Aplicaciones Hibridas

Alumnas: Abril Berenguer – María Florencia Bocutti.

Comisión: DWT4AV.

• Explicación de la arquitectura de la aplicación.

FrontEnd: React + Vue

El FrontEnd de la aplicación está desarrollado utilizando React y Vue.js, dos bibliotecas y frameworks para la construcción de interfaces de usuario. Estas tecnologías permiten consumir la API proporcionada por el backend, obteniendo y enviando datos de forma eficiente. Además, se encargan de presentar la información de manera intuitiva y dinámica.

BackEnd: Node.js + Express

El BackEnd utiliza Node. js como entorno de ejecución y Express como framework principal para la construcción de la API. Esta capa es responsable de manejar la lógica de negocio de la aplicación, procesar las solicitudes provenientes del FrontEnd y devolver respuestas en formato JSON. También implementa autenticación de usuarios y autorización para garantizar la seguridad y el acceso controlado a los recursos. Además, se integra con la base de datos para realizar operaciones CRUD (crear, leer, actualizar, eliminar). Es arquitectura basada en una API REST, utiliza los middlewares de Express que permiten gestionar errores y solicitudes complejas. Para mayor seguridad, se utilizó JSON Web Tokens (JWT) en los procesos de autenticación.

Base de Datos: MongoDB

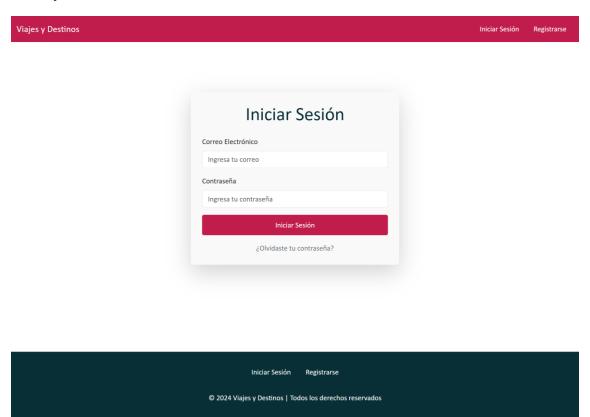
MongoDB se emplea como base de datos no relacional, basada en colecciones y documentos en formato JSON. Permite almacenar tanto información estructurada como no estructurada, facilitando el manejo de datos complejos. En esta aplicación, MongoDB se utiliza para guardar información relevante como datos de usuarios, autenticación, destinos y otros elementos clave para el funcionamiento del sistema.

El frontend y el backend interactúan mediante peticiones HTTP utilizando métodos como GET, POST, PUT, PATCH y DELETE hacia las rutas definidas en el servidor Express. Estas rutas actúan como puntos de entrada para procesar las solicitudes del cliente.

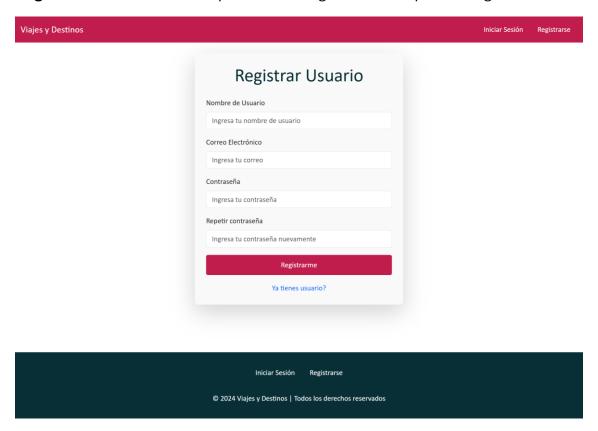
Por otro lado, el backend se comunica con la base de datos a través de los controladores. Estos controladores son responsables de recibir las solicitudes del servidor, interactuar con la base de datos MongoDB.

• Capturas de pantalla de las principales funcionalidades.

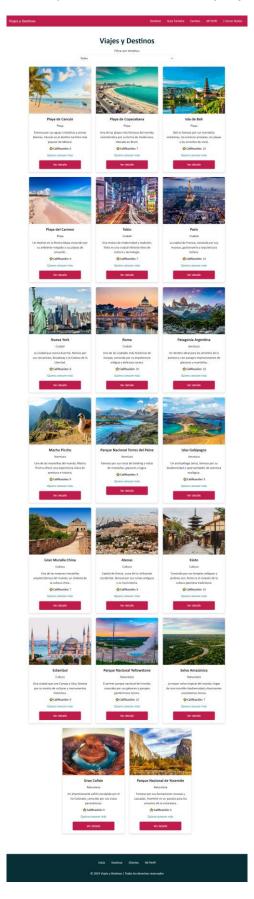
Login: donde los clientes que están autenticados pueden iniciar sesión con su email y contraseña.



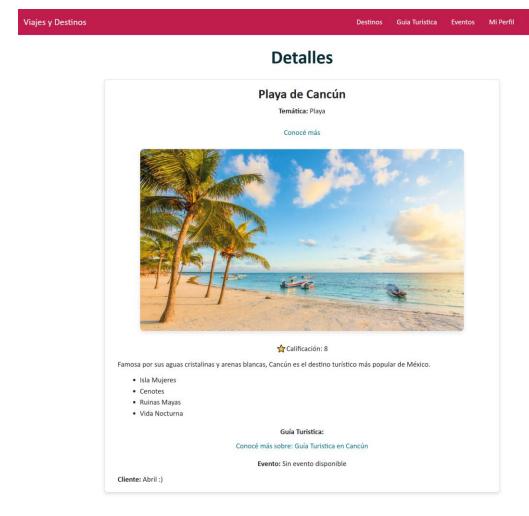
Registro: donde los clientes que no están registrados aún pueden registrarse.



Sección Viajes y Destinos: se puede visualizar los viajes y destinos.



Detalle de Destino: donde se puede visualizar la información de cada destino. Los cuales tienen vinculados el cliente que lo hizo y algunos las guías y/o eventos.

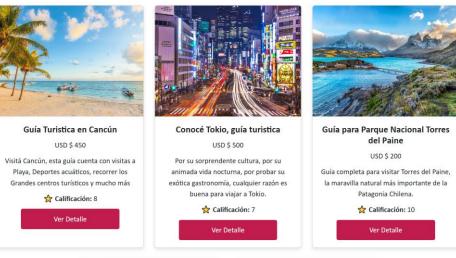




Sección Guías Turísticas: donde se pueden visualizar las guías turísticas.



Guías Turísticas







Detalle Guía Turística: donde se puede visualizar la información de cada guía turística. Las cuales tienen vinculadas el destino al que corresponde.

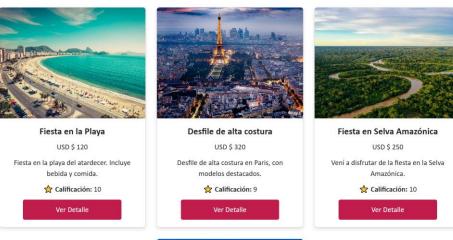




Sección Eventos: donde se pueden visualizar los eventos.



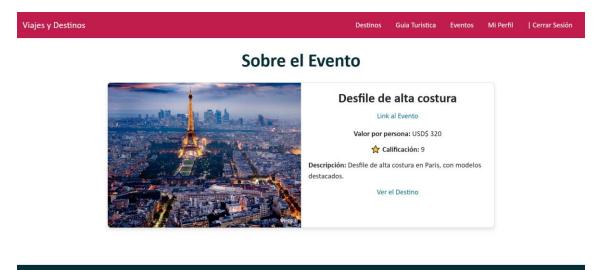
Eventos





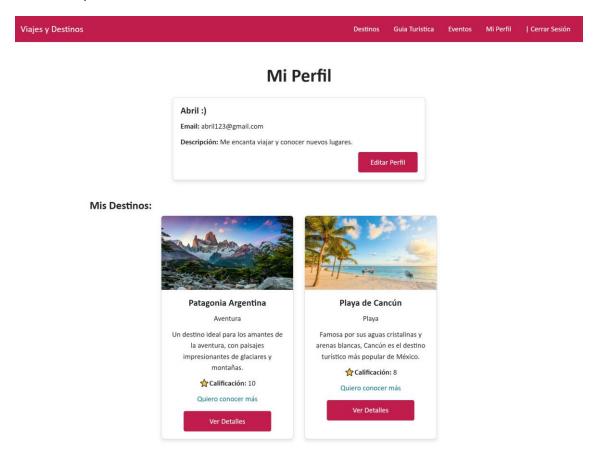


Detalle de Evento: donde se puede visualizar la información de cada evento. Los cuales tienen vinculados el destino al que corresponde.



© 2024 Viajes y Destinos | Todos los derechos reservados

Perfil: donde el cliente autenticado puede ver y editar su perfil. Además de ver los destinos que creo.





Explicación de los procesos de autenticación y autorización con JWT.

En la aplicación, se utiliza JSON Web Tokens (JWT) para implementar los procesos de autenticación y autorización de manera segura y eficiente.

1. Autenticación

La autenticación comienza cuando un usuario intenta iniciar sesión proporcionando sus credenciales (email y contraseña). Estas credenciales son enviadas al servidor, donde se validan contra los datos almacenados en la base de datos.

- Si las credenciales son válidas, el servidor genera un token JWT único, que incluye información relevante del usuario codificada y firmada con una clave secreta.
- o Este token es enviado al cliente y almacenado.

2. Autorización

La autorización verifica si un usuario autenticado tiene permisos para acceder a ciertos recursos o realizar determinadas acciones.

- Cada vez que el cliente pide acceso a un recurso protegido, manda el token JWT como parte de los datos extra de la solicitud.
- El servidor revisa el token para asegurarse de que sea válido, que no esté alterado y que tenga los permisos necesarios. Si todo está bien, le da acceso al recurso.
- Si el token no es válido, ya caducó o no tiene los permisos correctos, el servidor muestra un error.