

Universidad Marino Gálvez de Guatemala

Ingeniería en Sistemas de Información y Ciencias de la Computación

TERCER AREA _ 379 ÁREA ANÁLISIS DISEÑO DE SISTEMAS

Documentación Técnica

Autor: **Juan Amilcar Larias Méndez**

Carné: **4490-19-9119**

Profesor: **Maria Fernanda Montenegro Monterroso**

Fecha: **30.11.2024**

Introducción

La presente documentación técnica detalla la arquitectura, los diagramas de flujo, los modelos de datos y otros aspectos esenciales del sistema de gestión de pruebas y control de calidad. Este sistema está diseñado para optimizar la planificación, ejecución y supervisión de pruebas en proyectos de software, garantizando la entrega de productos confiables y libres de errores.

Sus funcionalidades abarcan la gestión integral de proyectos, la planificación y ejecución sistemática de pruebas, la administración eficiente de defectos, la generación de informes detallados y métricas, así como la integración continua en el proceso de desarrollo. Los usuarios del sistema incluyen gestores de proyectos, ingenieros de calidad y desarrolladores, cada uno con roles y responsabilidades claramente definidos.

Para asegurar una comprensión uniforme de los términos y acrónimos utilizados en este documento, se proporciona un glosario que servirá como referencia a lo largo de la lectura. Esta introducción establece el contexto necesario para profundizar en los detalles técnicos del sistema en las secciones siguientes, facilitando su comprensión y aplicación efectiva.

1. Arquitectura del Sistema

El sistema de gestión de pruebas y control de calidad se compone de tres elementos fundamentales: el frontend, el backend y la base de datos. Cada uno desempeña un papel crucial en el funcionamiento integral del sistema.

1.1. Frontend (React JS)

La interfaz de usuario está desarrollada con React JS, una biblioteca de JavaScript que permite construir interfaces interactivas y eficientes. React facilita la creación de componentes reutilizables, mejorando la mantenibilidad y escalabilidad de la aplicación. Su arquitectura basada en componentes permite una separación clara de responsabilidades, facilitando el desarrollo y las pruebas de la interfaz.

Para el diseño y la estilización, se ha utilizado Tailwind CSS, un framework que proporciona clases de utilidad para construir interfaces personalizadas de manera eficiente. Tailwind CSS permite aplicar estilos directamente en el marcado HTML, agilizando el proceso de diseño y facilitando la creación de componentes consistentes y responsivos. La gestión de dependencias y scripts del proyecto se ha realizado con Yarn, un gestor de paquetes que ofrece rapidez y seguridad en la instalación de módulos, permitiendo un control preciso de las versiones de las dependencias y optimizando el flujo de trabajo del desarrollo.

Para el empaquetado y desarrollo del frontend, se ha empleado Vite, una herramienta de construcción que proporciona un servidor de desarrollo rápido y eficiente, con funcionalidades como la recarga en caliente de módulos. Vite utiliza módulos ES nativos y ofrece una experiencia de desarrollo ágil, reduciendo significativamente los tiempos de compilación y mejorando la productividad del equipo.

1.2. Backend (Node.js)

El servidor backend está implementado con Node.js, un entorno de ejecución de JavaScript en el lado del servidor. Node.js permite manejar múltiples conexiones de manera eficiente, esencial para aplicaciones de gran escala. Se emplea el framework Express.js para simplificar la creación de rutas y gestionar las solicitudes HTTP, estableciendo una comunicación fluida entre el frontend y la base de datos.

1.3. Base de Datos (MySQL)

Para el almacenamiento de datos, se utiliza MySQL, un sistema de gestión de bases de datos relacional de código abierto. MySQL es conocido por su facilidad de uso, escalabilidad y rendimiento, lo que lo convierte en una opción popular para aplicaciones web y empresariales. Su compatibilidad multiplataforma y su amplia comunidad de soporte facilitan su implementación y mantenimiento.

La integración de estos componentes se realiza mediante una arquitectura de cliente-servidor. El frontend envía solicitudes al backend, que a su vez interactúa con la base de datos para recuperar o almacenar información.

Esta estructura modular permite una fácil escalabilidad y mantenimiento del sistema, asegurando un rendimiento óptimo y una experiencia de usuario satisfactoria.

Para el desarrollo del frontend, se ha utilizado Tailwind CSS, un framework de CSS que proporciona clases de utilidad para construir interfaces de usuario personalizadas de manera eficiente. Tailwind CSS permite aplicar estilos directamente en el marcado HTML, lo que agiliza el proceso de diseño y facilita la creación de componentes consistentes y responsivos.

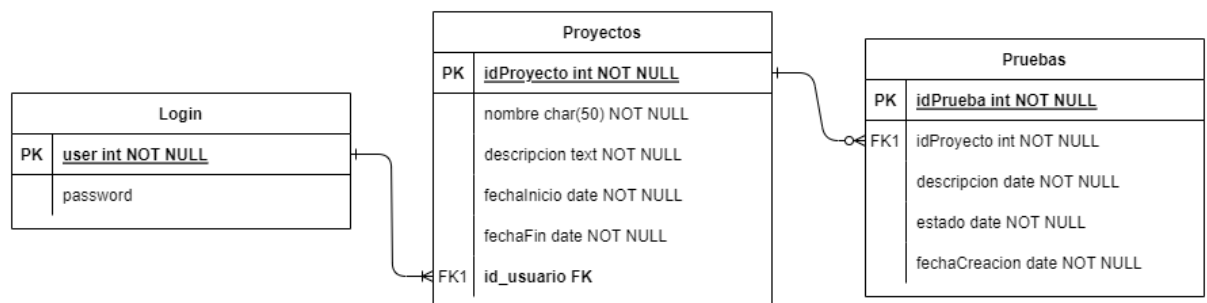
La gestión de dependencias y scripts del proyecto se ha realizado con Yarn, un gestor de paquetes que ofrece rapidez y seguridad en la instalación de módulos. Yarn permite un control preciso de las versiones de las dependencias y optimiza el flujo de trabajo del desarrollo.

Para el empaquetado y desarrollo del frontend, se ha empleado Vite, una herramienta de construcción que proporciona un servidor de desarrollo rápido y eficiente, con funcionalidades como la recarga en caliente de módulos. Vite utiliza módulos ES nativos y ofrece una experiencia de desarrollo ágil, reduciendo significativamente los tiempos de compilación y mejorando la productividad del equipo.

La combinación de estas tecnologías y herramientas permite un desarrollo ágil y eficiente del sistema, facilitando su mantenimiento y escalabilidad a largo plazo.

2. Modelado de Datos

Diagrama entidad-relación



3. Interfaz de Usuario

Prototipos de pantallas principale



A login form prototype centered on a light gray background. The form is a white rounded rectangle with a subtle drop shadow. It contains the following elements: a title 'Inicio de Sesión', a subtitle 'Ingresa tus credenciales para acceder al sistema.', a 'Username' label above a text input field with the placeholder 'Enter your username', a 'Password' label above a text input field with the placeholder 'Enter your password', and a black 'Login' button at the bottom.

Inicio de Sesión
Ingresa tus credenciales para acceder al sistema.

Username
Enter your username

Password
Enter your password

Login

Descripción de la navegación

Proyectos

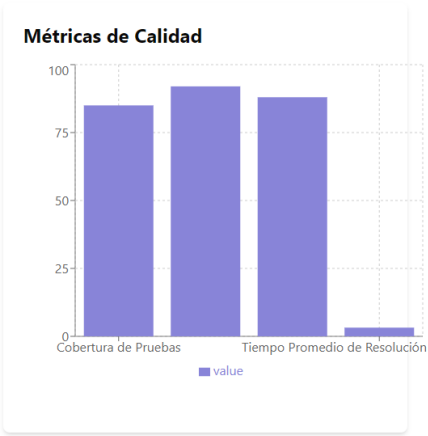
Nuevo Proyecto

| Nombre | Descripción | Fecha Inicio | Fecha Fin | Acciones |
|------------|--------------------------------------|--------------|------------|---------------------------|
| Biblioteca | Organiza los libros de la biblioteca | 2024-11-29 | 2024-12-29 | <div>EditarEliminar</div> |
| fdfdno | fdfdsss | 2024-11-08 | 2024-12-07 | <div>EditarEliminar</div> |
| nu | nu | 2024-11-20 | 2024-11-28 | <div>EditarEliminar</div> |

Pruebas

Nueva Prueba

| ID Proyecto | Descripción | Estado | Fecha Creación | Acciones |
|-------------|----------------|-----------|--------------------------|---------------------------|
| 1 | Prueba tecnica | Pendiente | 2024-11-30T05:02:17.000Z | <div>EditarEliminar</div> |



4. Integración Continua

Herramientas utilizadas

- Nodejs
- Reactjs
- MySQL
- Vite
- Tailwindsc
- Yarn

Proceso de integración y despliegue

- Hostinger
- Render

5. Seguridad

Para garantizar la seguridad en el acceso al sistema de gestión de pruebas y control de calidad, se ha implementado un mecanismo de autenticación y autorización basado en JSON Web Tokens (JWT). Este estándar abierto permite la transmisión segura de información entre el cliente y el servidor mediante la generación de tokens firmados digitalmente. Al autenticarse, el servidor emite un token JWT que el cliente debe incluir en las solicitudes subsecuentes, permitiendo al servidor verificar la identidad y los permisos del usuario de manera eficiente y sin necesidad de mantener el estado de la sesión en el servidor. Esta metodología no solo mejora la escalabilidad del sistema, sino que también refuerza la seguridad al minimizar la exposición de credenciales sensibles.

6. Anexos

- Autenticación: Proceso de verificación de la identidad de un usuario antes de otorgarle acceso al sistema.
- Autorización: Proceso que determina los permisos y niveles de acceso que un usuario autenticado tiene dentro del sistema.
- Backend: Parte del sistema que gestiona la lógica de negocio, el procesamiento de datos y la comunicación con la base de datos.
- Base de Datos Relacional: Sistema de almacenamiento de datos que organiza la información en tablas con relaciones definidas entre ellas.
- CI/CD (Integración Continua/Despliegue Continuo): Prácticas de desarrollo de software que automatizan la integración de cambios de código y su despliegue en producción.
- Cliente-Servidor: Arquitectura de red donde un cliente solicita servicios o recursos y un servidor los proporciona.
- CRUD (Crear, Leer, Actualizar, Eliminar): Operaciones básicas para el manejo de datos en una base de datos.
- Frontend: Parte del sistema que interactúa directamente con el usuario, presentando la interfaz y capturando las interacciones.
- JSON Web Token (JWT): Estándar abierto para la creación de tokens de acceso que permiten la transmisión segura de información entre partes.

- Node.js: Entorno de ejecución de JavaScript en el lado del servidor que permite construir aplicaciones escalables y de alto rendimiento.
- React JS: Biblioteca de JavaScript para construir interfaces de usuario de manera eficiente y modular.
- Tailwind CSS: Framework de CSS que proporciona clases de utilidad para construir interfaces de usuario personalizadas de forma rápida.
- Vite: Herramienta de desarrollo que ofrece un servidor de desarrollo rápido y eficiente, facilitando la construcción de aplicaciones frontend.
- Yarn: Gestor de paquetes que permite la instalación y gestión de dependencias en proyectos de JavaScript de manera rápida y segura.