

PROYECTO FINAL TÉCNICAS DE PRUEBAS DE SOFTWARE

INTEGRANTES:

LUIS EDUARDO LOPEZ GRISALES – 1859999-3743

JOAN MANUEL CALDERÓN GUTIÉRREZ – 1860241-3743

JUAN MANUEL URBANO TORRES - 1958634-3743

SANTIAGO GONZALEZ RODRIGUEZ - 1958714-3743

PROFESOR:

ING. JUAN PABLO PINILLOS REINA

TÉCNICAS DE PRUEBAS DE SOFTWARE

GRUPO: 51

UNIVERSIDAD DEL VALLE – SEDE TULUÁ

FACULTAD DE INGENIERÍA

2023

Diagrama de arquitectura

La arquitectura mostrada en la figura 1 es una arquitectura cliente-servidor. El cliente es el usuario, que interactúa con el sistema a través de un navegador web. El servidor es el componente que almacena y procesa los datos del sistema.

En este caso, el cliente utiliza la biblioteca ReactJS para renderizar la interfaz de usuario. La biblioteca ReactJS es una biblioteca de JavaScript que permite crear interfaces de usuario complejas y fáciles de mantener.

El servidor utiliza Firebase, una plataforma de backend como servicio (BaaS) que proporciona servicios como almacenamiento de datos, autenticación y seguridad. Firebase simplifica el desarrollo de aplicaciones backend, ya que proporciona una amplia gama de servicios listos para usar.

El flujo de datos del sistema es el siguiente:

- El usuario inicia la aplicación en su navegador web.
- La aplicación cliente envía una solicitud al servidor.
- El servidor procesa la solicitud y devuelve una respuesta al cliente.
- El cliente renderiza la respuesta en la interfaz de usuario.

La arquitectura cliente-servidor es una arquitectura común para aplicaciones web. Esta arquitectura permite separar la lógica de la aplicación entre el cliente y el servidor. Esto facilita el desarrollo y el mantenimiento de la aplicación, ya que cada componente puede ser desarrollado e implementado de forma independiente.

En este caso, la arquitectura cliente-servidor es adecuada para la aplicación, ya que permite al usuario interactuar con la interfaz de usuario de forma sencilla y segura. La biblioteca ReactJS proporciona una forma fácil de crear interfaces de usuario complejas, y Firebase proporciona una amplia gama de servicios que simplifican el desarrollo del backend.

A continuación se detallan los componentes individuales de la arquitectura:

Cliente

El cliente es el componente que interactúa con el usuario. En este caso, el cliente es un navegador web que utiliza la biblioteca ReactJS para renderizar la interfaz de usuario.

La biblioteca ReactJS es una biblioteca de JavaScript que permite crear interfaces de usuario complejas y fáciles de mantener. ReactJS utiliza un enfoque de componentes para la creación de interfaces de usuario. Los componentes son bloques de construcción reutilizables que se pueden combinar para crear interfaces de usuario complejas.

Servidor

El servidor es el componente que almacena y procesa los datos del sistema. En este caso, el servidor utiliza Firebase, una plataforma de backend como servicio (BaaS) que proporciona servicios como almacenamiento de datos, autenticación y seguridad.

Firebase simplifica el desarrollo de aplicaciones backend, ya que proporciona una amplia gama de servicios listos para usar. Por ejemplo, Firebase proporciona un servicio de almacenamiento de datos que permite almacenar datos en la nube. Firebase también proporciona un servicio de autenticación que permite a los usuarios registrarse y autenticarse en la aplicación.

Base de datos.

La base de datos es proporcionada por Firebase. Firebase proporciona un servicio de almacenamiento de datos que permite almacenar datos en la nube.

El servicio de almacenamiento de datos de Firebase es una base de datos NoSQL que admite varios tipos de datos, incluidos números, cadenas, objetos y arrays. El servicio de almacenamiento de datos de Firebase es escalable y puede manejar grandes volúmenes de datos.

Autenticación

La autenticación es el proceso de verificar la identidad de un usuario. En este caso, la autenticación es proporcionada por Auth0. Auth0 es un proveedor de servicios de identidad (IDP) que permite a las aplicaciones autenticar a los usuarios de forma segura y sencilla.

Auth0 utiliza un enfoque de tokens para la autenticación. Los tokens son credenciales que se utilizan para identificar a un usuario. Auth0 proporciona un servicio de generación de tokens que permite generar tokens para los usuarios.

Seguridad

La seguridad es la protección de los datos del sistema. En este caso, la seguridad es proporcionada por Firebase. Firebase proporciona una serie de funciones de seguridad que ayudan a proteger los datos del sistema, incluidas las siguientes:

Autenticación: el servicio de autenticación de Firebase ayuda a proteger los datos del sistema autenticando a los usuarios.

Encriptación: Firebase proporciona un servicio de encriptación que ayuda a proteger los datos del sistema en tránsito y en reposo.

Controles de acceso: Firebase proporciona controles de acceso que ayudan a restringir el acceso a los datos del sistema a los usuarios autorizados.

En general, la arquitectura mostrada en la imagen es una arquitectura cliente-servidor adecuada para la aplicación. La biblioteca ReactJS proporciona una forma fácil de crear

interfaces de usuario complejas, y Firebase proporciona una amplia gama de servicios que simplifican el desarrollo del backend.

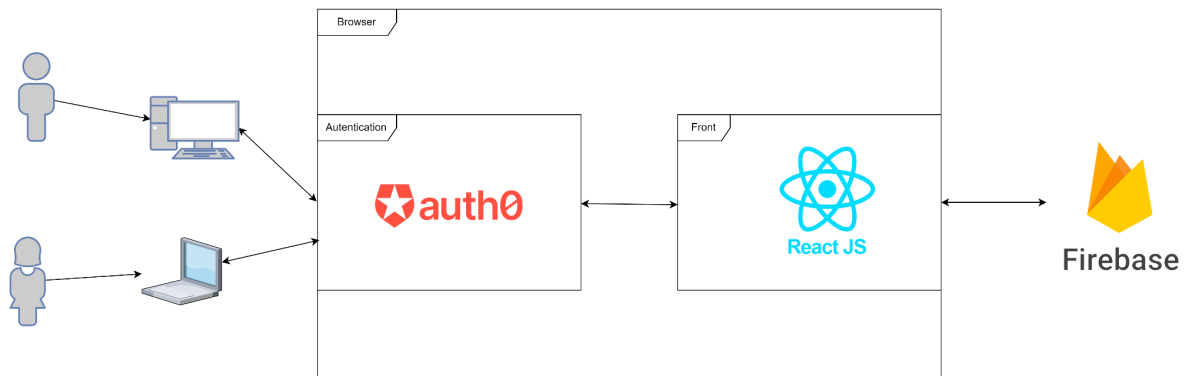


Figura 1.

Diagrama de Casos de uso

Actores

- Administrador: El administrador es responsable de gestionar el sistema y sus usuarios.
- Ingeniero de Pruebas: El ingeniero de pruebas es responsable de planificar, ejecutar y reportar pruebas de software.
- Desarrollador: El desarrollador es responsable de desarrollar el software que se está probando.

Casos de Uso

- Crear Proyecto: El administrador crea un nuevo proyecto para organizar las pruebas..
- Agregar Caso de Prueba: El ingeniero de pruebas agrega un nuevo caso de prueba al proyecto.
- Editar Caso de Prueba: El ingeniero de pruebas edita un caso de prueba existente.
- Eliminar Caso de Prueba: El ingeniero de pruebas elimina un caso de prueba existente.
- Registrar Resultados de Pruebas: El ingeniero de pruebas registra los resultados del conjunto de pruebas ejecutado.
- Generar Informe de Pruebas: El sistema genera un informe de pruebas que resume los resultados del conjunto de pruebas ejecutado.

