

PLATAFORMA COLABORATIVA DE HOSPEDAJE

Trabajo Práctico Grupal

Comisión S31

Integrantes:

Cassina Matias – Legajo 33212 – Usuario de Discord: matiascassina

González D'Angelo Santiago – Legajo 33211 – Usuario de Discord:
dangelo2315

Ochoa Matias – Legajo 32290 – Usuario de Discord: matiochoa

Otero Ian Facundo – Legajo 32532 – Usuario de Discord: facuotero

Porzio María Pía – Legajo 33243 – Usuario de Discord: piaporzio_



UTN – Facultad Regional La Plata
Desarrollo de Software

ENTREGA 1

Objetivo: ofrecer un sistema que permita gestionar la información de los servicios de hospedaje ofrecidos por la empresa de turismo.

Alcance: incluye el registro y gestión de usuarios, propiedades, reservas, pagos, servicios ofrecidos y calificaciones dentro de la plataforma. Permite que los anfitriones publiquen alojamientos, definan la disponibilidad de los mismos y reciban reservas. Mientras que, los huéspedes pueden buscar propiedades, reservarlas y calificar posteriormente su estadía. A su vez, incluye la administración de pagos y supervisión general de los usuarios.

Presentación de Mockups: se presenta el enlace al proyecto en Figma que contiene los mockups diseñados para el sistema, donde se pueden visualizar las distintas vistas, distribución de elementos y la propuesta de interfaz de usuario. En el mismo desarrollamos las vistas de registrarse, iniciar sesión y reserva de propiedades. Una de las metas es lograr un diseño responsive, por ello realizamos los mockups tanto para las vistas de un teléfono móvil como de una computadora.

https://www.figma.com/team_invite/redeem/SAJrry3wH8zOOh0CjrjwHB

ENTREGA 2

Método de autenticación

Para la implementación de los procesos de inicio de sesión y registro en la aplicación se seleccionó Auth0 como herramienta principal. Auth0 es un servicio de *Identity as a Service* (IDaaS), es decir, una plataforma especializada en la gestión de autenticación y autorización de usuarios. Este servicio provee un SDK (software development key) oficial, que incluye librerías, utilidades y documentación, lo que facilita la integración del sistema de autenticación en aplicaciones desarrolladas con React.

Dado que la plataforma contempla distintos perfiles de usuario, resulta necesario establecer un mecanismo de control de permisos. Auth0 posibilita la gestión de la autorización mediante la asignación de roles a los usuarios, tales como *administrador*, *anfitrión* o *huésped*. Esto se implementa mediante el modelo RBAC (*Role-Based Access Control*), el cual permite definir roles, asociarles permisos específicos y posteriormente asignarlos a los usuarios correspondientes.

En el diseño de la aplicación se definieron tres roles principales, cada uno con diferentes niveles de permisos:

- **Huésped:** constituye el rol predeterminado que se asigna a los usuarios al momento de registrarse. Sus permisos incluyen la navegación por la aplicación,

la interacción con las propiedades, la comunicación con los anfitriones, la gestión de reservas y la realización de pagos.

- **Anfitrión:** se asigna a los usuarios que publican una propiedad. Este rol hereda los permisos del huésped y añade la posibilidad de publicar propiedades, responder a consultas de los huéspedes, administrar reservas y recibir pagos.
- **Administrador:** corresponde a usuarios seleccionados por la empresa. Este rol habilita la ejecución de tareas administrativas, tales como el control de estadísticas, la gestión de usuarios, reservas y propiedades.

Desarrollo de APIs

La API diseñada para nuestra plataforma de hospedaje tiene como objetivo principal permitir la comunicación entre el frontend y la base de datos, garantizando que las operaciones de consulta, registro y actualización de información se realicen de manera segura y eficiente.

Los endpoints definidos cubren las funcionalidades esenciales de la aplicación, incluyendo la búsqueda de propiedades, la gestión de precios, el registro de usuarios y la creación de reservas.

A continuación, se detallan los endpoints que conforman la API, describiendo su propósito, métodos de acceso, parámetros y respuestas esperadas.

- **GET PROPIEDAD** (para buscador): Es un endpoint que devuelve la lista de propiedades disponibles según los filtros que el usuario aplique, tales como ubicación, cantidad de huéspedes, fecha de disponibilidad, etc.

Este endpoint utiliza el método GET y los posibles parámetros de consulta son filtrar por ciudad, por fechas de inicio y fin buscando propiedades disponibles en ese rango y cantidad máxima de huéspedes permitida.

- **GET PRECIO** (para actualizar precios por periodo de días de reserva): Es un endpoint que actualiza el precio de una propiedad dinámicamente en base a la cantidad de días a reservar.

Este endpoint utiliza el método GET y los parámetros que utiliza para filtrar son el día inicial y el día final de reserva.

- **POST RESERVA** (guardar las reservas): Es un endpoint que se encarga de enviar los datos de una nueva reserva para ser almacenados en la base de datos. Para ello se utiliza el método POST que envía los datos obtenidos en el cuerpo de la solicitud para que el servidor cree un registro nuevo en la base de datos.

- **POST USUARIO** (para registrar): Este endpoint permite registrar un nuevo usuario en el sistema. El método que utiliza es el POST enviando los datos en el cuerpo de la solicitud para que el servidor cree un nuevo registro en la base de datos.
- **PUT PRECIO** (actualizar precio de propiedad): Este endpoint permite modificar el precio de una propiedad existente. Es útil en el desarrollo de nuestra aplicación para ajustes de tarifas dinámicas o cambios administrativos. El método utilizado es el PUT.

A su vez, en la app haremos uso de dos APIs externas, la primera siendo Auth0 para realizar la autenticación y autorización de los usuarios.

La segunda API utilizada es DialogFlow, una plataforma de Google que permite crear chatbots e interfaces conversacionales basadas en inteligencia artificial. Su objetivo es que un sistema pueda entender lenguaje natural (lo que escribe o dice un usuario) y responder de manera adecuada.

En este proyecto, se emplea DialogFlow para implementar un chatbot que conteste preguntas frecuentes de los usuarios. La herramienta permite definir *Intents*, que representan lo que el usuario quiere, y configurar las respuestas predeterminadas para cada uno de ellos. De esta manera, se agrega dinamismo a la aplicación y se reducen los tiempos de espera de los usuarios para obtener respuestas.

El chatbot no responde por sí mismo, sino que se comunica con el servicio de DialogFlow de Google mediante un conector oficial. Cada vez que el usuario escribe un mensaje, el servicio interpreta la consulta y devuelve la respuesta configurada en el asistente.

En cuanto al funcionamiento técnico, el chatbox implementado con DialogFlow Messenger se comunica con la API de DialogFlow. Internamente utiliza el endpoint **detectIntent** el cual se encarga de recibir el mensaje del usuario, identificar la intención (intent) mediante el motor de la NLU (Natural Language Understanding) de DialogFlow y retornar la respuesta correspondiente definida por el agente.

En el código embebido del chatbox se emplea el componente <df-messenger>, que carga el script oficial de Google:

```
<script src="https://www.gstatic.com/dialogflow-console/fast/messenger/bootstrap.js?v=1"></script>
```

Este componente no expone directamente el endpoint, sino que se encarga internamente de gestionar las llamadas a la API de DialogFlow.

Base de Datos

A continuación, adjuntamos el diagrama de tablas de la base de datos de la aplicación la cual almacenará los datos más importantes de la misma para asegurar su correcto funcionamiento.

