# FUNDAMENTOS DE INGENIERÍA DE SOFTWARE

Tabla de contenido

[FUNDAMENTOS DE INGENIERÍA DE SOFTWARE 1](#_Toc147573517)

[1. Documentación de Requisitos 2](#_Toc147573518)

[1.1 Especificación de Requisitos: 2](#_Toc147573519)

[1.2 Registro de Entrevistas y Comunicación con el Cliente 2](#_Toc147573520)

[2. Documentación de Diseño 2](#_Toc147573521)

[2.1 Diseño de Arquitectura: 2](#_Toc147573522)

[3. Documentación de Código Fuente 2](#_Toc147573523)

[3.1 Comentarios en el Código: 2](#_Toc147573524)

[4. Documentación de Pruebas 2](#_Toc147573525)

[4.1 Planes de Pruebas: 2](#_Toc147573526)

[4.2 Registros de Pruebas: 2](#_Toc147573527)

[5. Documentación de Usuario 2](#_Toc147573528)

[5.1 Manuales de Usuario: 2](#_Toc147573529)

[6. Documentación de Proyecto 3](#_Toc147573530)

[6.1 Documentación de Proyecto General: 3](#_Toc147573531)

[6.2 Gestión de Cambios: 3](#_Toc147573532)

[6.3 Registro de Problemas: 3](#_Toc147573533)

## 1. Documentación de Requisitos

1.1 Especificación de Requisitos:

Este es uno de los documentos clave en la documentación del proyecto. Debe describir en detalle los requisitos funcionales y no funcionales del sistema. Incluye casos de uso, historias de usuario, diagramas de flujo y cualquier otro artefacto necesario para comprender las necesidades del cliente.

1.2 Registro de Entrevistas y Comunicación con el Cliente

Cualquier información obtenida del cliente durante reuniones, entrevistas o comunicación por correo electrónico debe ser registrada. Esto incluye preguntas, respuestas, aclaraciones y cambios en los requisitos. Puedes utilizar herramientas de seguimiento de problemas o sistemas de gestión de proyectos para esto.

## 2. Documentación de Diseño

2.1 Diseño de Arquitectura:

Describe la estructura del sistema, incluyendo componentes, módulos y su interacción. Puedes utilizar diagramas UML, esquemas, diagramas de flujo y descripciones narrativas.

## 3. Documentación de Código Fuente

3.1 Comentarios en el Código:

Cada pieza de código debe estar bien comentada para que los desarrolladores futuros puedan comprender su funcionamiento. Esto incluye comentarios sobre funciones, clases, variables y secciones importantes del código.

## 4. Documentación de Pruebas

### 4.1 Planes de Pruebas:

Describe cómo se llevarán a cabo las pruebas, qué se probará y cuáles serán los criterios de éxito.

4.2 Registros de Pruebas:

Mantén un registro de todas las pruebas realizadas, incluyendo los resultados y cualquier problema encontrado.

## 5. Documentación de Usuario

### 5.1 Manuales de Usuario:

Proporciona manuales que expliquen cómo utilizar el software desde la perspectiva del usuario final. Esto incluye instrucciones paso a paso y capturas de pantalla.

## 6. Documentación de Proyecto

6.1 Documentación de Proyecto General:

Incluye una descripción general del proyecto, su alcance, objetivos, cronograma y cualquier otro detalle relevante.

### 6.2 Gestión de Cambios:

Lleva un registro de los cambios realizados en el proyecto a lo largo del tiempo, incluyendo cambios en los requisitos, diseño y código.

### 6.3 Registro de Problemas:

Utiliza herramientas de seguimiento de problemas para registrar y gestionar problemas, errores y solicitudes de cambios.