

Documentación

Proyecto Final UX

Semana: 10

Nombre de los integrantes:

Pamela Giselle Ramírez Rivera 12141141
Bryan Alexander Espinal Andino - 12141139
Diego André Molina Valladares – 12141157
Gerardo André Díaz Urraco - 12141095

Sede de estudio:

Tegucigalpa

Docente:

Ing. Claudia Cortes

Sección:

1305

Fecha de entrega:

Sábado 16 de diciembre de 2023

FrontEnd

Para la creación de componentes en nuestro proyecto utilizamos la librería de UI llamada Tamagui en su versión más nueva (v1.79.12).

Documentación oficial de Tamagui

<https://tamagui.dev/docs/intro/introduction>

También hicimos uso de Expo en la versión 48.0.7 para poder correr la app desde nuestro teléfono, además de hacer uso de otras funciones incluidas en Expo.

Documentación oficial de Expo

<https://docs.expo.dev/tutorial/introduction/>

Para la conexión con el backend utilizamos Axios en la versión 1.6.2 o mayor.

Documentación oficial de Axios

<https://axios-http.com/docs/intro>

9:17 AM 34%

Inicio de Sesión

¡Bienvenido de vuelta!

Correo

Contraseña

¿Mantener sesión iniciada?

☒

Iniciar Sesión

¿No Tienes Una Cuenta?

Registrate

9:17 AM 34%

Registro

Bienvenido!!

Nombre

Apellido

Usuario

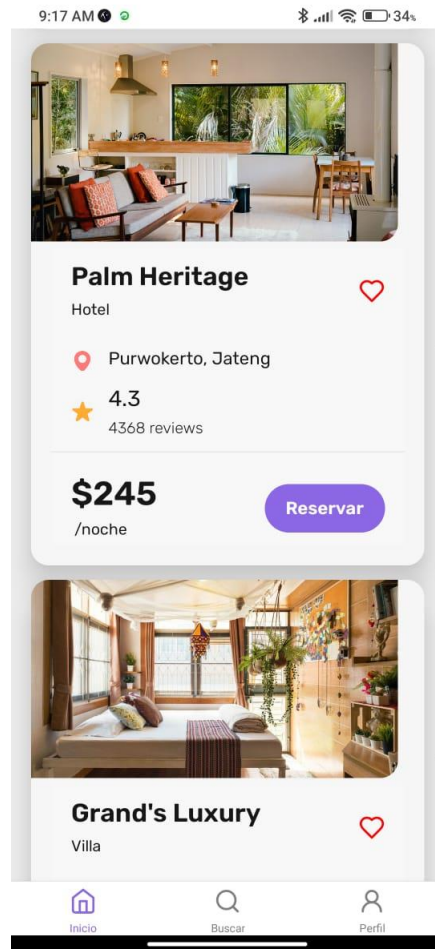
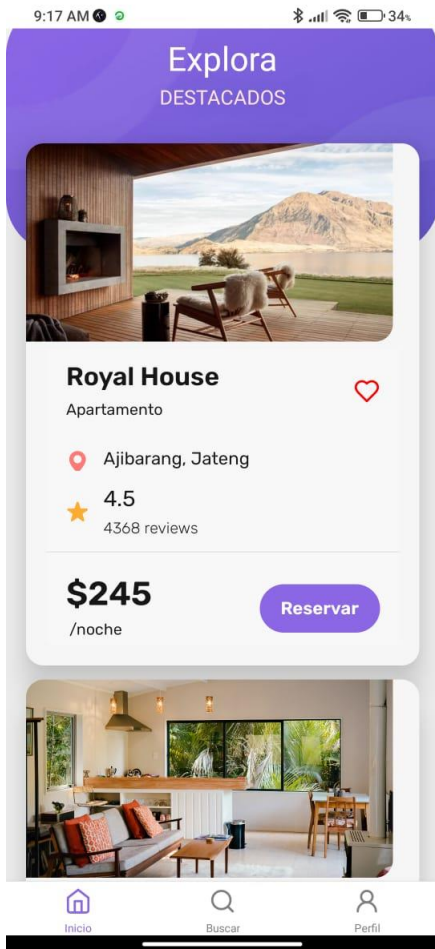
Correo

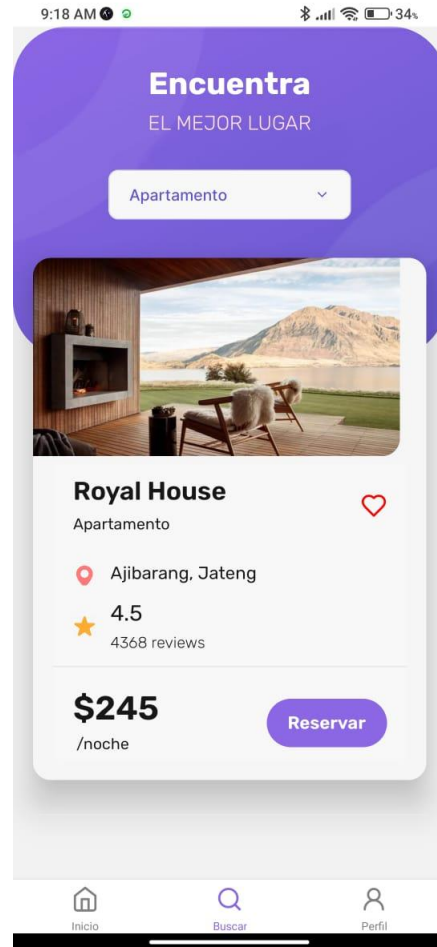
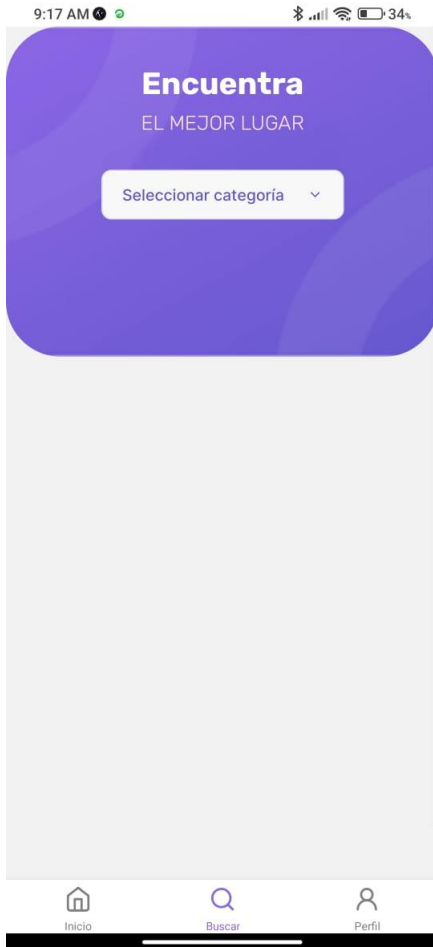
Contraseña

Aceptas Nuestros Terminos Y Condiciones?

☒

Registrarse





9:31



Cerrar Sesión

Iniciar sesión desde otra cuenta



Inicio



Buscar



Perfil

BackEnd

Para el desarrollo del backend, se utilizo mongoDb como base de datos principal. Librerias importantes para el desarrollo del mismo como cors, express, body-parser. Como servicio terciario, se utilizo firebase.

¿Qué es mongoDB y para que funciona?

MongoDB es un sistema de gestión de bases de datos NoSQL (No sólo SQL) que se basa en un modelo de datos de documentos. En lugar de utilizar tablas y filas como en las bases de datos relacionales tradicionales, MongoDB organiza la información en documentos JSON (JavaScript Object Notation), lo que facilita la representación y manipulación de datos complejos. MongoDB es altamente escalable y flexible, lo que lo hace ideal para aplicaciones que manejan grandes volúmenes de datos y requieren un esquema dinámico.

MongoDB se utiliza comúnmente en aplicaciones web y móviles, así como en entornos donde la velocidad y la escalabilidad son críticas. Su capacidad para manejar datos no estructurados o semiestructurados, así como su capacidad para escalar horizontalmente a través de múltiples servidores, lo convierte en una elección popular para desarrolladores que buscan una solución ágil y eficiente para almacenar y recuperar datos en tiempo real.

Documentacion: [MongoDB Documentation](#)

¿Qué es FireBase y para que funciona?

Firebase es una plataforma de desarrollo de aplicaciones que se destaca por su enfoque en tiempo real y su integración sencilla con las aplicaciones móviles y web. Ofrece servicios clave como Firestore, una base de datos NoSQL en tiempo real que permite la sincronización instantánea de datos entre clientes; Firebase Authentication, que simplifica la gestión de usuarios y autenticación; y Cloud Functions, que permite ejecutar código sin servidor en respuesta a eventos específicos. Además, Firebase proporciona almacenamiento en la nube, hosting web, análisis de aplicaciones, mensajería en tiempo real y más, convirtiéndose en una solución integral para el desarrollo ágil y eficiente de aplicaciones. Su facilidad de uso y escalabilidad lo hacen popular entre desarrolladores que buscan una plataforma robusta y fácil de implementar para sus proyectos.

Documentación: [Build Documentation](#) | [Firebase Documentation \(google.com\)](#)

¿Qué es NPMs cors y para que funciona?

El paquete NPM llamado "cors" facilita la implementación del concepto de "Cross-Origin Resource Sharing" (Compartir Recursos entre Orígenes Cruzados) en aplicaciones Node.js. Cuando un navegador intenta hacer solicitudes a un servidor en un dominio diferente al de la página web actual, CORS se encarga de gestionar si es seguro permitir o bloquear ese acceso. Este middleware agrega encabezados HTTP a las respuestas del servidor, indicando si los recursos pueden ser compartidos con el origen que hizo la solicitud. En resumen, CORS es esencial para permitir la comunicación segura entre dominios en aplicaciones web.

Para usar CORS en una aplicación Node.js, simplemente instalas el paquete "cors" desde NPM, lo integras en tu código y lo configuras según tus necesidades. Al habilitar CORS, puedes permitir que tu servidor acepte solicitudes desde diferentes dominios, lo que es útil, por ejemplo, al construir APIs que serán consumidas por clientes web en lugares distintos al servidor.

[cors - npm \(npmjs.com\)](https://www.npmjs.com/package/cors)

¿Qué es express y para que funciona?

Express es un marco de desarrollo web para Node.js que simplifica la creación de aplicaciones y APIs. Su popularidad radica en su enfoque minimalista y flexibilidad, facilitando la definición de rutas, el uso de middleware para procesar solicitudes, y la gestión eficiente de vistas y archivos estáticos. Con Express, los desarrolladores pueden construir de manera sencilla servidores web, diseñar rutas para manejar diversos métodos HTTP, implementar funciones intermedias, y crear APIs de manera eficaz, haciendo que el desarrollo en Node.js sea más accesible y eficiente.

[express - npm \(npmjs.com\)](https://www.npmjs.com/package/express)

¿Qué es body-parser y para que funciona?

body-parser es un middleware para Express, un marco de desarrollo web para Node.js. Su función principal es analizar el cuerpo de las solicitudes HTTP entrantes en el servidor y hacer que los datos estén disponibles de manera más fácil de manejar en el código de tu aplicación.

Cuando un cliente realiza una solicitud HTTP, como enviar datos a través de un formulario HTML o realizar una solicitud POST a una API, la información se envía en el cuerpo de la solicitud. body-parser simplifica la tarea de extraer y analizar estos datos del cuerpo, convirtiéndolos en un formato que es más conveniente para su manipulación en el código de tu servidor. Esto es especialmente útil para aplicaciones que manejan datos de formularios web o solicitudes de API que envían datos en formato JSON o urlencoded. En resumen, body-parser facilita el manejo de datos del cuerpo de las solicitudes HTTP en aplicaciones Node.js construidas con Express.

[body-parser - npm \(npmjs.com\)](https://www.npmjs.com/package/body-parser)