|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CONTROL DE VERSIONES** | | | |  |  |
| Autor(es) | Fecha de modificación | Versión | Descripción del cambio | Revisó | Estado |
| LIDH  MFRC  JUZA | 27/10/2025 | 1.0 | Creación del Documento | MATH | APROBADO |
|  |  |  |  |  |  |

**Propósito**

Detallar el cómo los requisitos de software deben ser implementados y determinar si se han abordado todos los requisitos necesarios, también proporciona al equipo de desarrollo una orientación general sobre la arquitectura del proyecto de software y los modelos a seguir, facilita a identificar los elementos de diseño que no están alineados con un requerimiento, para determinar si el elemento de diseño es necesario; se convierte en la base para limitar los cambios en el alcance de un proyecto, ya que sirve como una guía durante toda la vida del sistema para los miembros del proyecto. El documento debe ser tan detallado como sea posible, a fin de mantener el equipo de desarrollo de software centrado y alineado al objetivo del sistema.

**Arquitectura del sistema**

Se mostrará la arquitectura del proyecto de trabajo terminal I, para lo cual se tomará en cuenta la definición de arquitectura del software que la IEEE std. 1471-2000, y que a la letra dice: *La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente, y los principios que orientan su diseño y evolución*.

La arquitectura de software se refiere a la “estructura general del software y las formas en la que la estructura proporciona una integridad conceptual para un sistema”. En su forma más simple, la arquitectura es la estructura u organización de los componentes del programa (módulos), la manera en que estos componentes interactúan; así como, la estructura de los datos que utilizan los componentes. La especificación de la arquitectura del software es importante porque:

* Las representaciones de la arquitectura del software permiten la comunicación entre todas las partes integrantes o participantes interesadas en el desarrollo de un sistema de cómputo.
* Destaca las decisiones iniciales relacionadas con el diseño que tendrán un impacto profundo en todo el trabajo de la Ingeniería de Software.
* Constituye un modelo relativamente pequeño e intelectualmente comprensible de cómo está estructurado el sistema y cómo trabajan juntos sus componentes.

**Diseño de la base de datos (cuando aplique).**

En caso de que el proyecto de trabajo terminal desarrollado utilice una base de datos, ésta se describirá en este apartado y deberá mostrarse la representación de la estructura de la base de datos (preferentemente gráfica).

**Manejo de archivos (cuando aplique).**

En caso de que el proyecto de trabajo terminal desarrollado utilice manejo de archivos, éste se describirá en este apartado. Deberá especificar:

* Estructura del contenido del/los archivo(s)
* Ruta relativa de la ubicación del/los archivo(s).

**Diagramas UML**

Se debe generar para el proyecto de trabajo terminal los diagramas en el Lenguaje Unificado de Modelado (UML), el cual describe los límites, la estructura y el comportamiento del sistema y los objetos que contiene, recordando que se debe hacer una descripción de cada diagrama construido.

Diagramas UML estructurales

* **Diagrama de clases**: Son la base principal de toda solución orientada a objetos. Las clases dentro de un sistema, atributos y operaciones, y la relación entre cada clase. Las clases se agrupan para crear diagramas de clases al crear diagramas de sistemas grandes.
* **Diagrama de componentes**: Muestra la relación estructural de los elementos del sistema de software, muy frecuentemente empleados al trabajar con sistemas complejos con componentes múltiples. Los componentes se comunican por medio de interfaces.
* **Diagrama de estructura compuesta:** Los diagramas de estructura compuesta se usan para mostrar la estructura interna de una clase.
* **Diagrama de implementación**: Ilustra el hardware del sistema y su software. Útil cuando se implementa una solución de software en múltiples máquinas con configuraciones únicas.
* **Diagrama de objetos**: Muestra la relación entre objetos por medio de ejemplos del mundo real e ilustra cómo se verá un sistema en un momento dado. Dado que los datos están disponibles dentro de los objetos, estos pueden usarse para clarificar relaciones entre objetos.
* **Diagrama de paquetes**: Hay dos tipos especiales de dependencias que se definen entre paquetes: la importación de paquetes y la fusión de paquetes. Los paquetes pueden representar los diferentes niveles de un sistema para revelar la arquitectura. Se pueden marcar las dependencias de paquetes para mostrar el mecanismo de comunicación entre niveles.

Diagramas UML de comportamiento

* **Diagrama de caso de uso:** Representa una funcionalidad particular de un sistema. Se crea para ilustrar cómo se relacionan las funcionalidades con sus controladores (actores) internos/externos**\***.

|  |  |
| --- | --- |
| Caso de uso | |
| Nombre |  |
| Actor participante |  |
| Objetivo |  |
| Disparador |  |
| Precondición |  |
| Postcondición |  |
| Flujo básico |  |
| Flujo alternativo 1 |  |
| Suposiciones |  |

* **Diagramas de actividades**: Flujos de trabajo de negocios u operativos representados gráficamente para mostrar la actividad de alguna parte o componente del sistema. Los diagramas de actividades se usan como una alternativa a los diagramas de máquina de estados.
* **Diagrama de comunicación**: Similar a los diagramas de secuencia, pero el enfoque está en los mensajes que se pasan entre objetos. La misma información se puede representar usando un diagrama de secuencia y objetos diferentes.
* **Diagrama de secuencia:** Muestra cómo los objetos interactúan entre sí y el orden de la ocurrencia. Representan interacciones para un escenario concreto.
* **Diagrama de máquina de estados:** Similar a los diagramas de actividades, describen el comportamiento de objetos que se comportan de diversas formas en su estado actual.
* **Diagrama de temporización:** Al igual que en los diagramas de secuencia, se representa el comportamiento de los objetos en un período de tiempo dado. Si hay un solo objeto, el diagrama es simple. Si hay más de un objeto, las interacciones de los objetos se muestran durante ese período de tiempo particular.

**Diseño de prototipos**

Para el desarrollo del proyecto de trabajo terminal se deben construir los prototipos, los cuales son medios de comunicación entre analistas, clientes o usuarios, que muestran las decisiones tomadas y permite dar resolución a los problemas de comprensión presentes en la etapa de levantamiento de requerimientos, ayudan al diseño y pueden ser usadas como una técnica de determinación y validación de requerimientos.