQ fallan, los mensajes deberán aparecer en el gestor de sucesos anómalos. Si los mensajes no aparecen en el gestor de sucesos anómalos, asegúrese de que el valor de la propiedad de máximo de reintentos en el puerto receptor subyacente sea mayor o igual que 1. Si establece este valor en 1 o más, se habilita la interacción con el gestor de sucesos anómalos durante las invocaciones de SCA para los mensajes JMS de MQ.

### **Escenarios de utilización errónea: comparación con los enlaces de WebSphere MQ**

El enlace JMS de WebSphere MQ se ha diseñado para interoperar con las aplicaciones JMS desplegadas en WebSphere MQ, que expone los mensajes de acuerdo con el modelo de mensaje JMS. La importación y exportación de WebSphere MQ, sin embargo, están diseñadas principalmente para interoperar con las aplicaciones WebSphere MQ nativas y exponer el contenido completo del cuerpo del mensaje WebSphere MQ en las mediaciones.

Los escenarios siguientes se deben crear utilizando el enlace JMS de WebSphere MQ, no el enlace de WebSphere MQ:

- Invocación de un bean controlado por mensaje JMS (MDB) desde un módulo SCA, donde el MDB se despliega en el proveedor de JMS de WebSphere MQ. Utilice una importación de JMS de WebSphere MQ.

- Permitir que se invoque el módulo SCA desde un servlet de componentes Java EE o desde EJB por medio de JMS. Utilice una exportación de JMS de WebSphere MQ.
- Mediación de los contenidos de un MapMessage JMS, en tránsito entre WebSphere MQ. Utilice una exportación y una importación de JMS de WebSphere MQ junto con el enlace de datos o manejador de datos adecuado.

Existen situaciones en las cuales es posible que se espera la interoperatividad entre el enlace WebSphere MQ y el enlace JMS de WebSphere MQ. En particular, cuando realiza un puente entre aplicaciones Java EE y no Java EE de WebSphere MQ, utilice una exportación de WebSphere MQ y una importación de JMS de WebSphere MQ (o viceversa) junto con los enlaces de datos apropiados o los módulos de mediación (o ambos).

### Manejo de excepciones:

El modo en que está configurado el enlace de datos determina cómo se tratan las excepciones generadas por los manejadores de datos o los enlaces de datos. Además, la naturaleza del flujo de mediación dicta el comportamiento del sistema cuando se genera una excepción de este estilo.

Pueden surgir varios problemas cuando su enlace llama a un manejador de datos o enlace de datos. Por ejemplo, un manejador de datos puede recibir un mensaje que tiene una carga útil dañada, o puede tratar de leer un mensaje que tenga un formato incorrecto.

El modo en que el enlace gestiona una excepción de este estilo viene determinado por el modo de implementar el manejador de datos o el enlace de datos. El comportamiento recomendado es diseñar su propio enlace de datos para producir una excepción **DataBindingException**.

La situación es similar para un manejador de datos. Dado que el enlace de datos invoca el manejador de datos, cualquier excepción del manejador de datos se incluye en una excepción de enlace de datos. Por lo tanto, se le notifica una excepción **DataHandlerException** como una **DataBindingException**.

Cuando se produce cualquier excepción de tiempo de ejecución, incluida la excepción **DataBindingException**:

- Si el flujo de mediación se ha configurado para ser transaccional, el mensaje JMS se almacena en el gestor de sucesos anómalos de forma predeterminada para una respuesta o supresión manual.

**Nota:** Puede cambiar la modalidad de recuperación en el enlace, de forma que el mensaje se retrotrae, en lugar de almacenarse en el gestor de sucesos anómalos.

- Si el flujo de mediación no es transaccional, el flujo de mediación se anota y se pierde el mensaje.

La situación es similar para un manejador de datos. Puesto que el enlace de datos llama al manejador de datos, se produce una excepción de manejador de datos dentro de una excepción de enlace de datos. Por lo tanto, se informa de una excepción **DataHandlerException** como una **DataBindingException**.

## Enlaces de WebSphere MQ

El enlace de WebSphere MQ proporciona la conectividad SCA (Service Component Architecture) con las aplicaciones WebSphere MQ.

Utilice los enlaces de exportación e importación de WebSphere MQ para integrarse directamente con un sistema basado en WebSphere MQ desde el entorno del servidor. Esto elimina la necesidad de utilizar las características de Enlace MQ o Enlace de cliente de Service Integration Bus.

Cuando un componente interactúa con un servicio de WebSphere MQ a través de una importación, el enlace de importación de WebSphere MQ utiliza una cola a la que se enviarán los datos y una cola donde se puede recibir la respuesta.

Cuando un módulo SCA proporciona un servicio a clientes de WebSphere MQ, el enlace de exportación de WebSphere MQ utiliza una cola donde se puede recibir la petición y se puede enviar la respuesta. El selector de función proporciona una correlación con la operación del componente de destino que se va a invocar.

La conversión de los datos de carga útil a y desde un mensaje MQ se realiza mediante el manejador de datos del cuerpo MQ o el enlace de datos. La conversión de los datos de la cabecera a y desde un mensaje MQ se realiza mediante el enlace de datos de la cabecera MQ.

Para obtener información sobre las versiones soportadas de WebSphere MQ, consulte la página web de requisitos del sistema: Requisitos del sistema de IBM Business Process Manager

### Visión general de enlaces WebSphere MQ:

El enlace de WebSphere MQ proporciona integración con las aplicaciones basadas en MQ nativo.

### Tareas administrativas de WebSphere MQ

El administrador del sistema WebSphere MQ debe crear el gestor de colas WebSphere MQ subyacente, que utilizarán los enlaces de WebSphere MQ antes de ejecutar una aplicación que contenga estos enlaces.

### Tareas administrativas de WebSphere

Debe establecer la propiedad **Vía de acceso de biblioteca nativa** del adaptador de recursos MQ en Websphere en la versión de WebSphere MQ soportada por el servidor y reinicie el servidor. Esto garantiza que se utilicen las bibliotecas de una versión soportada de WebSphere MQ. Puede encontrar los requisitos detallados de hardware y software en las páginas de soporte de IBM.

### Enlaces de importación WebSphere MQ

El enlace de importación de WebSphere MQ permite a los componentes del módulo SCA comunicarse con los servicios proporcionados por las aplicaciones externas basadas en WebSphere MQ. Debe utilizar una versión soportada de WebSphere MQ. Puede encontrar los requisitos detallados de hardware y software en las páginas de soporte de IBM.

La interacción con los sistemas WebSphere MQ externos incluye la utilización de colas para enviar peticiones y recibir respuestas.

Se da soporte a dos tipos de escenarios de uso para los enlaces de importación de WebSphere MQ, dependiendo del tipo de operación que se invoque:

- Unidireccional: la importación de WebSphere MQ coloca un mensaje en la cola configurada en el campo **Cola de destino de envío** del enlace de importación. No se envía nada al campo replyTo de la cabecera MQMD.
- Bidireccional (petición y respuesta): la importación de WebSphere MQ coloca un mensaje en la cola configurada en el campo **Cola de destino de envío**.

La cola de recepción se establece en el campo de la cabecera MQMD replyTo. Se despliega un bean MDB (Message Driven Bean) para escuchar en la cola de recepción y cuando se recibe una respuesta, el MDB vuelve a pasar la respuesta al componente.

El enlace de importación se puede configurar (mediante el campo **Esquema de correlación de respuesta** para esperar a que el ID de correlación de mensaje de respuesta se haya copiado del ID mensaje de petición (el valor por omisión) o del ID de correlación de mensaje de petición.

Resulta importante tener en cuenta que WebSphere MQ es un enlace asíncrono. Si un componente llamante invoca una importación de WebSphere MQ de forma síncrona (para una operación bidireccional), el componente llamante se bloquea hasta que el servicio WebSphere MQ devuelve la respuesta.

Figura 38 ilustra cómo se enlaza la importación con el servicio externo.

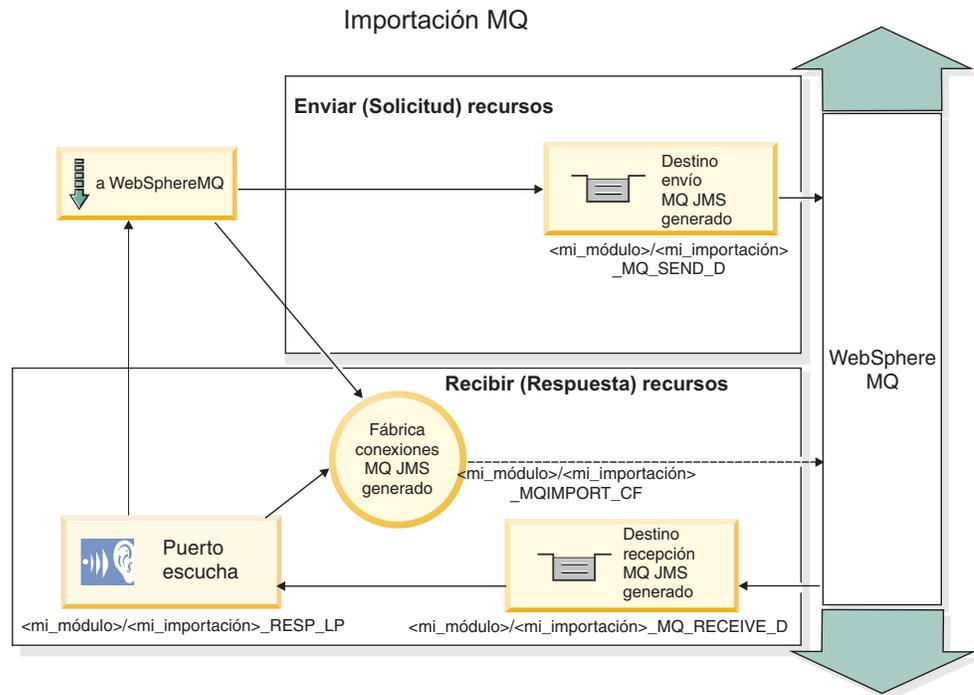


Figura 38. Recursos de enlace de importación de WebSphere MQ

### Enlaces de exportación de WebSphere MQ

El enlace de exportación de WebSphere MQ proporciona el medio mediante el cual los módulos SCA pueden ofrecer servicios a aplicaciones externas basadas en WebSphere MQ.

Se despliega un MDB para que escuche las peticiones de entrada en la **cola de destino de recepción** especificada en el enlace de exportación. La cola especificada en el campo **Cola de destino de envío** se utiliza para enviar la respuesta a la petición de entrada si el componente invocado proporciona una respuesta. La cola especificada en el campo replyTo del mensaje de respuesta altera temporalmente la cola especificada en el campo **Cola de destino de envío**.

Figura 39 en la página 126 ilustra cómo se enlaza el solicitante externo con la exportación.

## Exportación MQ

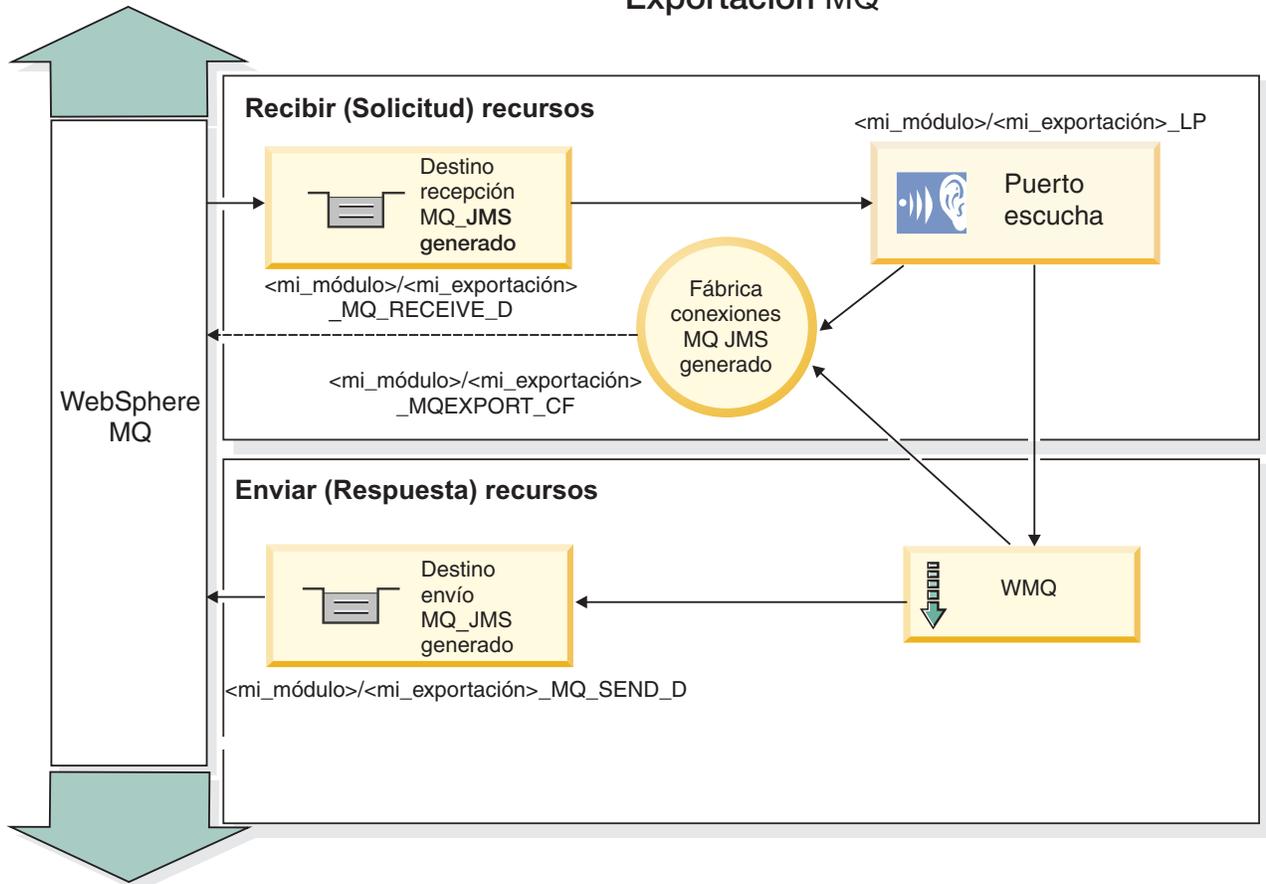


Figura 39. Recursos de enlace de exportación de WebSphere MQ

**Nota:** La Figura 38 en la página 125 y la Figura 39 ilustran cómo una aplicación de una versión anterior de IBM Business Process Manager se enlaza a un servicio externo. Para aplicaciones desarrolladas para IBM Business Process Manager Versión 7.0, se utiliza la Especificación de activación en lugar del Puerto receptor y de la Fábrica de conexiones.

### Características clave de un enlace WebSphere MQ:

Entre las características clave de un enlace WebSphere MQ se incluyen las cabeceras, los artefactos Java EE y los recursos Java EE creados.

### Esquemas de correlación

Una aplicación de petición/respuesta de WebSphere MQ puede utilizar una de las diferentes técnicas para correlacionar los mensajes de respuesta con las peticiones, creadas alrededor de los campos de MQMD MessageID y CorrelID. En la gran mayoría de los casos, el solicitante permite que el gestor de colas seleccione un MessageID y espera que la aplicación de respuesta lo copie en el CorrelID de la respuesta. En muchos casos, la aplicación de petición y de respuesta sabe de forma implícita qué técnica de correlación se ha de utilizar. Ocasionalmente, la aplicación de respuesta reconocerá los diferentes distintivos del campo Report de la petición que describe cómo se manejan estos campos.

Los enlaces de exportación para los mensajes WebSphere MQ se pueden configurar con las opciones siguientes:

### Opciones del MsgId de respuesta:

**Nuevo MsgID**

Permite al gestor de colas seleccionar un MsgId exclusivo para la respuesta (valor por omisión).

**Copiar de MsgID de petición**

Copia el campo MsgId desde el campo MsgId de la petición.

**Copiar del mensaje SCA**

Establece el MsgId que se ha de transportar en las cabeceras de WebSphere MQ en el mensaje de respuesta de SCA o deja que el gestor de colas defina un nuevo Id si el valor no existe.

**Opciones como Informe**

Inspecciona el campo Report del MQMD de la petición para obtener una sugerencia acerca de cómo manejar MsgId. Las opciones MQRO\_NEW\_MSG\_ID y MQRO\_PASS\_MSG\_ID están soportadas y se comportan como Nuevo MsgId y Copiar de MsgID de petición, respectivamente.

**Opciones del CorrelId de respuesta:****Copiar de MsgID de petición**

Copia el campo CorrelId desde el campo MsgId de la petición (valor por omisión).

**Copiar de CorrelID de petición**

Copia el campo CorrelId desde el campo CorrelId de la petición.

**Copiar del mensaje SCA**

Establece el CorrelId que se ha de transportar en las cabeceras de WebSphere MQ en el mensaje de respuesta SCA o lo deja en blanco si el valor no existe.

**Opciones como Informe**

Inspecciona el campo Report del MQMD de la petición para obtener una sugerencia acerca de cómo manejar CorrelId. Las opciones MQRO\_COPY\_MSG\_ID\_TO\_CORREL\_ID y MQRO\_PASS\_CORREL\_ID están soportadas y se comportan como Copiar desde y Copiar de CorrelID de petición, respectivamente.

Los enlaces de importación para los mensajes WebSphere MQ se pueden configurar con las opciones siguientes:

**Opciones del MsgId de respuesta:****Nuevo MsgID**

Permite al gestor de colas seleccionar un MsgId exclusivo para la petición (valor por omisión).

**Copiar del mensaje SCA**

Establece el MsgId que se ha de transportar en las cabeceras de WebSphere MQ en el mensaje de solicitud de SCA o deja que el gestor de colas defina un nuevo Id si el valor no existe.

**Opciones del correlación de respuesta:****La respuesta tiene el CorrelID copiado de MsgId**

Espera que el mensaje de respuesta tenga un campo CorrelId establecido como el MsgId de la petición (por omisión).

**La respuesta tiene el MsgID copiado de MsgId**

Espera que el mensaje de respuesta tenga un campo MsgId establecido como el MsgId de la petición.

**La respuesta tiene el CorrelID copiado de CorrelID**

Espera que el mensaje de respuesta tenga un campo CorrelId establecido como el CorrelId de la petición.

**Recursos Java EE**

Se crea una serie de recursos Java EE cuando se despliega un enlace WebSphere MQ en un entorno Java EE.

## Parámetros

### Fábrica de conexiones MQ

Utilizada por los clientes para crear una conexión con el proveedor MQ de WebSphere.

### Fábrica de conexiones de respuesta

Utilizada por el tiempo de ejecución SCA de MQ cuando el destino de envío está en un gestor de colas diferente que el destino de recepción.

### Especificación de activación

Una especificación de activación JMS se asocia con uno o más beans controlados por mensajes y proporciona la configuración necesaria para que ellos reciban mensajes.

### Destinos

- Destino de envío: donde se envía la petición o el mensaje de salida (importación); donde se enviará el mensaje de respuesta (exportación), si no se reemplaza con el campo de cabecera ReplyTo en el mensaje de entrada.
- Destino de recepción: donde se debe colocar el mensaje de respuesta/petición o de entrada.

### Cabeceras de WebSphere MQ:

Las cabeceras WebSphere MQ incorporan determinados convenios para la conversión de mensajes SCA (Service Component Architecture).

Un mensaje WebSphere MQ está formado por una cabecera de sistema (el MQMD), cero o más cabeceras MQ adicionales (del sistema o personalizadas), y un cuerpo de mensaje. Si existen varias cabeceras de mensajes en el mensaje, el orden de las cabeceras es importante.

Cada cabecera contiene información que describe la estructura de la siguiente cabecera. El MQMD describe la primera cabecera.

### Cómo se analizan las cabeceras MQ

Para analizar las cabeceras MQ se utiliza un enlace de datos de cabecera MQ. Se da soporte, de forma automática, a las cabeceras siguientes:

- MQRFH
- MQRFH2
- MQCIH
- MQIIH

Las cabeceras que empiezan con **MQH** se manejan de forma distinta. Los campos específicos de la cabecera no se analizan; permanecen como bytes no analizados.

Para otras cabeceras MQ, puede escribir enlaces de datos de cabecera MQ personalizados para analizar aquéllas.

### Cómo se accede a las cabeceras MQ

Se puede acceder a las cabeceras MQ del producto en una de dos maneras:

- A través del objeto de mensaje de servicio (SMO) en una mediación.
- A través de la API ContextService.

Las cabeceras MQ se representan internamente con el elemento MQHeader de SMO. MQHeader es un contenedor de los datos de cabecera que amplía MQControl pero que contiene un elemento de valor de anyType. Contiene el MQMD, MQControl (la información de control del cuerpo del mensaje MQ), y una lista de otras cabeceras MQ.

- MQMD representa los contenidos de la descripción del mensaje WebSphere MQ, excepto para la información que determina la estructura y la codificación del cuerpo.
- MQControl contiene información que determina la estructura y la codificación de un cuerpo de mensaje.
- Las MQHeader contienen una lista de objetos MQHeader.

La cadena de la cabecera MQ se despliega de forma que, dentro del SMO, cada cabecera MQ incluye su propia información de control (CCSID, codificación y formato). Las cabeceras pueden añadirse o suprimirse fácilmente, sin por ello alterar otros datos de la cabecera.

### Definición de campos en MQMD

Puede actualizar MQMD mediante la API Contexto o mediante el objeto de mensaje de servicio (SMO) en una mediación. Los campos siguientes se propagan automáticamente al mensaje MQ de salida:

- Cifrado
- Juego de caracteres codificado
- Formato
- Notificar
- Caducidad
- Comentarios
- Prioridad
- Persistencia
- ID de correlación
- Indicadores de mensajes

Configure el enlace MQ en una Importación o Exportación para propagar las siguientes propiedades al mensaje MQ de salida:

#### MsgID

Establezca el **ID de mensaje de solicitud** en copiar desde el mensaje SCA.

#### MsgType

Borre el recuadro de selección **Establecer tipo de mensaje en MQMT\_DATAGRAM o MQMT\_REQUEST para la operación de solicitud-respuesta.**

#### ReplyToQ

Deseleccione el cuadro de selección **Alterar temporalmente la respuesta de una cola del mensaje de solicitud.**

#### ReplyToQMgr

Deseleccione el cuadro de selección **Alterar temporalmente la respuesta a la cola de mensaje de solicitud.**

A partir de la versión 7.0, los campos de contexto pueden alterarse temporalmente mediante una propiedad personalizada en la definición de destino JNDI. Establezca la propiedad personalizada MDCTX con el valor SET\_IDENTITY\_CONTEXT en el destino de envío para propagar los campos siguientes al mensaje MQ de salida:

- UserIdentifier
- AppIdentityData

Establezca la propiedad personalizada MDCTX con el valor SET\_ALL\_CONTEXT en el destino de envío para propagar las propiedades siguientes al mensaje MQ de salida:

- UserIdentifier
- AppIdentityData

- PutApplType
- PutApplName
- ApplOriginData

Algunos campos no se propagan en el mensaje MQ de salida. Los campos siguientes se alteran temporalmente durante el envío del mensaje:

- BackoutCount
- AccountingToken
- PutDate
- PutTime
- Offset
- OriginalLength

#### **Adición estática de MQCIH a un enlace WebSphere MQ:**

IBM Business Process Manager soporta la adición de datos de cabecera MQCIH de forma estática sin utilizar un módulo de mediación.

Hay varias maneras de añadir información de cabecera MQCIH a un mensaje (por ejemplo, utilizando la primitiva de mediación del método setter de cabecera). Es posible que resulte útil añadir esta información de cabecera estáticamente, sin utilizar un módulo de mediación adicional. La información de cabecera estática, incluido el nombre de programa CICS, el ID de transacción y otros detalles de cabecera de formato de datos, se puede definir y añadir como parte del Enlace de WebSphere MQ.

WebSphere MQ, el puente MQ CICS y CICS deben estar configurados para que se añada estáticamente los datos de cabecera de MQCIH.

Puede utilizar Integration Designer para configurar la importación de WebSphere MQ con valores estáticos que son necesarios para la información de cabecera MQCIH.

Cuando llega un mensaje y lo procesa la importación de WebSphere MQ, se realiza una comprobación para ver si la información de cabecera MQCIH ya está presente en el mensaje. Si MQCIH existe, se utilizan los valores estáticos definidos en la importación de WebSphere MQ para alterar temporalmente los valores dinámicos correspondientes en el mensaje. Si MQCIH no existe, se crea una en el mensaje y se añaden los valores estáticos definidos en la importación de WebSphere MQ.

Los valores estáticos definidos en la importación de WebSphere MQ son específicos de un método. Puede especificar diferentes valores MQCIH estáticos para diferentes métodos en la misma importación de WebSphere MQ.

Este recurso no se utiliza para proporcionar valores por omisión si MQCIH no contiene información de cabecera específica porque un valor estático definido en la importación de WebSphere MQ alterará temporalmente un valor correspondiente proporcionado en el mensaje de entrada.

#### **Cientes externos:**

IBM Business Process Manager puede enviar mensajes o recibirlos de clientes externos mediante los enlaces WebSphere MQ.

Un cliente externo (por ejemplo, un portal Web o un sistema de empresa) puede enviar un mensaje a un componente SCA de la aplicación mediante una exportación, o lo puede invocar un componente SCA de la aplicación mediante una importación.

El enlace de exportación de WebSphere MQ despliega los beans controlados por mensajes (MDB) para escuchar las peticiones procedentes del destino especificado en receive especificado en el enlace de exportación. El destino especificado en el campo send se utiliza para enviar la respuesta a la petición de entrada si la aplicación invocada proporciona una respuesta. De esta forma, un cliente externo puede invocar aplicaciones a través del enlace de exportación.

Las importaciones WebSphere MQ se enlazan con clientes externos y pueden entregar mensajes a éstos. Este mensaje puede necesitar o no una respuesta del cliente externo.

Puede encontrar más información sobre cómo interactuar con clientes externos utilizando WebSphere MQ en el Centro de información de WebSphere MQ.

### **Resolución de problemas de enlaces WebSphere MQ:**

Puede diagnosticar y solucionar las anomalías y las condiciones de error que se producen con los enlaces WebSphere MQ.

### **Condiciones de error principales**

Las condiciones de error principales de los enlaces WebSphere MQ quedan determinadas por la semántica de transacciones, por la configuración de WebSphere MQ o por la referencia a un comportamiento existente de otros componentes. Las primeras condiciones de error son:

- No se puede conectar a la cola o al gestor de colas de WebSphere MQ.  
Si no se puede conectar a WebSphere MQ para recibir mensajes, no se podrá iniciar el puerto receptor MDB. Esta condición se registrará en el archivo de registro de WebSphere Application Server. Los mensajes persistentes permanecerán en la cola de WebSphere MQ hasta que se hayan recuperado correctamente (o hayan caducado mediante WebSphere MQ).  
Si no se puede conectar a WebSphere MQ, para enviar mensajes de salida, se provocará la retrotracción de la transacción que controla el envío.
- No se puede analizar un mensaje de entrada o construir un mensaje de salida.  
Una anomalía en el enlace de datos provoca la retrotracción de la transacción que controla el trabajo.
- No se puede enviar el mensaje de salida.  
Una anomalía al enviar un mensaje provoca una retrotracción de la transacción correspondiente.
- Varios mensajes de respuesta o mensajes inesperados.  
La importación espera sólo un mensaje de respuesta para cada mensaje de solicitud. Si llega más de una respuesta o si llega una respuesta con demora (una para la que se ha pasado el tiempo de caducidad de respuesta de SCA), se genera una excepción de tiempo de ejecución de servicio. Se retrotrae la transacción y el mensaje de respuesta se vuelve a poner fuera de la cola o lo gestiona el Gestor de sucesos anómalos.

### **Escenarios de utilización incorrecta: comparación con los enlaces de JMS de WebSphere MQ**

La importación y exportación de WebSphere MQ se han diseñado principalmente para interoperar con aplicaciones WebSphere MQ nativas, y exponer el contenido completo del cuerpo del mensaje WebSphere MQ en las mediaciones. El enlace de JMS de WebSphere MQ, sin embargo, se ha diseñado para interoperar con las aplicaciones JMS desplegadas en WebSphere MQ, que expone los mensajes de acuerdo con el modelo de mensaje JMS.

Los escenarios siguientes se deben crear utilizando el enlace JMS de WebSphere MQ, no el enlace de WebSphere MQ:

- Invocación de un bean controlado por mensaje JMS (MDB) desde un módulo SCA, donde el MDB se despliega en el proveedor de JMS de WebSphere MQ. Utilice una importación de JMS de WebSphere MQ.

- Permitir que se invoque el módulo SCA desde un servlet de componentes Java EE o desde EJB por medio de JMS. Utilice una exportación de JMS de WebSphere MQ.
- Mediación de los contenidos de un MapMessage JMS, en tránsito entre WebSphere MQ. Utilice una exportación y una importación de JMS de WebSphere MQ junto con el enlace de datos adecuado.

Existen situaciones en las cuales es posible que se espera la interoperatividad entre el enlace WebSphere MQ y el enlace JMS de WebSphere MQ. En particular, cuando realiza un puente entre aplicaciones Java EE y no Java EE de WebSphere MQ, utilice una exportación de WebSphere MQ y una importación de JMS de WebSphere MQ (o viceversa) junto con los enlaces de datos apropiados o los módulos de mediación (o ambos).

### **Mensajes sin entregar**

Si WebSphere MQ no puede entregar un mensaje a su destino previsto (por ejemplo, debido a errores de configuración), envía los mensajes en su lugar a una cola de mensajes no entregados nominada.

Al hacerlo, se añade una cabecera de mensaje en espera al inicio del cuerpo del mensaje. Esta cabecera contiene las razones de la anomalía, el destino original y otra información.

### **Mensajes SCA basados en MQ que no aparecen en el gestor de sucesos anómalos**

Si los mensajes SCA originados debido a una anomalía en una interacción WebSphere MQ, los mensajes deberán aparecer en el gestor de sucesos anómalos. Si dichos mensajes no aparecen en el gestor de sucesos anómalos, asegúrese de que el destino WebSphere MQ tenga un valor máximo de entregas con error mayor que 1. Si establece este valor en 2 o más, se permite la interacción con el gestor de sucesos anómalos durante las invocaciones de SCA para los mensajes WebSphere.

### **Los sucesos anómalos de MQ se reproducen en un gestor de colas erróneo**

Cuando debe utilizarse una fábrica de conexiones predefinida para las conexiones de salida, las propiedades de conexión deben coincidir con las definidas en la especificación de activación utilizada para las conexiones de entrada.

La fábrica de conexiones predefinidas se utiliza para crear una conexión al reproducir un suceso anómalo y, por lo tanto, debe configurarse para utilizar el mismo gestor de colas del que se recibió el mensaje originalmente.

### **Manejo de excepciones:**

El modo en que está configurado el enlace de datos determina cómo se tratan las excepciones generadas por los manejadores de datos o los enlaces de datos. Además, la naturaleza del flujo de mediación dicta el comportamiento del sistema cuando se genera una excepción de este estilo.

Pueden surgir varios problemas cuando su enlace llama a un manejador de datos o enlace de datos. Por ejemplo, un manejador de datos puede recibir un mensaje que tiene una carga útil dañada, o puede tratar de leer un mensaje que tenga un formato incorrecto.

El modo en que el enlace gestiona una excepción de este estilo viene determinado por el modo de implementar el manejador de datos o el enlace de datos. El comportamiento recomendado es diseñar su propio enlace de datos para producir una excepción **DataBindingException**.

La situación es similar para un manejador de datos. Dado que el enlace de datos invoca el manejador de datos, cualquier excepción del manejador de datos se incluye en una excepción de enlace de datos. Por lo tanto, se le notifica una excepción **DataHandlerException** como una **DataBindingException**.

Cuando se produce cualquier excepción de tiempo de ejecución, incluida la excepción **DataBindingException**:

- Si el flujo de mediación se ha configurado para ser transaccional, el mensaje JMS se almacena en el gestor de sucesos anómalos de forma predeterminada para una respuesta o supresión manual.

**Nota:** Puede cambiar la modalidad de recuperación en el enlace, de forma que el mensaje se retrotrae, en lugar de almacenarse en el gestor de sucesos anómalos.

- Si el flujo de mediación no es transaccional, el flujo de mediación se anota y se pierde el mensaje.

La situación es similar para un manejador de datos. Puesto que el enlace de datos llama al manejador de datos, se produce una excepción de manejador de datos dentro de una excepción de enlace de datos. Por lo tanto, se informa de una excepción **DataHandlerException** como una **DataBindingException**.

## Limitaciones de los enlaces

Los enlaces tienen algunas limitaciones por lo que respecta a su uso que se listan aquí.

### Limitaciones del enlace MQ:

El enlace MQ tiene algunas limitaciones por lo que respecta a su uso que se listan aquí.

### Sin distribución de mensajes publicación-suscripción

El enlace MQ no admite actualmente el método de publicación-suscripción para la distribución de mensajes aunque WMQ admite el método de publicación-suscripción. No obstante, el enlace JMS de MQ sí que admite este método de distribución.

### Colas de recepción compartidas

Diversos enlaces de exportación e importación WebSphere MQ esperan que los mensajes presentes en su cola de recepción configurada sean para esa exportación o importación. Los enlaces de importación y exportación deberán configurarse con las siguientes consideraciones:

- Cada importación MQ debe tener una cola de recepción distinta ya que el enlace de importación MQ presupone que todos los mensajes de la cola de recepción son respuestas a peticiones que envió. Si más de una importación comparte la cola de recepción, una importación incorrecta podría recibir las respuestas y no podrían correlacionarse con el mensaje de solicitud original.
- Cada exportación MQ debería tener una cola de recepción distinta ya que, de lo contrario, no puede prever a qué exportación llegará un mensaje de solicitud concreto.
- Las importaciones y las exportaciones MQ pueden señalar a la misma cola de envío.

### Limitaciones de los enlaces JMS, MQ JMS y JMS genéricos:

Los enlaces JMS y MQ JMS tienen algunas limitaciones.

### Implicaciones de la generación de enlaces predeterminados

Las limitaciones del uso de los enlaces JMS, MQ JMS y JMS genéricos se exponen en las secciones siguientes:

- Implicaciones de la generación de enlaces predeterminados
- Esquema de correlación de respuesta
- Soporte de idioma bidireccional

Cuando genere un enlace, se llenarán varios campos automáticamente como valores predeterminados, si no decide especificar los valores usted mismo. Por ejemplo, se creará automáticamente el nombre de una fábrica de conexiones. Si sabe que pondrá su aplicación sobre un servidor y accederá a ella remotamente

con un cliente, al crear el enlace debería especificar nombres JNDI en lugar de aceptar los valores predeterminados, ya que probablemente querrá controlar estos valores mediante la consola de administración en tiempo de ejecución.

No obstante, si ha aceptado los valores predeterminados y luego se encuentra con que no puede acceder a su aplicación desde un cliente remoto, puede utilizar la consola de administración para definir explícitamente el valor de la fábrica de conexiones. Localice el campo de puntos finales del proveedor en los valores de la fábrica de conexiones y añada un valor como, por ejemplo `<nombre_host_servidor>:7276` (si utiliza el número de puerto predeterminado).

### Esquema de correlación de respuesta

Si utiliza el esquema de correlación de respuesta de ID de correlación a ID de correlación, que se utiliza para correlacionar mensajes en una operación de solicitud-respuesta, debe disponer de un ID de correlación dinámica en el mensaje.

Para crear un ID de correlación dinámica en un módulo de mediación utilizando el editor de flujos de mediación, añada un nodo XSLT antes de realizar la importación con el enlace JMS. Abra el editor de correlaciones XSLT. Las cabeceras de arquitectura de componentes de servicio conocidos estarán disponibles en el mensaje de destino. Arrastre y suelte un campo que contenga un ID exclusivo en el mensaje de origen al ID de correlación de la cabecera JMS del mensaje de destino.

### Soporte de idioma bidireccional

Java Naming and Directory Interface (JNDI) sólo admite caracteres ASCII para los nombres en tiempo de ejecución.

### Colas de recepción compartidas

Diversos enlaces de exportación e importación esperan que los mensajes presentes en su cola de recepción configurada sean para esa exportación o importación. Los enlaces de importación y exportación se deben configurar con las siguientes consideraciones:

- Cada enlace de importación debe tener una cola de recepción distinta ya que el enlace de importación presupone que todos los mensajes de la cola de recepción son respuestas a peticiones que envió. Si más de una importación comparte la cola de recepción, una importación incorrecta podría recibir las respuestas y no podrían correlacionarse con el mensaje de solicitud original.
- Cada exportación debería tener una cola de recepción distinta ya que, de lo contrario, no puede prever a qué exportación llegará un mensaje de solicitud concreto.
- Las importaciones y las exportaciones pueden señalar a la misma cola de envío.

---

## Objetos de negocio

La industria del software ha desarrollado varios modelos de programación e infraestructuras en los que los *objetos de negocio* proporcionan una representación natural de los datos de negocio para el proceso de aplicaciones.

En general, estos objetos de negocio tienen estas características:

- Están definidos de acuerdo con un estándar industrial
- Correlacionan de forma transparente datos con tablas de base de datos o sistemas de información empresarial
- Soportan la utilización de protocolos de invocación remota
- Proporcionan la base del modelo de programación de datos para la programación de aplicaciones

Desde la perspectiva de las herramientas, Integration Designer proporciona a los desarrolladores un modelo de objetos de negocio común para representar diferentes clases de entidades de negocio pertenecientes a diversos dominios. Durante el desarrollo, este modelo permite a los desarrolladores definir objetos de negocio como definiciones de esquema XML.

Durante la ejecución, los datos de negocio definidos por las definiciones de esquema XML se representan como objetos de negocio Java. En este modelo, los objetos de negocio están parcialmente basados en versiones preliminares de la especificación SDO (Service Data Object) y proporcionan el conjunto completo de interfaces de aplicación del modelo de programación necesarias para manipular datos de negocio.

## Definición de objetos de negocio

Los objetos de negocio se definen utilizando el editor de objetos de negocio en Integration Designer. El editor de objetos de negocio almacena los objetos de negocio como definiciones de esquema XML.

Utilizar el esquema XML para definir objetos de negocio proporciona varias ventajas:

- Los esquemas XML proporciona un modelo de definición de datos basado en estándares y una base para la interoperatividad entre sistemas y aplicaciones heterogéneos dispares. Los esquemas XML se utilizan conjuntamente con Web Services Description Language (WSDL) para proporcionar contratos de interfaz basados en estándares entre componentes, aplicaciones y sistemas.
- Los esquemas XML definen un modelo de definición de datos ricos para representar datos de negocio. Este modelo incluye tipos complejos, tipos simples, tipos definidos por el usuario, herencia de tipos y cardinalidad, entre otras características.
- Los objetos de negocio pueden definirse mediante interfaces empresariales y datos definidos en Web Services Description Language, así como mediante esquemas XML de organizaciones estándar del sector, o de otros sistemas y aplicaciones. Integration Designer puede importar estos objetos de negocio directamente.

Integration Designer también proporciona soporte para descubrir datos de negocio en bases de datos y sistemas de información empresarial y generando, a continuación, la definición de objeto de negocio de esquema XML basado en estándares de esos datos de negocio. Los objetos de negocio generados de esta manera suelen conocerse como *objetos de negocio específicos de la aplicación* ya que imitan la estructura de los datos de negocio definidos en el sistema de información empresarial.

Cuando un proceso manipula datos de muchos sistemas de información distintos, puede resultar útil transformar las distintas representaciones de datos de negocio (por ejemplo, CustomerEIS1 y CustomerEIS2 u OrderEIS1 y OrderEIS2) en una única representación canónica (por ejemplo, Cliente o Pedido). A la representación canónica se la suele conocer como *objeto de negocio genérico*.

Las definiciones de objeto de negocio, concretamente para objetos de negocio genéricos, las utiliza frecuentemente más de una aplicación. Para dar soporte a esta reutilización, Integration Designer permite que se creen objetos de negocio en bibliotecas que entonces pueden asociarse con varios módulos de aplicación.

Los contratos para los servicios proporcionados y consumidos por un módulo de aplicación de Service Component Architecture (SCA), así como los contratos utilizados para crear los componentes dentro de un módulo de aplicación se definen mediante el Web Services Description Language. Un WSDL puede representar tanto las operaciones como los objetos de negocio, que un esquema XML define para representar los objetos de negocio, de un contrato.

## Utilización de objetos de negocio

Service Component Architecture (SCA) proporciona la infraestructura para definir un módulo de aplicación, los servicios que proporciona, los servicios que consume y la composición de los componentes que proporcionan la lógica empresarial del módulo de aplicación. Los objetos de negocio juegan un papel

importante en la aplicación, definiendo los datos de negocio que se utilizan para describir los contratos de servicio y componente y los datos de negocio que los componentes manipulan.

El siguiente diagrama representa un módulo de aplicación SCA e ilustra muchos de los lugares donde el desarrollador trabaja con objetos de negocio.

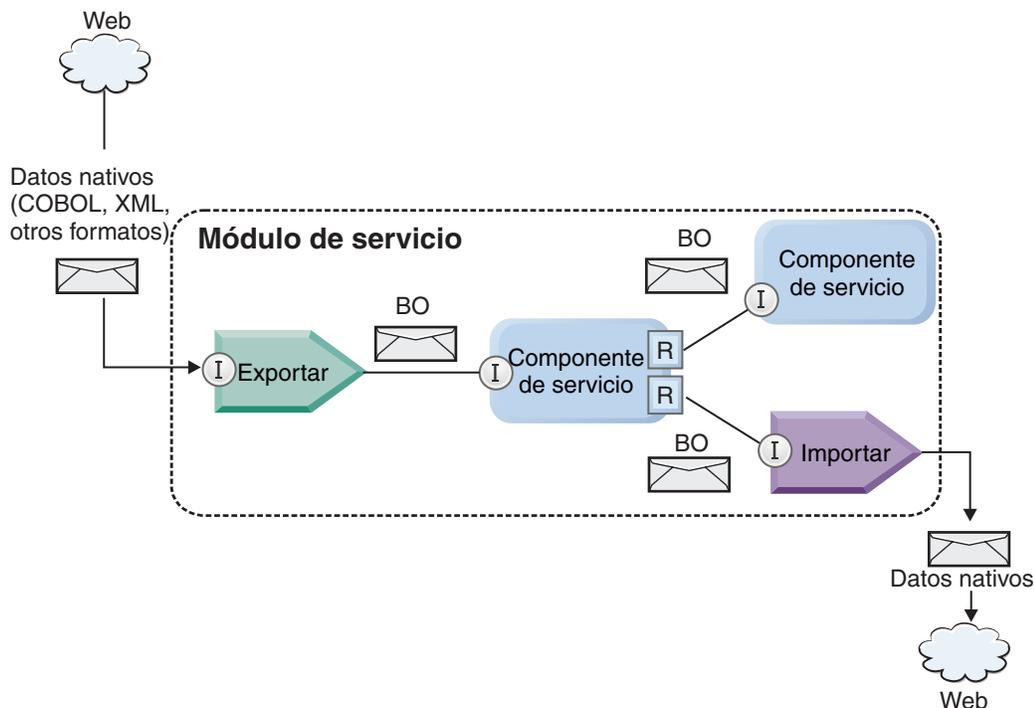


Figura 40. Los objetos de negocio representan los datos que fluyen entre los servicios de una aplicación

**Nota:** En este tema se describe cómo los módulos de aplicación SCA utilizan los objetos de negocio. Si está utilizando interfaces Java los módulos de aplicación SCA también pueden procesar objetos Java.

## Modelo de programación de objetos de negocio

El modelo de programación de objetos de negocio consta de un conjunto de interfaces Java que representan:

- La definición de objeto de negocio y los datos de la instancia
- Un conjunto de servicios que dan soporte a las operaciones en los objetos de negocio

Las definiciones de tipo de objeto de negocio están representadas por las interfaces `commonj.sdo.Type` y `commonj.sdo.Property`. El modelo de programación de objetos de negocio proporciona un conjunto de reglas para correlacionar la información de tipo complejo del esquema XML con la interfaz `Type` y cada uno de los elementos de la definición de tipo complejo con la interfaz `Property`.

Las instancias de objetos de negocio están representadas por la interfaz `commonj.sdo.DataObject`. El modelo de programación de objetos de negocio no tiene tipo, lo que significa que puede utilizarse la misma interfaz `commonj.sdo.DataObject` para representar distintas definiciones de objetos de negocio, tales como `Customer` y `Order`. La definición de qué propiedades pueden establecerse y recuperarse de cada objeto de negocio viene determinada por la información de tipo definida en el esquema XML asociado con cada objeto de negocio.

El comportamiento del modelo de programación de objetos de negocio está basado en la especificación Service Data Object 2.1. Para obtener más información, consulte la especificación SDO 2.1 para Java, guías de aprendizaje y javadocs de la web: <http://osoa.org/display/Main/Service+Data+Objects+Specifications>.

Los servicios de objetos de negocio dan soporte a diversas operaciones de ciclo de vida (tales como creación, igualdad, análisis y serialización) en los objetos de negocio.

Para conocer detalles sobre el modelo de programación de objetos de negocio, consulte Programación mediante servicios de objeto de negocio y Paquete com.ibm.websphere.bo.

## **Enlaces, enlaces de datos y manejadores de datos**

Como se muestra en Figura 40 en la página 136, los datos de negocio que se utilizan para invocar servicios proporcionados por módulos de aplicación SCA se transforman en objetos de negocio para que los componentes SCA puedan manipular los datos de negocio. De forma similar, los objetos de negocio manipulados por componentes SCA se convierten al formato de datos que necesitan los servicios externos.

En algunos casos, como en el enlace de servicio web, el enlace utilizado para servicios de exportación e importación transforma automáticamente los datos al formato adecuado. En otros casos, como en el enlace JMS, los desarrolladores pueden proporcionar un enlace de datos o un manejador de datos que convierte los formatos no nativos a objetos de negocio representados por la interfaz DataObject.

Para obtener más información sobre el desarrollo de enlaces de datos y manejadores de datos, consulte “Manejadores de datos” en la página 41 y “Enlaces de datos” en la página 43.

## **Componentes**

Los componentes SCA definen sus contratos de servicio de suministro y consumo mediante una combinación de Lenguaje de descripción de servicios web y esquemas XML. Los datos de negocio que SCA pasa entre componentes están representados como objetos de negocio mediante la interfaz DataObject. SCA verifica que estos tipos de objetos de negocio son compatibles con el contrato de interfaz definido por el componente a invocar.

Las abstracciones del modelo de programación para manipular objetos de negocio varían de componente a componente. El componente POJO y la primitiva Personalizada del componente de flujo de mediación proporciona la manipulación directa de los objetos de negocio al habilitar la programación Java directamente utilizando las interfaces y servicios de programación de objetos de negocio. La mayoría de los componentes proporcionan abstracciones de nivel superior para manipular objetos de negocio, pero también proporcionan fragmentos de código Java para definir comportamiento personalizado en las interfaces y servicios de objeto de negocio.

Los objetos de negocio pueden transformarse utilizando la combinación de los componentes Mediación de flujo de interfaz y Correlación de objetos de negocio, o el componente de flujo de mediación y su primitiva Correlación XML. Estas prestaciones de transformación de objetos de negocio son de utilidad para convertir objetos de negocio específicos de la aplicación a objetos de negocio genéricos y viceversa.

## **Objetos empresariales especiales**

Los objetos de mensaje de servicio y los gráficos de negocio son dos tipos especializados de objetos de negocio que se utilizan para finalidades específicas de las aplicaciones.

## Objeto de mensaje de servicio

Un objeto de mensaje de servicio (SMO) es un objeto de negocio especializado que los componentes de flujo de mediación utilizan para representar la colección de datos asociados con una invocación de servicio.

Un SMO tiene una estructura fija de nivel superior que consta de cabeceras, contexto, cuerpo y adjuntos (si los hay).

- Las cabeceras llevan información relacionada con la invocación de servicio por un protocolo o enlace concreto. Algunos ejemplos son las cabeceras SOAP y las cabeceras JMS.
- Los datos de contexto llevan información lógica adicional asociada con la invocación mientras el componente de flujo de mediación los procesa. Esta información no suele formar parte de los datos de aplicación enviados o recibidos por clientes.
- El cuerpo del SMO lleva los datos de negocio de carga útil, que representan el mensaje de aplicación central o datos de invocación en forma de un objeto de negocio estándar.

El SMO también puede llevar datos de adjuntos para invocaciones del servicio web que utilicen SOAP con adjuntos.

Los flujos de mediación realizan tareas tales como direccionamiento de peticiones y transformación de datos, y el SMO proporciona la vista combinada de contenido de cabecera y carga útil en una sola estructura unificada.

## Gráfico de negocio

Un gráfico de negocio es un objeto de negocio especial utilizado para proporcionar soporte para la sincronización de datos en escenarios de integración.

Fíjese en este ejemplo en el que dos sistemas de información empresarial tienen una representación de un pedido específico. Cuando el pedido cambia en un sistema, puede enviarse un mensaje al otro sistema para sincronizar los datos del pedido. Los gráficos de negocio dan soporte al concepto de enviar solamente la parte del pedido que ha cambiado al otro sistema y anotarlo con información de resumen de cambios para definir el tipo de cambio.

En este ejemplo, un gráfico de negocio Pedido indicaría al otro sistema que uno de los elementos de línea del pedido se ha suprimido y que se ha actualizado la propiedad de fecha de envío prevista del pedido.

Los gráficos de negocio pueden añadirse fácilmente a objetos de negocio existentes en Integration Designer. Se encuentran con más frecuencia en los casos en los que se utilizan adaptadores WebSphere y para dar soporte a la migración de aplicaciones de WebSphere InterChange Server.

## Modalidad de análisis de objetos de negocio

Integration Designer proporciona una propiedad en módulos y bibliotecas que puede utilizar para configurar la modalidad de análisis de XML para objetos de negocio como diligente (*eager*) o lento (*lazy*).

- Si la opción se establece como *diligente*, las corrientes de bytes XML se analizan rápidamente para crear el objeto de negocio.
- Si la opción se establece como *lento*, el objeto de negocio se crea normalmente, pero el análisis de la corriente de bytes XML se aplaza y se analiza parcialmente sólo cuando se accede a las propiedades del objeto de negocio.

En cualquiera de las dos modalidades de análisis de XML, los datos no XML siempre se analizan rápidamente para crear el objeto de negocio.

## Consideraciones cuando se selecciona la modalidad de análisis de objetos de negocio

La modalidad de análisis del objeto de negocio determina cómo se analizan los datos XML en tiempo de ejecución. La modalidad de análisis de un objeto de negocio se define en un módulo o biblioteca cuando se crea. Puede cambiar la modalidad de análisis del módulo o la biblioteca, sin embargo, debe ser consciente de las implicaciones.

La versión de la infraestructura de tiempo de ejecución del objeto de negocio se establece a nivel de módulo y biblioteca. Los módulos que se crearon en una versión de IBM Integration Designer anterior a la versión 7 se ejecutarán en la modalidad de análisis diligente sin cambios necesarios. De forma predeterminada, a los módulos y las bibliotecas que se crean en IBM Integration Designer Integration Developer versión 7 y versiones posteriores se les asignará la modalidad de análisis más adecuada, en función de diversos factores tales como la modalidad de análisis de proyectos existentes en su espacio de trabajo, o la modalidad de análisis de proyectos dependientes u otros proyectos de la misma solución, y así sucesivamente. Puede cambiar la modalidad de análisis de objetos de negocio de un módulo o biblioteca para que se adapte a su implementación, sin embargo debe tener en cuenta las siguientes consideraciones.

### Consideraciones

- La modalidad de análisis de objeto de negocio lenta procesa los datos XML más rápidamente; sin embargo existen diferencias de compatibilidad entre el modo diligente y la modalidad lenta que se deben tener en cuenta antes de cambiar la configuración de un módulo o biblioteca. Estas diferencias afectarán al comportamiento de ejecución de los módulos. Para obtener información sobre la modalidad de análisis que sea óptima para su aplicación, consulte Ventajas del uso de la modalidad de análisis lento frente al diligente
- Un módulo sólo se puede configurar para que se ejecute en una modalidad de análisis. Las bibliotecas se pueden configurar para dar soporte a una de las modalidades de análisis o a ambas. Una biblioteca que está configurada para dar soporte a ambas modalidades de análisis puede ser referenciada tanto por un módulo utilizando la modalidad de análisis diligente como por un módulo utilizando la modalidad de análisis lento. La modalidad de análisis de una biblioteca en tiempo de ejecución viene determinada por los módulos que hacen referencia a la biblioteca. En tiempo de ejecución, un módulo declara su modalidad de análisis y esa es la modalidad de análisis que utilizan el módulo y las bibliotecas que el módulo utiliza.
- Los módulos y bibliotecas que están configurados para modalidades de análisis distintas son compatibles en los casos siguientes:
  - Los módulos y bibliotecas configurados con la modalidad de análisis lento son compatibles con las bibliotecas que utilizan la modalidad de análisis lenta o ambas modalidades, la diligente y la lenta.
  - Los módulos y bibliotecas configurados con la modalidad de análisis diligente son compatibles con las bibliotecas que utilizan la modalidad de análisis diligente o ambas modalidades, la diligente y la lenta.
  - Las bibliotecas configuradas con las modalidades de análisis lenta y diligente son compatibles sólo con las bibliotecas que utilizan ambas modalidades, la lenta y la diligente.
- Utilice la misma infraestructura para interactuar con los módulos que se comunican mediante el enlace SCA. Si los módulos se comunican mediante modalidades de análisis distintas, se pueden producir problemas de rendimiento.

### Ventajas del uso de la modalidad de análisis lento (lazy) frente al diligente (eager)

Algunas aplicaciones se benefician de la modalidad de análisis de XML lento mientras que otras ven una mejora del rendimiento con la modalidad de análisis diligente. Es recomendable probar su aplicación en ambas modalidades de análisis para determinar qué modalidad de ajusta mejor a las características específicas de la aplicación.

La sección siguiente proporcione directrices generales sobre los tipos de aplicaciones que se benefician de cada tipo de modalidad de análisis:

- Aplicaciones que se benefician de la modalidad de análisis XML lento  
Las aplicaciones que analizan secuencias de datos XML de gran tamaño son las que tienen más probabilidades de ver mejoras en el rendimiento cuando se utiliza la modalidad de análisis XML lento. Las ventajas de rendimiento aumentan a medida que aumenta el tamaño de la corriente de bytes XML y disminuye la cantidad de datos de la corriente de bytes a que accede la aplicación.

**Nota:** La modalidad de análisis retardada para objetos de negocio está soportada en WebSphere Process Server versión 7.0.0.3 y versiones posteriores y en IBM Process Server. Los módulos y módulos de mediación que incluyen componentes de flujo de mediación no están soportados.

- Aplicaciones que se benefician de la modalidad de análisis rápida  
Las siguientes aplicaciones seguramente funcionarán mejor en modalidad de análisis diligente:
  - Aplicaciones que analizan corrientes de datos no XML
  - Aplicaciones que se crean mediante el servicio BOFactory
  - Aplicaciones que analizan mensajes XML muy pequeños

## Consideraciones sobre la migración de aplicaciones y el desarrollo

Si está configurando una aplicación que originalmente se desarrolló utilizando la modalidad de análisis diligente para ahora utilizar la modalidad de análisis diferida, o si piensa conmutar una aplicación entre las modalidades de análisis diligente y diferida, tenga en cuenta las diferencias entre las modalidades y las consideraciones al cambiar de modalidad.

## Manejo de errores

Si la corriente de bytes XML que se analiza está mal formada, se producen excepciones de análisis.

- En modalidad de análisis XML diligente, estas excepciones se producen en cuanto se analiza el objeto de negocio desde la corriente XML entrante.
- Si se ha configurado la modalidad de análisis XML lento, las excepciones de análisis se producen de forma latente cuando se accede a las propiedades de objetos de negocio y se analiza la parte del XML que está mal formada.

Para tratar XML mal formado, seleccione una de las opciones siguientes:

- Despliegue un bus de servicio empresarial en los límites para validar el XML entrante
- Cree una lógica de detección de errores lenta en el punto en que se accede a las propiedades de objetos de negocio

## Pilas de excepciones y mensajes

Debido a que las modalidades de análisis XML diligente y lento tienen distintas implementaciones subyacentes, los seguimientos de pila emitidos por las interfaces de programación y servicios de objetos de negocio tienen el mismo nombre de clase de excepción, pero podrían no contener el mismo mensaje de excepción o conjunto de clases de excepción específicas de la implementación.

## Formato de serialización XML

La modalidad de análisis XML lento proporciona una optimización del rendimiento que intenta copiar XML no modificado desde la corriente de bytes de entrada a la corriente de bytes de salida al realizarse la serialización. El resultado es un aumento del rendimiento, pero el formato de serialización de la corriente de bytes XML de salida podría ser distinto si el objeto de negocio entero se actualiza en modalidad de análisis XML lento o si se estaba ejecutando en modalidad de análisis XML diligente.

Aunque es posible que el formato de serialización de XML no sea el equivalente exacto sintácticamente, el valor de semántica proporcionado por el objeto de negocio es equivalente independientemente de las modalidades de análisis, y se puede pasar XML con seguridad entre aplicaciones que se ejecuten en modalidades de análisis distintas con equivalencia semántica.

## Validador de instancias de objetos de negocio

El validador de instancias de modalidad de objeto de negocio de análisis XML lento proporciona una validación de objetos de negocio de mayor fidelidad, especialmente la validación de facetas de valores de propiedades. Debido a estas mejoras, el validador de instancias de modalidad de análisis lento detecta problemas adicionales que no se detectan en la modalidad de análisis diligente y proporciona unos mensajes de error más detallados.

## Correlaciones XML de la versión 602

Los flujos de mediación desarrollados originalmente antes de WebSphere Integration Developer Versión 6.1 podrían contener primitivos XSLT que utilicen una correlación o una hoja de estilo que no se pueda ejecutar directamente en modalidad de análisis XML lento. Cuando se migra una aplicación para su uso en modalidad de análisis XML lento, el asistente de migración puede actualizar los archivos de correlación asociados con primitivas XSLT automáticamente para que se ejecuten en la nueva modalidad. No obstante, si una primitiva XSLT hace referencia directamente a una hoja de estilo que se ha editado manualmente, no se migra la hoja de estilo y no se puede ejecutar en modalidad de análisis XML lento.

## APIs privadas no publicadas

Si una aplicación aprovecha las interfaces de programación de objetos de negocio no publicadas, privadas y específicas de la implementación, es muy probable que la aplicación no consiga compilar cuando se cambie de modalidad de análisis. En modalidad de análisis diligente, estas interfaces privadas suelen ser clases de implementación de objetos de negocio definidas por la infraestructura de modelado de Eclipse (EMF).

En todos los casos, se recomienda encarecidamente que se eliminen de la aplicación las API privadas.

## APIs EMF de objeto de mensaje de servicio

Un componente de mediación en IBM Integration Designer proporciona la capacidad de manipular el contenido de mensajes utilizando las clases e interfaces Java proporcionadas en el paquete `com.ibm.websphere.sibx.smobo`. En modalidad de análisis XML lento, las interfaces Java del paquete `com.ibm.websphere.sibx.smobo` pueden utilizarse igualmente, pero lo más probable es que fallen los métodos que hacen referencia directamente a clases e interfaces de la infraestructura de modelado de Eclipse (EMF) o que se heredan de interfaces EMF.

El `ServiceMessageObject` y su contenido no puede convertirse a objetos EMF en modalidad de análisis XML lento.

## Servicio BOMode

El servicio BOMode se utiliza para determinar si la modalidad de análisis XML que se ejecuta actualmente es diligente o lento.

## Migración

Todas las aplicaciones anteriores a la versión 7.0.0.0 se ejecutan en modalidad de análisis XML diligente. Cuando se migran en tiempo de ejecución mediante las herramientas de migración en tiempo de ejecución BPM, continúan ejecutándose en modalidad de análisis XML diligente.

Para habilitar una aplicación anterior a la versión 7.0.0.0 para que se configure para utilizar la modalidad de análisis XML lento, primero utilice Integration Designer para migrar los artefactos de la aplicación. Después de la migración configurará la aplicación para que utilice el análisis XML lento.

Consulte [Migración de artefactos de origen](#) para obtener información sobre la migración de artefactos en Integration Designer, y consulte [Configuración de la modalidad de análisis de objetos de negocio de módulos y bibliotecas](#) para obtener información sobre cómo establecer la modalidad de análisis.

