

# Nombre del documento

<Descripción del diseño de software>

Representante

<nombre del representante>

Fechas de aprobación:

<dd/mm/yyyy>

## **INTRODUCCION**

*Define brevemente el alcance y propósito del software, provee la definición de los términos usados dentro del contexto del software, provee un marco de trabajo para entender los SDDs en el contexto de su preparación y uso, describe los requerimientos y organización de un SDD, define varios puntos de vista del diseño para su uso en la producción de SDDs, provee la bibliografía utilizada, contiene la plantilla para organizar un SDD conforma a los requerimientos.*

## **NOTAS PARA LOS USUARIOS**

### **Copyrights**

*Restricciones de uso, distribución, modificación, incluye leyes y regulaciones y el uso en una regulación privada*

### **Patentes**

*Identificar las patentes esenciales que requieran una licencia o una validación legal. Definir el alcance de las patentes y determinar los términos, condiciones y acuerdos de licencia.*

### **Participantes**

*Debe contener a todos los participantes dentro del Software, separados por secciones de trabajo.*

## Índice

### Contenidos

1. Resumen
  - 1.1 Alcance
  - 1.2 Propósito
  - 1.3 Audiencia meta
  
2. Definiciones
  
3. Modelo conceptual para la descripción del diseño de software
  - 3.1 Contexto del diseño de software
  - 3.2 Descripción del diseño de software dentro del ciclo de vida
  
4. Información contenida dentro de la descripción del diseño
  - 4.1 Introducción
  - 4.2 Identificación de SDD
  - 4.3 Diseño de involucrados y sus intervenciones
  - 4.4 Vistas de diseño
  - 4.5 Elementos del diseño
  - 4.6 Diseño de lenguajes
  
5. Puntos de vista del diseño
  - 5.1 Introducción
  - 5.2 Punto de vista del contexto
  - 5.3 Punto de vista de la composición
  - 5.4 Punto de vista de la lógica
  - 5.5 Punto de vista de las dependencias
  - 5.6 Punto de vista de la interfaz
  - 5.7 Punto de vista de la estructura
  - 5.8 Punto de vista de la interacción
  - 5.9 Punto de vista del algoritmo
  - 5.10 Punto de vista de los recursos

### Bibliografía

### Anexos

## Contenidos

### 1. Resumen

#### 1.1 Alcance

*Describe los diseños del software y establece la información contenida y organización de SDD (software design description). Describe la metodología para el diseño, la gestión de la configuración y la aseguración de la calidad*

#### 1.2 Propósito

*Especifica los requisitos para la sección de diseño de lenguajes que va a ser usado en el SDD y los requisitos de documentación de los puntos de vista del diseño a ser usados*

#### 1.3 Audiencia meta

*Se define la audiencia a técnicos y a los stakeholders que utilizan el SDD. Para que el diseñador lo tome en cuenta en la selección, organización y presentación del diseño de la información, asegurando que las descripciones sean completas, concisas, consistentes, apropiadas, bien organizadas y fáciles de comunicar.*

### 2. Definiciones

*En esta sección se hace referencia a todos los tecnicismos utilizados a través de este documento para su fácil comprensión.*

### 3. Modelo conceptual para la descripción del diseño de software

#### 3.1 Contexto del diseño de software

*Se describe la conceptualización del diseño (se muestra el software en la fase de diseño), donde se aprecian las características esenciales y se muestra un medio básico, para satisfacer la necesidades del cliente, así mismo sirve para análisis, evaluación y orientación del software. Se muestran los elementos del sistema, subsistemas, componenetes, librerías, frameworks, API's, diseño y descripción de la base de datos, descripción de interfaces, descripción de los elementos del sistema. Se suele acompañar de los estándares IEEE Std 1016-1998, IEEE Std 1471, IEEE Std 12207-2008*

#### 3.2 Descripción del diseño de software dentro del ciclo de vida

*Se describe el diseño de varias situaciones en las cuales el SDD puede ser creado y usado, se acompaña del siguiente documento: IEEE Std 12207-2008, en esta sección la clave es el SRS en el cual se capturan los requerimientos del software, para el que se puede utilizar el siguiente documento IEEE Std 830™-1998. También se diseña la verificación y validación del software, acompañada de los siguientes documentos para su apoyo IEEE Std 1028™-2008 y IEEE Std 12207-2008*

## **4. Información contenida dentro de la descripción del diseño**

### **4.1 Introducción**

*Identificación del SDD, del diseño de los stakeholders. Selección de los puntos de vista, con la definición del diseño de los elementos y del lenguaje utilizado. Diseño de vistas y diseño de overlays*

### **4.2 Identificación de SDD**

*Incluye la siguiente descripción de información:  
Fecha de problemas y estatus, alcance, organización emisora, autorship, referencias, context, cuerpo, resumen y glosario*

### **4.3 Diseño de involucrados y sus intervenciones**

*Se identifican a los involucrados (stakeholders) y sus intervenciones*

### **4.4 Vistas de diseño**

*Se pueden organizar las vistas en uno o más grupos, el objetivo de esta sección es permitirle a los stakeholders centrarse en el diseño de los detalles desde una perspectiva específica y revelando los requerimientos  
Cada diseño debe ser especificado por un nombre, tema, descripción de entidades, atributos, relaciones, limitantes, métodos, aplicaciones*

### **4.5 Elementos del diseño**

*Incluye los elementos clave del diseño del software, como entidades que pueden ser sistemas, subsistemas, librerías, frameworks, plantillas, componentes, data stores, módulos, unidades, atributos de las entidades. Esta sección de acompaña del siguiente documento IEEE Std 1016-1998.*

### **4.6 Diseño de lenguajes**

*En el diseño de lenguajes se deben de soportar las siguientes definiciones:*

*Sintaxis y semántica bien definida*

*Un estándar disponible*

*Suelen incluirse los siguientes documentos IEEE Std 1320.1™-1998, ISO/IEC 13817-1:1996*

## **5 Puntos de vista del diseño**

### **5.1 Introducción**

*Define varios puntos de vista de diseño para el uso en el SDD. Ilustra la realización de estos diseños de puntos de vista en cuanto a la selección del diseño, como se relaciona con problemas de diseño de puntos de vista, y establece un lenguaje (notación y método) nombres neutrales para estos puntos de vista, se crear un resumen de los puntos de vista a través de una tabla que describa los diagramas*

## **5.2 Punto de vista del contexto**

*Provee un diseño del sistema con referencias y contenido explícito de los actores, usuarios y stakeholders que interactúan con el ambiente diseñado provee una perspectiva similar a la de un “black box”*

## **5.3 Punto de vista de la composición**

*Describe la forma en la que el diseño del software es estructurado y de las partes que lo constituyen así como de las funciones y roles establecidos. También se diseñan algunas de las posibles fallas o preocupaciones del sistema, subsistemas, componentes, etc.*

## **5.4 Punto de vista de la lógica**

En esta sección se preocupa por elaborar existencias, elementos de diseño, tipos de diseño y sus implementaciones como clases e interfaces con su estructura, relaciones, entidades, limitaciones y atributos. Suelen utilizarse diagramas de clases y objetos de UML

## **5.5 Punto de vista de las dependencias**

Especifica las relaciones de interconexión y acceso entre las entidades, estas relaciones incluyen el intercambio de información, orden, ejecución o parametrización de interfaces, también puede ayudar a los encargados de mantenimiento a asilar entidades causantes de fallas en el sistema. Suele acompañarse de diagramas de componentes, paquetes, que muestren las dependencias entre los sistemas de UML

## **5.6 Punto de vista de la interfaz**

*En esta sección se proporciona información a los diseñadores, programadores y testers de los medios para saber cómo utilizar correctamente los servicios prestados por el sistema. Esta descripción incluye los detalles externos e interfaces internas no previstas en el SRS*

## **5.7 Punto de vista de la estructura**

*En este punto se documenta la constitución interna y la organización de sistema o software en términos de elementos, componentes, entidades utilizando elementos de diseño como entidades, relaciones, atributos y limitantes. Suelen utilizar diagramas de estructura compuesta, de clases y paquetes de UML.*

## **5.8 Punto de vista de la interacción**

*Define las estrategias de interacción entre las entidades con respecto al porque, como y su nivel de seguridad, incluye una evaluación de las responsabilidades en las colaboraciones, algunos elementos de diseño que se utilizan son clases métodos, estados, eventos, timing y sincronización*

### **5.9 Punto de vista del algoritmo**

*Se incluye el diseño y la descripción detallada de las operaciones (métodos, funciones), también detalles internos y lógica del diseño de la entidad, suele acompañarse de “pseudo código”, tablas de decisión, diagramas de flujo en los cuales se incluye la secuencia de eventos, procesos, prioridades, pasos, ciclos y especificación de las tareas realizadas.*

### **5.10 Punto de vista de los recursos**

*En este punto se modelan o definen las características y utilización de recursos diseñados para el sistema o software, se modelan las entidades a utilizar su “performance” y su función, se puede hablar de recursos físicos (hardware) como dispositivos o lógicos (software) como librerías, SO, servicios y también de recursos de procesamientos como CPU, ciclos, memoria, buffers.*

### **Bibliografía**

*Se incluyen las bibliografías de los documentos utilizados*

### **Anexos**

*Documento de carácter informativo de ayuda al lector*