

Descripción del Diseño del Software

A continuación, encontrará los elementos requeridos para su entrega del documento de *Descripción del Diseño del Software*. Todos los párrafos del color de este texto son las instrucciones para completar cada sección. Estos deben ser removidos del documento a entregar.

Cada sección indica los elementos que debe contener. No altere el formato del documento (distribución de los títulos y secciones, tamaños, tipos de letra, espaciado, márgenes, etc.). Las figuras y tablas deben estar numeradas y etiquetadas con un tamaño de letra menor o igual a la del texto de este párrafo.

Consistencia y uniformidad del documento

Si bien el documento está dividido en secciones que tratan diferentes elementos del proceso de gestión del proyecto, todos estos elementos de alguna manera se relacionan unos con otros y es por ello importante poder conectarlos (usando referencias cruzadas a secciones o artefactos generados en otros procesos). De igual manera es importante que el documento sea claro y consistente para que pueda ser útil en la gestión adecuada de su proyecto. En ese sentido, es importante ser claro y conciso en la redacción.

Este documento no debe exceder en su totalidad las 25 páginas. El documento se puede alargar por referencias bibliográficas.

Portada

La primera página es la portada del documento. Incluye el nombre del proyecto, de la empresa, la fecha de entrega, la versión actual, un logo que identifique a la organización y el nombre de los autores del documento

Tamaño

1 página.

1 Historial de Cambios

Propósito

Describir brevemente los cambios que ha sufrido el documento, con el fin de llevar una adecuada administración de configuración.

Contenido

Una tabla que indique los cambios que ha sufrido el documento. Debe incluir, al menos: fecha del cambio, descripción del cambio y persona o personas que realizaron el cambio.

2 Resumen

Propósito

Esta sección también debe proveer una descripción general del contenido del documento, especificando su organización o distribución de forma que el lector pueda estructurar la información que encontrará en el documento.

Contenido

Descripción breve del contenido, propósito y alcance del proyecto. Descripción de la audiencia esperada del documento (para quién es importante leer este documento).

Tamaño sugerido

1/2 página un párrafo de 8 a 10 líneas máximo.

Referencias

[1], sección 5.

3 Tabla de Contenidos

Propósito

Que el lector encuentre rápidamente una sección específica del documento.

4 Lista de Figuras

Propósito

Que el lector encuentre rápidamente una figura específica del documento.

5 Lista de Tablas

Propósito

Que el lector encuentre rápidamente una tabla específica del documento.

6 Introducción

Propósito

Describir brevemente el alcance del documento y dar una breve explicación o resumen del mismo. Para ubicar al lector de los detalles del producto, en caso de no haber leído el SPMP o SRS.

Contenido

- Propósito del documento: describe la intención con la que se realiza el documento, es decir, se deben exponer las razones por las que es importante.
- Alcance del producto: Describir el producto de software sobre el cual se realiza el SDD, indicando su nombre, las funcionalidades que incluirá y su utilidad (objetivos, beneficios).

Tamaño sugerido

1 página

7 Arquitectura

Propósito

Que el desarrollador entienda la arquitectura del sistema, es decir, la definición de estructura y comportamiento del sistema con un alto nivel de abstracción.

Contenido

Ver subsecciones.

Referencias

[1], [2].

7.1 Vista lógica del sistema

Propósito

Que el desarrollador sepa cuáles son los componentes más gruesos del sistema, sin preocuparse cómo serán instalados en el hardware.

Contenido

- Diagrama de paquetes o de componentes que describa los principales componentes del software y sus interfaces, junto con texto o tablas que expliquen cada parte del diagrama.
- Si se utilizan algunos patrones arquitectónicos (por ejemplo, MVC o N-capas), éstos deberían ser evidentes en el diagrama.
- Cada paquete o componente debe ser descrito brevemente debajo de la Figura de Arquitectura.

El diagrama debe incluir como componentes todas las librerías, máquinas virtuales y cualquier otro software requerido por el sistema para su funcionamiento.

Tamaño sugerido

2 páginas

Referencias

[4], sección *The Development Architecture* y *The Logical Architecture*

7.2 Vista física del sistema

Propósito

Que el desarrollador sepa cuáles son los componentes físicos (hardware) más importantes del sistema e identifique dónde serán instalados los componentes de software.

Contenido

- Un diagrama de despliegue, junto con texto o tablas que expliquen cada parte del diagrama que no haya sido descrita en la sección 7.1.
- Este diagrama debe incluir los componentes que aparecen en la sección 7.1, (si usted utilizó un diagrama de paquetes, puede transformarlo en componentes) pero sin describir interfaces, sino enfatizando los lugares donde estarían instalados todos los componentes. Por ese motivo, algunos componentes como librerías, máquinas virtuales, etc., los cuales pueden ser instalados en múltiples computadores, podrían aparecer múltiples veces en este diagrama.
- Deben quedar claras las características mínimas de software y hardware de cada Nodo físico.

Tamaño sugerido

2 páginas

Referencias

[4], secciones *The Physical Architecture*.

7.3 Vista de procesos del sistema

Propósito

Que el desarrollador sepa cuáles son los principales procesos en los que el usuario interactúa con el sistema.

Contenido

- Seleccione las principales interacciones entre el actor y el sistema para ser modeladas.
- Represente a través de diagramas de actividades las principales acciones que los usuarios realizan en el sistema. Estos diagramas deben ser consistentes con el diagrama de navegabilidad de la interfaz gráfica. Los diagramas deben ir acompañados de texto o tablas que los expliquen.

Los diagramas a utilizar pueden ser de los siguientes tipos:

- Diagramas de actividad
- Diagrama BPMN
- Diagrama de estados (del sistema completo)

Puede utilizar más de un tipo de diagrama, pero sin ser redundante en la información presentada. Por ejemplo, evite hacer un diagrama de actividad y uno de estados que representen las mismas acciones.

Tamaño sugerido

3 páginas

Referencias

[4]

8 Diseño Detallado

Propósito

Mostrar los detalles más importantes del diseño de bajo nivel del sistema.

Contenido

Ver subsecciones

8.1 Estructura del sistema

Propósito

Que el desarrollador comprenda los detalles de la estructura de todos los componentes de software del sistema. El objetivo de la sección es proveer una guía para implementar las clases utilizadas en el software.

Contenido

- Para cada paquete o componente de software identificado, un diagrama de clases acompañado de texto o tablas que lo expliquen. Todas las clases, métodos y asociaciones deben estar explicados.
- Estos diagramas de clases pueden ser anexos del proyecto.

Tamaño sugerido

1 página

8.2 Comportamiento del sistema

Propósito

Que el desarrollador entienda los detalles de cómo cambia el sistema en el tiempo. El objetivo de la sección es proveer una guía para implementar los métodos más complejos de las clases del software.

Contenido

- Seleccione aquellas funcionalidades o métodos que constituyan el núcleo de la aplicación, o sean de alta complejidad. Deben estar en concordancia con la sección 6.3.

- Realice un diagrama de secuencia, de comunicación o de estados que describa cómo se realiza dicha acción. Dichos diagramas de secuencia deben utilizar instancias y métodos de las clases descritas en la sección 8.1.
- Incluya diagramas de secuencia sólo para aquellas acciones que sean suficientemente complejas. Por ejemplo, si una acción involucra sólo una o dos llamadas a métodos, no vale la pena describirla en un diagrama.
- No describa en estos diagramas lo que sucede dentro de las librerías que utilice y que no sean de su creación.

Tamaño sugerido

1 página

8.3 Persistencia

Propósito

Que el desarrollador entienda cómo se van a almacenar los datos del sistema en forma persistente. Proveer una guía para crear las bases de datos, archivos u otros medios de almacenamiento persistentes utilizados por el software.

Contenido

- Diagramas que describan la forma en que la información será almacenada, ya sea en bases de datos, archivos, etc.
- Los diagramas que se pueden usar son: diagramas E-R, diagramas relacionales o diagramas de clases con estereotipos [5].
- Los elementos de los diagramas deben ser descritos en párrafos o tablas.

Tamaño sugerido

3 páginas

8.4 Interfaz de usuario

Propósito

Que el desarrollador entienda los detalles de la interfaz de usuario.

Contenido

- Esta sección está unida a la descripción detallada de los CU realizada en el SRS.

- Incluir un diagrama que muestre el orden de navegación entre las pantallas. Un diagrama de navegación.
- Mencionar algunos componentes especiales de interfaz gráfica que se hayan utilizado, en caso de existir alguno. Algo diferente que valga la pena documentar (no un botón o un cuadro de texto).
- Mencionar algunos de los diseños de interfaz realizados en caso de que sean interesantes o tengan elementos complejos para el desarrollador.
- No poner pantallas reales de la aplicación.

Tamaño sugerido

3 páginas

9 Anexos

Agregue aquí cualquier información adicional relevante para el proyecto para que sea referenciada desde las secciones previas.

Es importante tener en cuenta que hasta antes de esta sección, el documento debe ser autocontenido, es decir, el lector no debe necesitar leer ningún documento adicional o archivo externo para entender el proyecto. La información que se coloque aquí es simplemente un complemento, en caso de que el lector requiriera más detalles sobre algún tema.

10 Referencias

Indique aquí todas las referencias bibliográficas utilizadas en el documento. Utilice formato IEEE o APA para definir las. Para administrar automáticamente las referencias, se recomienda el uso de la herramienta Zotero (<http://www.zotero.org>) o de JabRef (<http://www.jabref.org>).

Referencias usadas en el formato

[1] ISO/IEC/IEEE 42010:2011. Systems and software engineering - Architecture description.

[2] IEEE 1016:2009. Standard for Information Technology - Systems Design - Software Design Descriptions.

[3] RUP. Software Architecture Document.

http://sce.uhcl.edu/helm/RUP_school_example/wcsoftwareprocessweb/templates/ana_desi/pt_sad.htm

[4] Kruchten, P. Architectural Blueprints—The “4+1” View Model of Software Architecture.
<http://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf>

[5] Ambler, S. UML Data Modeling Profile.

<http://www.agiledata.org/essays/umlDataModelingProfile.html>

[6] Ambler, S. User Interface Flow Diagrams.

<http://www.agilemodeling.com/artifacts/uiFlowDiagram.htm>