Universidad CAECE, Mar del Plata, Argentina

Desarrollo de Proyecto

Personaliza Tu Viaje

Proyecto Final

Autor: Gioia Agustín

GitHub: github.com/agusgioia

Carrera: Tecnicatura en Programación Universitaria

Tutor: Miguel Méndez Garabetti

Fecha de Presentación: Diciembre de 2024

Abstract **4**
Introducción **4**
Objetivos y Metas del Proyecto **5**
Justificación del Proyecto **5**
Público Objetivo **6**
Visión General del Sistema **6**
Requisitos del Sistema **7**
Requerimientos Funcionales 7
El sistema deberá permitir a un usuario: 7
Requerimientos No Funcionales 8
Requerimientos Futuros 9
Diagramas de casos de uso **9**
Historias de usuario o casos de uso 10
CU 004: Crear viaje 13
Diagrama de Clases **16**
Diagramas de Secuencia **17**
Figura 3- Diagrama de Secuencia **17**
Diseño de Interfaces Gráficas **17**
Enfoque de Diseño General 18
Restricciones de diseño 18
Prototipos de diseño propuestos inicialmente 18
Diseño Arquitectónico **22**
Arquitectura Cliente-Servidor: 22
Componentes principales(Angular): 23
Autenticación 23
Módulo de Búsqueda de Servicios 23
Módulos de Viajes y Cuenta 23
Escalabilidad **24**
Otros Componentes **24**
Interacción en Tiempo Real 24
Manejo de Errores 24
Rutas y Navegación 25
Api de Spring Boot **25**
Spring Boot **25**
Implementación **26**
Diseño de la base de datos 26
Backend del Sistema 26
Frontend del Sistema 30
Proceso de desarrollo del proyecto **31**
Manual de Usuario **33**
Página principal 33
Descripción de la Página Principal 33
Formulario de Iniciar Sesión 34
Home **36**

Viajes.....39
Lista de Viajes..... 39
Account.....39
Suscripción.....40
Conclusiones.....41
Referencias.....43

Abstract

El presente proyecto tiene como objetivo el desarrollo de un sistema de reserva de servicios turísticos tales como reservas de pasajes de vuelos y reservas de alojamiento en hoteles a través del uso de una Api externa conocida como Amadeus. Este sistema está diseñado para que cada usuario que ingrese en la plataforma pueda realizar dichas reservas y tener registro de las mismas en tiempo real.

Introducción

La plataforma "Personaliza tu viaje" está diseñada con una estructura clara e intuitiva que incluye las siguientes secciones principales: Inicio, Mis Viajes, Perfil, Suscribirse, Registro y Login. Estas funcionalidades permiten a los usuarios navegar de manera sencilla a través de diferentes opciones cómo las de gestionar sus datos personales y gestionar sus viajes.

El sistema organiza la búsqueda de servicios en base a los datos personales del usuario como el presupuesto disponible del usuario y otras preferencias personales.

El enfoque del proyecto viene a facilitar la problemática de los usuarios a la hora de la organización de un viaje y todo lo que conlleva la contratación de servicios para llevarlo a cabo. Se integran servicios de transporte, alojamiento y actividades turísticas todo integrado en la misma plataforma promoviendo una planificación eficiente.

Al fomentar la exploración de nuevas actividades y facilitar la interacción con servicios específicos, la plataforma busca crear una comunidad activa y conectada,

promoviendo tanto el bienestar personal como el crecimiento del mercado de actividades recreativas.

Objetivos y Metas del Proyecto

1. **Contratación de Servicios mediante una API:** La plataforma permite a los usuarios obtener servicios de transporte, alojamiento y actividades turísticas.
2. **Personalizar perfil de usuarios:** La plataforma permite a los usuarios agregar preferencias a su perfil que sirvan para la búsqueda de servicios.
3. **Suscripción a la Página:** Además de registrarse, los usuarios pueden suscribirse a la página de forma mensual para acceder a algunas de las funciones que no están disponibles a todos los usuarios.
4. **Conectar a usuarios**: Los usuarios podrán compartir sus viajes con familiares o amigos de forma privada o de forma pública en un foro.
5. **Facilitar a las empresas la promoción de sus servicios y actividades.**

Justificación del Proyecto

El proyecto "Personaliza tu Viaje" responde a una doble necesidad en el ámbito social y tecnológico: fomentar el turismo y aprovechar la digitalización para conectar a los usuarios entre ellos y con las empresas. La plataforma se justifica en la creciente demanda de soluciones tecnológicas que no solo faciliten el acceso a actividades turísticas, sino que también impulsen el crecimiento de un ecosistema digital para el mercado de servicios relacionados con viajes.

Público Objetivo

El proyecto está dirigido a:

- **Usuarios individuales** de todas las edades interesados en realizar un viaje ya sea con fines laborales o recreativos.

Visión General del Sistema

El software conocido como “Personaliza tu Viaje” está diseñado para abordar las necesidades de los usuarios permitiéndoles gestionar las reservas de sus servicios así como también funcionar como un registro de los mismos. El sistema está compuesto por tres componentes principales:

- **Frontend**: Una interfaz web diseñada para ser intuitiva y responsiva. Implementada en Angular con un formato de componentes standalone, esto quiere decir que cada componente de la interfaz importa sus propios módulos. Esta interfaz permite a los usuarios su registro en la plataforma, gestionar sus servicios y consultar sus datos.
- **Backend**: Una API RESTful desarrollada en Spring Boot que sirve como puente entre el frontend y la base de datos. Este componente maneja la lógica de negocio, como el enlace con la base de datos de Firebase y la gestión de los datos de los usuarios.
- **Firebase**: Una base de datos no relacional basada en la lógica de negocio planteada en el backend, siendo la misma adaptada a los campos de los datos recuperados por la API de Amadeus para poder almacenar de forma correcta y precisa la información.

Requisitos del Sistema

Al comenzar el proceso de descubrimiento de las necesidades de la plataforma, se creó una lista de requerimientos clave para guiar el desarrollo de la aplicación. Estos requerimientos se enfocaron en los aspectos funcionales y técnicos necesarios para cumplir con los objetivos del proyecto. A continuación, se detalla dicha lista:

Requerimientos Funcionales

El sistema deberá permitir a un usuario:

RF001 El sistema debe permitir registrarse mediante un formulario ingresando un nombre de usuario, correo electrónico y contraseña.

RF002 El sistema debe garantizar la autenticación segura para el inicio de sesión.

RF003 El sistema debe permitir al usuario crear y editar su perfil.

RF004 El sistema debe permitir la reserva de servicios turísticos basados en los datos necesarios para realizar peticiones a la API de Amadeus.

Requerimientos No Funcionales

RNF 001: **Rendimiento** - La aplicación debe cargar y responder rápidamente a las solicitudes de los usuarios.

RNF 002: **Rendimiento** - Debe ser capaz de manejar múltiples solicitudes simultáneamente sin bajar su rendimiento.

RNF 003: **Usabilidad** - Interfaz intuitiva y fácil de usar.

RNF 004: **Seguridad** - Proteger los datos del usuario con medidas de seguridad como autenticación segura.

RNF 005: **Escalabilidad -** La aplicación debe ser escalable para manejar un número creciente de usuarios y datos sin una disminución notable en el rendimiento.

RNF 006: **Disponibilidad** - La aplicación debe estar disponible y operativa el 99.9% del tiempo, con un tiempo de inactividad mínimo planificado para mantenimiento.

RNF 007 Código bien documentado y estructurado para facilitar futuras modificaciones y mejoras.

RNF 008: **Extensibilidad** - Arquitectura modular que permita agregar o actualizar funcionalidades sin grandes cambios en el sistema.

RNF 009 El sistema deberá ser desarrollado utilizando HTML, CSS, JavaScript, Angular, TypeScript, y Spring boot.

Requerimientos Futuros

1. El sistema deberá permitir la creación y personalización de itinerarios de viaje con opciones para agregar destinos, actividades, alojamiento y transporte.
2. El sistema deberá permitir la visualización de itinerarios en un formato cronológico y mapas interactivos.
3. El sistema deberá poder conectar con APIs de servicios como Google Maps y Amadeus, para proporcionar información en tiempo real sobre transporte, alojamiento y actividades.

4. El sistema deberá permitir que los usuarios compartan sus itinerarios con amigos o familiares y colaboren en la planificación del viaje.
5. El sistema debe permitir dejar comentarios y sugerencias en itinerarios compartidos.
6. El sistema deberá brindar notificaciones en casos de cambios en precios, clima, vuelos u otros eventos importantes relacionados con el viaje.
7. El sistema deberá brindar recordatorios para actividades planificadas.
8. Adaptación a diferentes dispositivos, incluyendo desktops, tablets y smartphones (diseño responsive).

Diagramas de casos de uso

Se confeccionó un diagrama de casos de uso para representar las interacciones clave entre los usuarios y la aplicación, destacando los distintos casos de uso que cubren los requerimientos funcionales del sistema. Este diagrama ayudó a visualizar las funcionalidades principales y los actores involucrados en el sistema, facilitando el entendimiento de cómo cada componente interactúa con los usuarios y los servicios proporcionados por la aplicación. (ver figura 1)

Diagrama de Casos de Uso

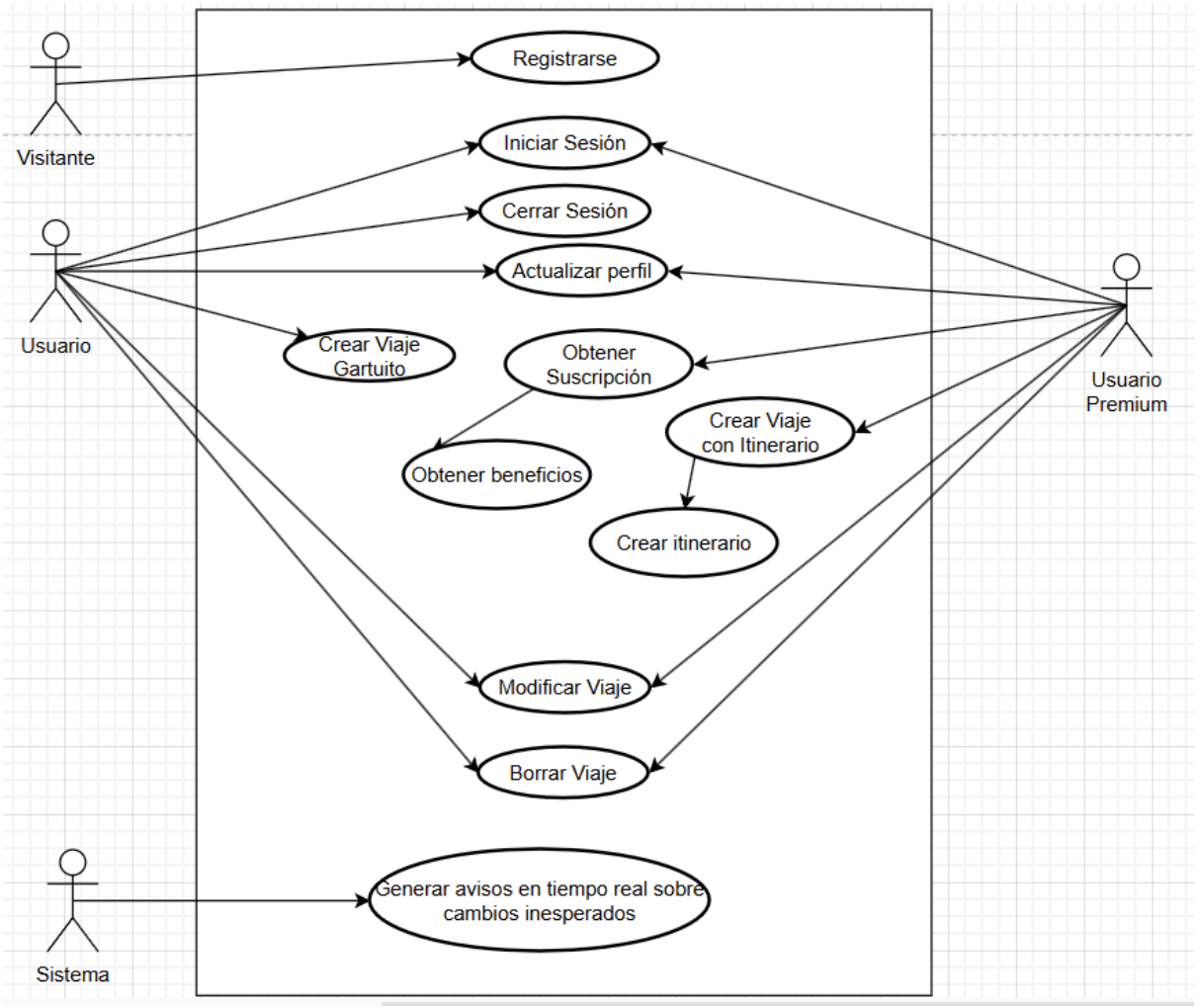


Figura 1 - Diagrama de casos de uso.

Historias de usuario o casos de uso.

Para definir claramente las funcionalidades del sistema, se crearon varias historias de usuario que describen de manera detallada cómo los diferentes usuarios interactuaron con la aplicación. Aquí se describe una de ellas:

CU-001: Registrar Usuario

Actores: Usuario.

Precondiciones:

- No debe de existir un documento del usuario a registrar.

Flujo principal:

1. El usuario ingresa a la plataforma y se muestra la interfaz de LOGIN.
2. El usuario hace clic en el botón de registro ya que no tiene una cuenta registrada.
3. El sistema muestra la interfaz de REGISTRO.
4. El usuario ingresa su nombre de usuario, su email y su contraseña, luego da click en el botón de registrarse.[4.1]
5. El sistema registra al usuario y le asigna un identificador único a su documento y lo guarda en la base de datos de Firebase.

Flujo Alternativo:

[4.1]:

4.1.1 Si los datos no son válidos el sistema muestra un mensaje de error.

4.1.2 El sistema vuelve al punto 4.

Postcondiciones: El usuario está registrado en la base de datos de Firebase.

CU-002: Iniciar Sesión

Actores: Usuario.

Precondiciones:

- Que el usuario esté registrado.

Flujo Normal:

1. El usuario ingresa en la plataforma y se muestra la interfaz de LOGIN.
2. El usuario ingresa su email y contraseña.[2.1]
3. El sistema valida los datos ingresados y muestra la interfaz de HOME.

Flujo Alternativo:

[2.1]:

2.1.1 El sistema muestra un mensaje de error.

2.1.2 El sistema vuelve al punto 2.

Postcondiciones:

CU-003: Editar Perfil - El sistema debe permitir al usuario editar su perfil.

Precondiciones:

- El usuario debe tener la sesión iniciada.

Flujo Normal:

1. El usuario hace click en la pestaña Account de la barra de navegación.
2. El sistema muestra la interfaz ACCOUNT.

3. El usuario ingresa sus datos personales como nombre de usuario, email, edad, género, presupuesto y hace click en guardar perfil.[3.1]
4. El sistema valida los datos ingresados y actualiza el documento en Firebase.

Flujo Alternativo:

[3.1]:

3.1.1 El sistema muestra un mensaje de error.

3.1.2 El sistema vuelve al punto 3.

Postcondiciones: El perfil del usuario se actualizó con éxito.

CU 004: Crear viaje.

Objetivo: El usuario debe poder crear un viaje y visualizarlo en su perfil.

Actor: Usuario

Pre Condición: Iniciar sesión

Flujo Normal

1. El usuario ingresa la ciudad de origen, la ciudad de destino, una fecha de salida, una fecha de regreso(opcional) y la cantidad de adultos en la interfaz GIU HOME.
2. El sistema recupera la información deseada a través de la Api de Amadeus y despliega una serie de ofertas de vuelo que cumplen las condiciones ingresadas por el usuario. [2.1].
3. El usuario elige una de las ofertas mostradas por el sistema y hace clic en el botón seleccionar viaje.

4. El usuario ingresa la ciudad donde quiere alojarse, la fecha de ingreso, la fecha de salida, la cantidad de adultos y la cantidad de habitaciones deseadas.
5. El sistema recupera la información deseada a través de la Api de Amadeus y despliega una serie de ofertas de alojamiento que cumplen con las condiciones ingresadas por el usuario[5.1]
6. El usuario elige una de las ofertas mostradas por el sistema y hace clic en el botón de seleccionar alojamiento.
7. El usuario presiona en el botón “Avanzar con la reserva”.
8. El sistema abre la GIU BOOKING mostrándole al usuario las ofertas que seleccionó y le brinda la opción de reservar los servicios o volver atrás la operación.[8.1]
9. El usuario hace clic en reservar y es redireccionado a la interfaz GIU PAYMENT TYPE donde se le ofrecen distintas formas de abonar el servicio.
10. El usuario selecciona el método que desee y realiza el pago correspondiente.
11. El sistema muestra un formulario con los datos correspondientes a ingresar para confirmar el pago.
12. El Usuario completa los datos necesarios y acepta el pago.[12.1][12.2]
13. El sistema muestra un aviso de que el pago se realizó con éxito y abre la GIU TRIPS mostrando el viaje reservado.

Flujos Alternativos:

[2.1]:

2.1.1 El sistema no muestra ninguna oferta y avisa que no se obtuvo ninguna oferta que cumpla las condiciones provistas.

2.1.2 Regresa al punto 1.

[5.1]:

5.1.1 El sistema no muestra ninguna oferta y avisa que no se obtuvo ninguna oferta que cumpla las condiciones provistas.

5.1.2 Regresa al punto 4.

[8.1]:

8.1.1 El usuario hace clic en el botón cancelar.

8.1.2 El sistema muestra la GIU HOME con los formularios vacíos listos para una nueva búsqueda.

8.1.3 Regresa al punto 1.

[12.1]:

12.1.1 Se produce un error en el método de pago externo y el sistema muestra un mensaje de error.

12.1.2 Regresa a punto 12

[12.2]:

12.2.1 El sistema muestra un mensaje de error debido a que los datos ingresados son incorrectos.

12.2.2 Regresa a punto 12.

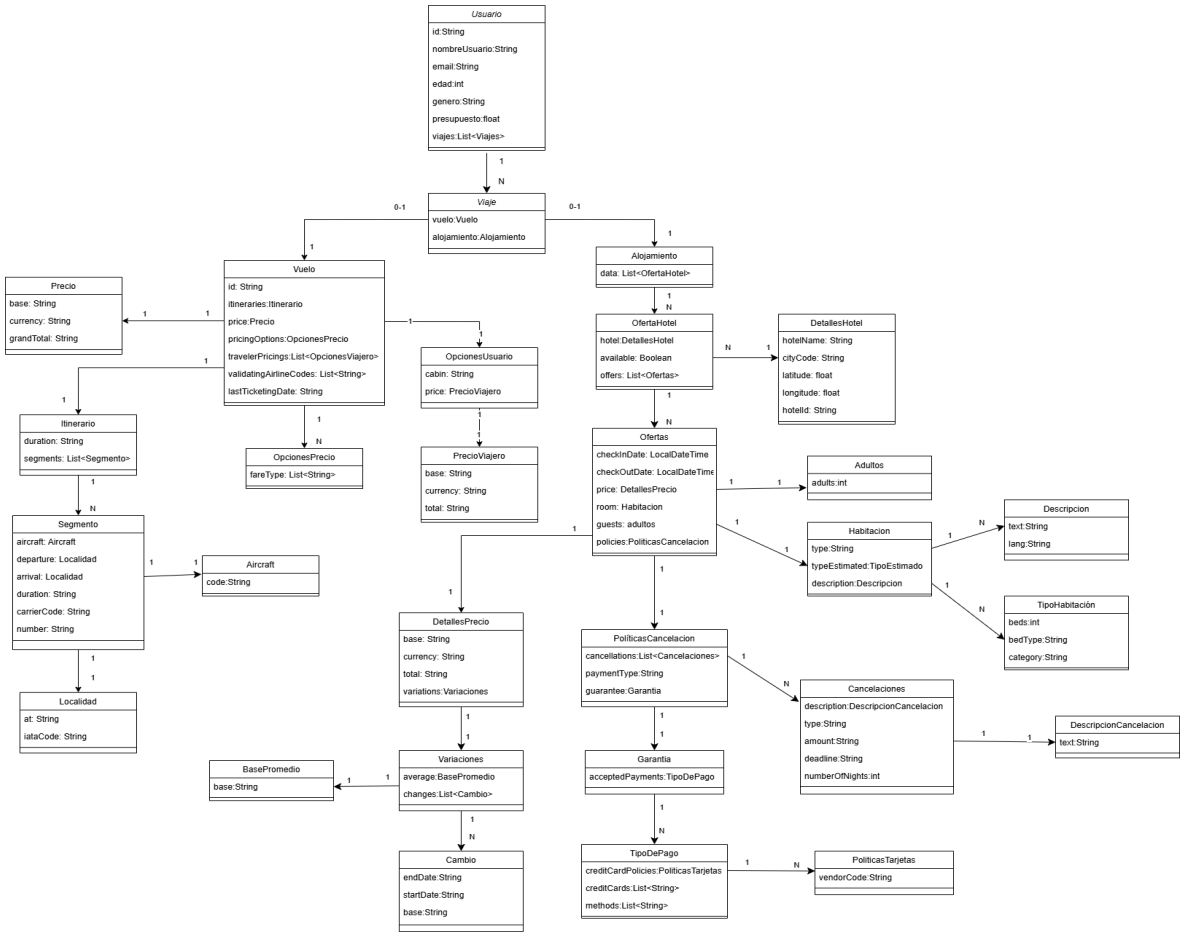
Postcondiciones: Viaje reservado.

Diagrama de Clases

Para respaldar el diseño y las funcionalidades de la aplicación, se ha confeccionado un diagrama de clases que representa la estructura de almacenamiento en Firebase, actuando como un modelo de base de datos.

Este diagrama detalla las entidades principales, sus relaciones y los campos gestionados en la base de datos. (ver figura 2)

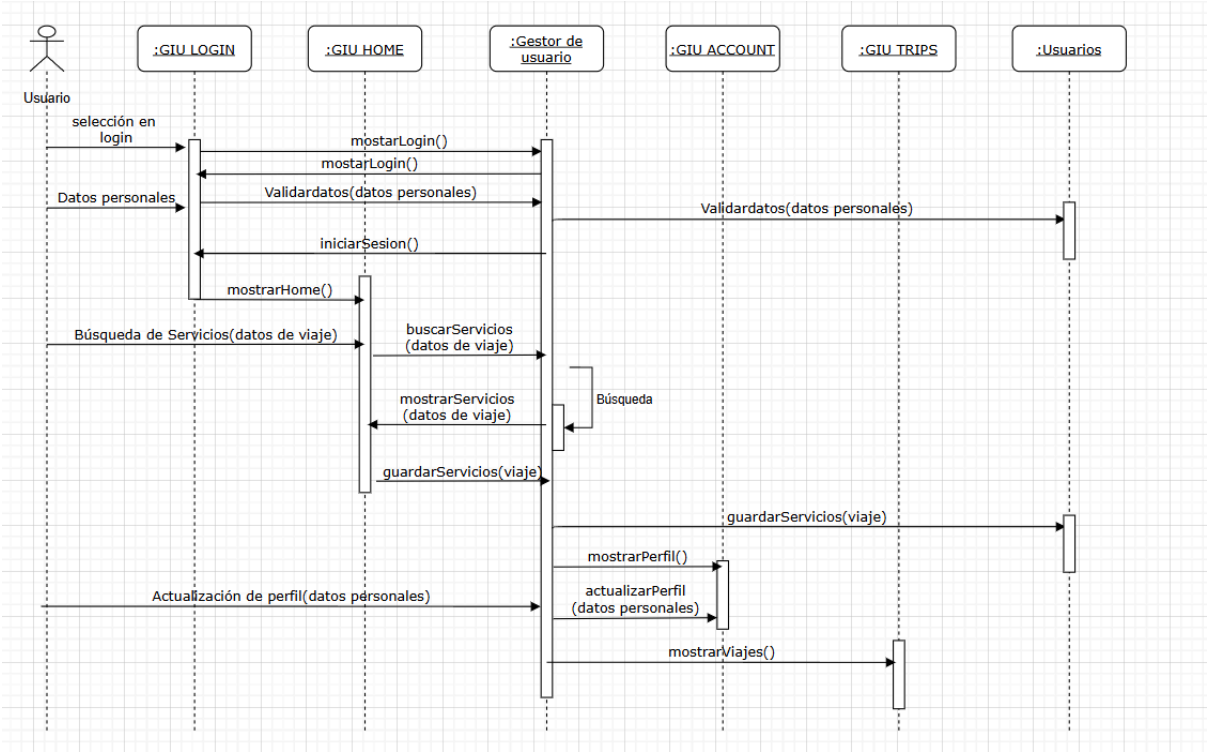
Figura 2 - Diagrama de Clases



Diagramas de Secuencia

Para modelar las interacciones detalladas entre los diferentes componentes del sistema, se confeccionó un diagrama de secuencia. Este diagrama muestra cómo el actor (usuario) interactúa con el sistema a lo largo del tiempo. Los diagramas de secuencia son clave para visualizar los flujos de procesos en la aplicación. (ver figura 3)

Figura 3- Diagrama de Secuencia



Diseño de Interfaces Gráficas

Enfoque de Diseño General

El diseño de la aplicación sigue un enfoque minimalista, enfocado en la experiencia del usuario (UX) y la accesibilidad así como también buscando simplificar la implementación de las funciones de la aplicación. Se utilizó Bootstrap combinado con módulos provistos por NgPrime(junto con Primeflex) como los framework de CSS principal para acelerar el desarrollo y garantizar una interfaz responsive y coherente.

Restricciones de diseño

Este proyecto es compatible con una amplia gama de dispositivos móviles y de escritorio, incluyendo smartphones, tablets, computadoras portátiles y de escritorio, con diferentes capacidades de procesamiento y memoria.

Además, es compatible con múltiples sistemas operativos, incluyendo Windows, Linux, macOS, iOS y Android, para garantizar una amplia accesibilidad para los usuarios.

El diseño de la aplicación es escalable para permitir el crecimiento futuro y la adición de nuevas funcionalidades y características sin comprometer el rendimiento o la estabilidad del sistema.

Prototipos de diseño propuestos inicialmente

En las figuras 4, 5, 6, 7 y 8 se presentan los prototipos de diseño propuestos inicialmente.

Figura 5 - GIU REGISTRO

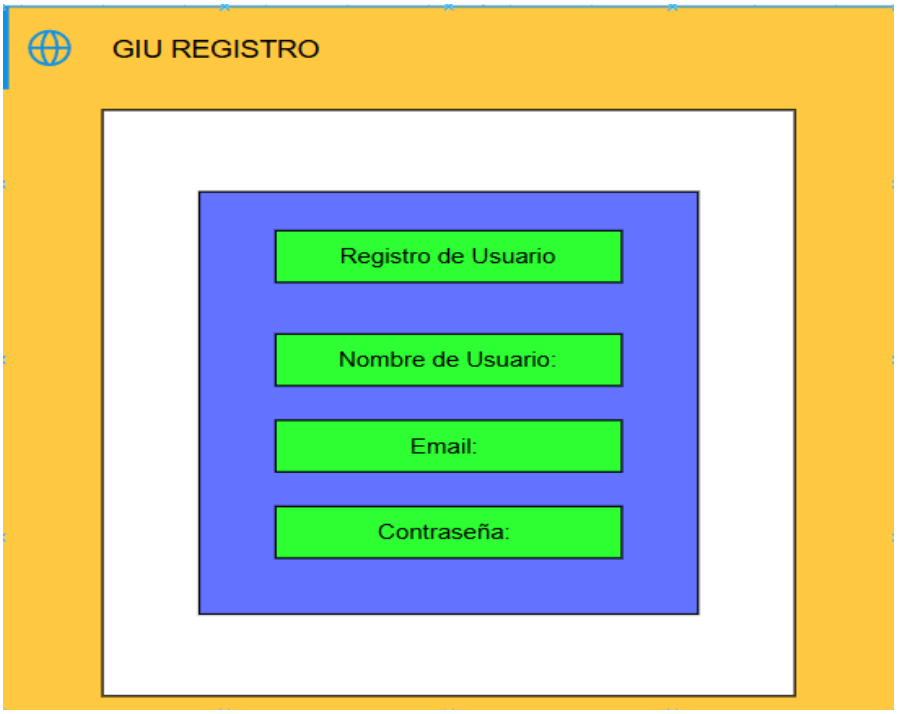


Figura 4 - GIU LOGIN

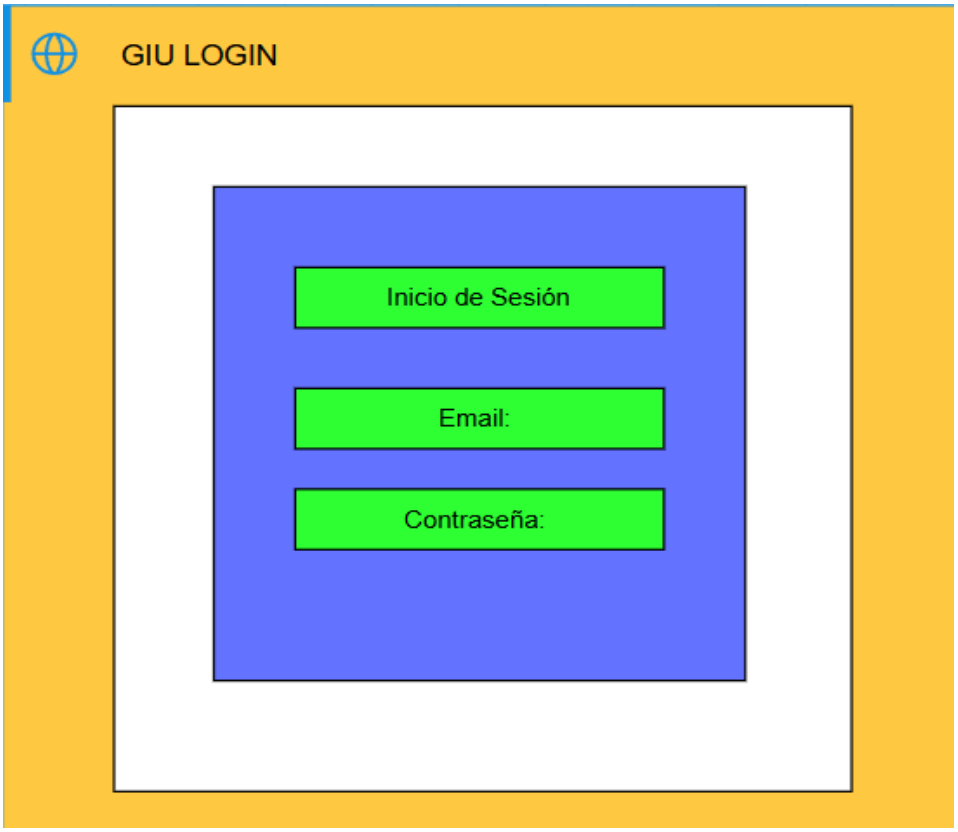


Figura 6 - GIU HOME

Figura 7 - GIU TRIPS

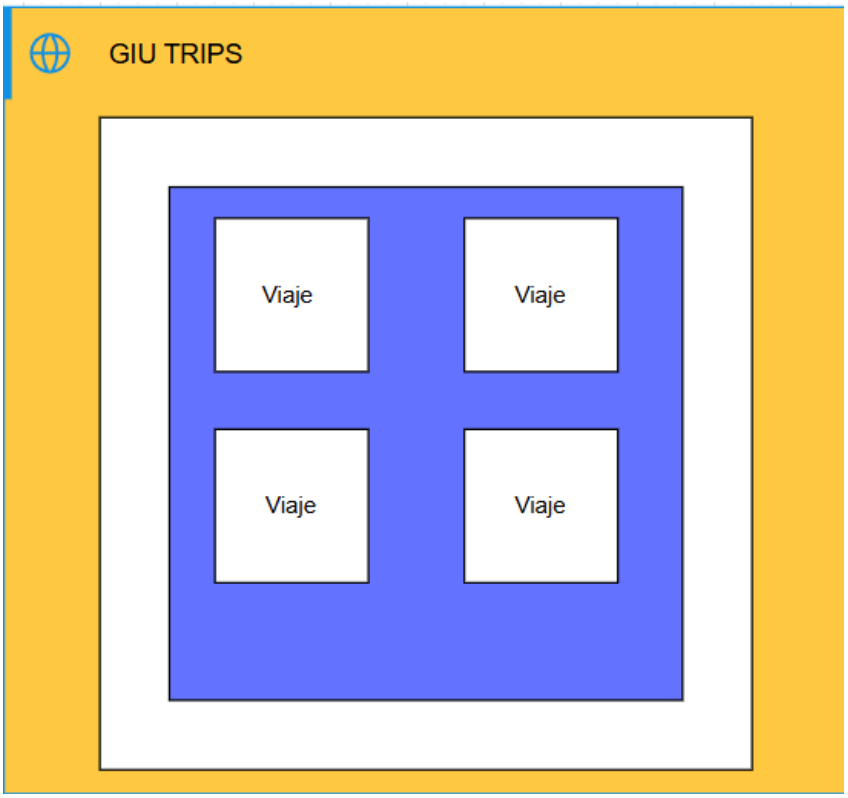


Figura 8 - GIU ACCOUNT

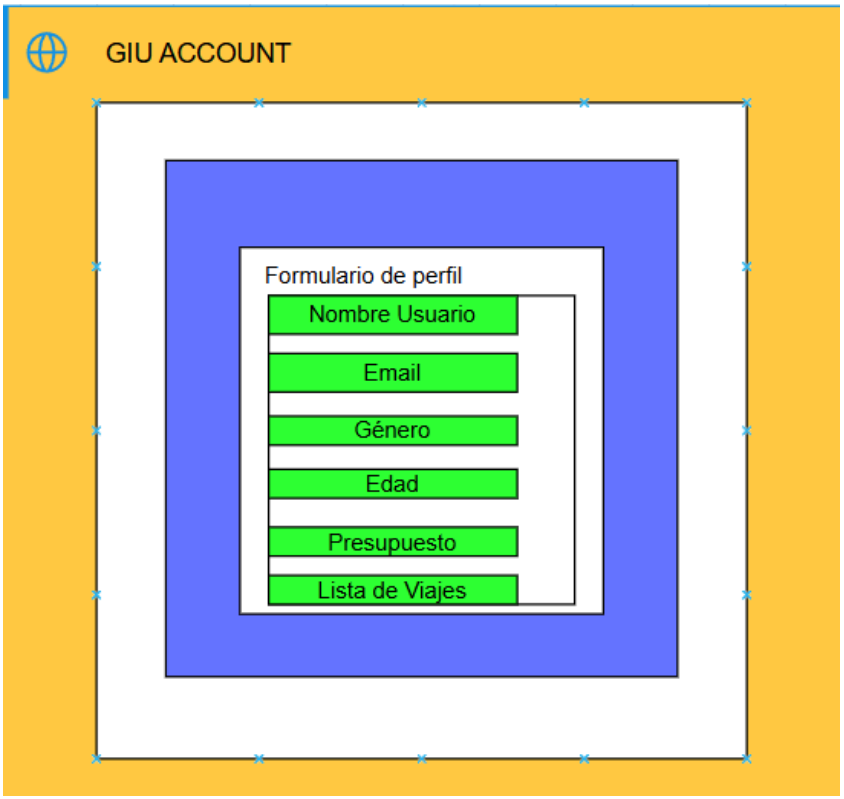
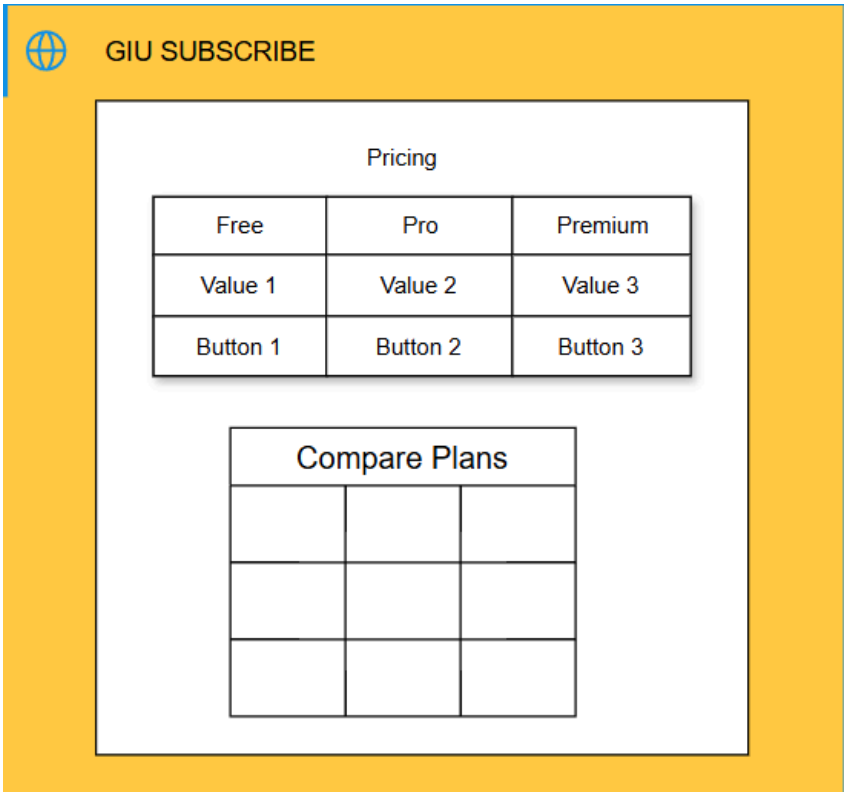


Figura 9 - GIU SUBSCRIBE



Diseño Arquitectónico

Arquitectura Cliente-Servidor:

El diseño arquitectónico de la aplicación sigue el modelo Cliente-Servidor, donde Angular actúa en el lado del cliente y Firebase se utiliza como servidor para la base de datos y la autenticación.

Cliente(Angular):

Angular se implementó como parte del cliente en la aplicación. Se encarga de gestionar la interfaz de usuario y la lógica de presentación, permitiendo a los usuarios interactuar con la aplicación de manera fluida. Para la comunicación entre el cliente y el servidor, se desarrolló un servicio en Angular para conectar con el servidor de Spring Boot que conecta con la base de datos en Firebase. Además, Angular se encarga de manejar todos los aspectos relacionados con la autenticación y presentación de los datos a través de FirebaseAuth.

Servidor(Firebase y Spring Boot):

A través de una Api desarrollada en Spring Boot que proporciona la infraestructura necesaria para almacenar los datos, se realiza una conexión con la Cloud Firebase de Firebase donde se almacenan y se recuperan los datos de los usuarios. Por otro lado, Firebase Authentication se encarga del registro y autenticación de los usuarios.

Componentes principales(Angular):

Autenticación

Se configuraron los componentes y páginas de autenticación utilizando Angular. Este módulo interactúa con Firebase Autenticación para manejar el registro e inicio de sesión de los usuarios, proporcionando una capa de seguridad robusta y fácil de integrar con otras funcionalidades de Firebase.

Módulo de Búsqueda de Servicios

Se implementó un módulo donde se le presentan al usuario dos formularios cada uno con campos a completar para realizar búsquedas de ofertas de servicios que ofrece la aplicación a partir del uso de la Api de Amadeus. En este caso se tiene un formulario para la búsqueda de ofertas de vuelo y otro para búsqueda de ofertas de alojamiento con toda la información necesaria para que el usuario decida qué oferta desea reservar.

Módulos de Viajes y Cuenta

Se implementó un módulo en Angular donde cada usuarios puede gestionar los viajes ya reservados. Dichos datos están en la Cloud Firebase lo que asegura la consistencia y disponibilidad de la información en tiempo real. Además se implementó otro módulo donde se le permite al usuario actualizar sus datos personales que serán usados para la reserva de servicios.

Módulo de Suscripción

Se implementó un GIU para la suscripción a la página que permitiría acceder a funciones futuras de la aplicación cuando estas estén implementadas.

Escalabilidad

A medida que la web crezca, Firebase puede manejar cargas significativas. Firebase Hosting es una opción para servir la web de manera rápida y escalable.

Otros Componentes

- Componentes Reutilizables: Se diseñaron componentes como el Header, Footer y formularios para mantener la coherencia en la interfaz y facilitar el mantenimiento del código.

Gestión del Estado

Servicios como AuthService, UserService, HotelService, FlightService, SharedService permiten separar la lógica de presentación de la lógica de datos, mejorando la organización y escalabilidad.

Interacción en Tiempo Real

- Cloud Firestore permite sincronizar datos en tiempo real, reflejando cambios inmediatamente en la interfaz sin necesidad de recargar la página.

Manejo de Errores

- Se implementaron tipos de datos Observables para que el flujo del código sea claro y separar el código del caso exitoso y el error, tanto en el uso de Apis externas como en la autenticación.
- Para los mensajes de error se usó un módulo perteneciente a un framework de Angular llamado ngprime que permite unos mensajes de formato pop up amigables para el usuario, mejorando la experiencia.

Rutas y Navegación

- Utilización de Router Link de Angular para gestionar la navegación entre componentes, cada componente es independiente por ende importa todos los módulos que utiliza cada uno, todos los componentes están conectados por la barra de navegación del Header, salvo algunos a los que se accede al interactuar con otros elementos de la página.

Api de Spring Boot

Spring Boot

- UserController: En este paquete como su nombre lo indica están los endpoints correspondientes con los que se realizan todas las peticiones http de la aplicación para la gestión de los datos del usuario.
- Firebase: En este paquete se define una clase que permite la conexión a la base de datos de Firestore una vez que se ejecuta el método main de la clase ejecutable.

- Lógica De Negocio: En este paquete están todas las clases correspondientes a los datos de los usuarios que son almacenados en la base de datos de Cloud Firestore.
- Service: En este paquete se encuentra definida una interfaz donde están declarados los métodos correspondientes del controlador, además de la clase que implementa esta interfaz y donde se desarrolla su implementación.
- Main: Clase principal responsable de ejecutar la aplicación.

Implementación

Diseño de la base de datos

Se adoptó como base de datos una herramienta de la plataforma conocida como Firebase que otorga una base de datos no relacional en la nube, donde se pueden crear colecciones con documentos cada uno con un identificador único. Esto resultó muy útil ya que facilita mucho el registro de los campos desde el backend.

Un ejemplo de un documento registrado sería el siguiente:

El id del documento y dentro del documento los siguientes campos:

- Nombre de Usuario.
- Email.
- Edad.
- Género.
- Presupuesto.

- Viajes: Este campo es donde se almacena toda la información de las reservas de servicios del usuario.

Backend del Sistema

Configuración y Estructura General del Backend

El backend del sistema fue desarrollado utilizando JAVA y el framework Spring Boot para la construcción de una API RESTful. Esta arquitectura permite una comunicación eficiente entre el frontend y la base de datos de Firebase. Además, se implementó un controlador para gestionar la lógica del negocio y rutas para mapear los endpoints de la API.

Ejemplo de controlador: Usuario

El controlador de propiedades gestiona las operaciones CRUD relacionadas con las propiedades de los usuarios. A continuación, se presenta un ejemplo de la implementaci´ on del controlador de usuarios:

```
@RestController
@RequestMapping(value = "/user")
@CrossOrigin("http://localhost:4200/")
public class UserController {


  @Autowired
  private UserService service;


  @RequestMapping(value = "/{id}",method = RequestMethod.GET)
  public ResponseEntity ObtenerUsuario(@PathVariable(value = "id")String id)throws
Exception  {
     return new ResponseEntity(service.ObtenerUsuario(id),HttpStatus.OK);
```

```
    }

    @RequestMapping(value = "/lista", method = RequestMethod.GET)
    public ResponseEntity ListaUsuarios() {
        return new ResponseEntity(service.ListaUsuarios(), HttpStatus.OK);
    }

    @RequestMapping(value = "/nuevo", method = RequestMethod.POST)
    public ResponseEntity NuevoUsuario(@RequestBody Usuario usuario) {
        return new ResponseEntity(service.NuevoUsuario(usuario), HttpStatus.OK);
    }

    @RequestMapping(value = "/{id}/editar", method = RequestMethod.PUT)
    public ResponseEntity EditarUsuario(@PathVariable(value = "id") String id,
@RequestBody Usuario usuario) {
        return new ResponseEntity(service.EditarUsuario(id,usuario), HttpStatus.OK);
    }

    @RequestMapping(value = "/{id}/patch", method = RequestMethod.PATCH)
    public ResponseEntity PatchUsuario(@PathVariable (value = "id") String id,
@RequestBody Map<String, Object> updates){
        return new ResponseEntity(service.PatchUsuario(id,updates),HttpStatus.OK);
    }

    @RequestMapping(value = "/{id}/borrar",method = RequestMethod.DELETE)
```

```
  public ResponseEntity EliminarUsuario(@PathVariable(value = "id")String id){
      return new ResponseEntity(service.EliminarUsuario(id),HttpStatus.OK);
  }

  @RequestMapping(value = "/{id}/borrarViaje/{idViaje}", method =
RequestMethod.DELETE)
  public ResponseEntity EliminarViaje(@PathVariable(value = "id") String id,
@PathVariable(value = "idViaje") Integer idViaje){
      return new ResponseEntity(service.EliminarViaje(id,idViaje),HttpStatus.OK);
  }
}
```

Documentación de la API

La API RESTful del sistema de gestión de alquileres temporarios cuenta con múltiples endpoints organizados en módulos, cada uno diseñado para gestionar una funcionalidad específica del sistema. A continuación, se describe la estructura y funcionalidad de cada endpoint.

GET/users/lista

- Obtener la lista de usuarios

GET/users/{id}

- Obtener un usuario por su id

POST/users/nuevo

- Crear un nuevo usuario.

PUT/users/{id}/editar/

- Editar totalmente el documento de un usuario específico.

PATCH/users/{id}/patch

- Editar algún campo del documento de un usuario específico.

DELETE/users/{id}/borrar

- Eliminar un documento de un usuario específico

DELETE/users/{id}/borrarViaje/{idViaje}

- Eliminar un viaje de un documento de un usuario.

Frontend del Sistema

Tecnologías utilizadas para la interfaz de usuario

El frontend del sistema fue desarrollado utilizando tecnologías estándar para aplicaciones web modernas, asegurando accesibilidad y facilidad de uso. A continuación, se describen las herramientas principales empleadas:

Ejemplo de Integración con la API

El frontend se comunica con el backend a través de solicitudes HTTP utilizando el estándar RESTful. Esta integración permite al sistema realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre los datos almacenados en la base de datos.

Ejemplo: solicitud GET al endpoint /users/{id} para obtener el documento correspondiente a ese usuario

```
@Override
public Usuario ObtenerUsuario(String id) throws Exception {
  ApiFuture<DocumentSnapshot> future = getCollection().document(id).get();
  DocumentSnapshot document = future.get();
```

```
        if (document.exists()){
            return document.toObject(Usuario.class);
        }
        else {
            return null;
        }

    }
```

```
    private CollectionReference getCollection() {
        return firebase.getFirestore().collection("Users");
    }
```

Ventajas del Enfoque Adoptado

El enfoque de utilizar una API RESTful para la comunicación entre el frontend y el backend ofrece varias ventajas:

- **Modularidad**: La separación entre frontend y backend facilita el mantenimiento y futuras expansiones del sistema.
- **Escalabilidad**: El frontend puede adaptarse fácilmente a otras plataformas, como aplicaciones móviles, utilizando la misma API.
- **Reusabilidad**: Los componentes y funciones implementados en JavaScript pueden ser reutilizados en diferentes secciones de la aplicación.

Proceso de desarrollo del proyecto

El proceso de desarrollo del proyecto se empezó a llevar a cabo con la creación del proyecto utilizando Angular CLI. Además, se realizó la instalación y configuración completa de Firebase, con el propósito de gestionar la autenticación, las bases de datos y el almacenamiento de archivos en la nube.

Se implementó el servicio de Cloud Firestore para gestionar los datos en tiempo real, y se realizaron pruebas iniciales para la inserción de datos, como usuarios, ofertas de vuelos y ofertas de alojamientos, verificando su correcta visualización en tiempo real en la aplicación. Asimismo, se configuró y desarrolló cada componente en formato standalone y se empaquetó cada componente de forma separada, al igual que los servicios desarrollados.

Se desarrollaron los primeros componentes como el Registro y Login utilizando autenticación por correo electrónico y contraseña a través de Firebase Authentication. También se implementó la funcionalidad para permitir la creación de viajes, así como su modificación o eliminación, y la personalización de los datos personales del usuario.

Se mejoró la interfaz de usuario utilizando Bootstrap, logrando un diseño limpio, responsivo y fácil de usar. Además, se utilizaron otros frameworks como PrimeNG y PrimeFlex otorgando nuevos componentes a la interfaz para visualizar de una forma más ordenada la información obtenida por Apis.

En la página principal, se desarrolló la funcionalidad que permite al usuario buscar ofertas tanto de servicios de vuelo como servicios de alojamiento. Para esto se implementa la Api de Amadeus con un servicio que se encarga de realizar las peticiones a la Api y se muestra en el componente correspondiente. Una vez que el usuario selecciona las ofertas que

desea pasa a otra ventana donde debe resolver la forma de pago para confirmar la reserva y que sea almacenada en la base de datos

En una fase del proyecto, se evaluó la posibilidad de incorporar un conjunto de APIs externas diferentes a amadeus tales como Booking, SkyScanner, que son mundialmente conocidas pero al no contar con los recursos inmediatos para utilizarlas(se debe ser socio o estar suscrito y para ello hay que pagar) se decidió optar por una API externa como Amadeus que si bien no es muy conocida en Argentina, la misma otorga una modalidad de uso para la API en forma de Testing que permite a un desarrollador utilizar la API sin ningún pre requisito. Esto fue debido a que la idea principal del proyecto era lograr poder implementar las funcionalidades básicas propuestas en el alcance, por lo tanto, se decidió centrar los recursos en el desarrollo y prueba de funcionalidades esenciales, como el registro de usuarios, la gestión de viajes y la personalización de perfiles, con el objetivo de garantizar una experiencia de usuario fluida y sin interrupciones.

La funcionalidad de incluir itinerarios en los viajes y que el usuario tenga la posibilidad de subir sus viajes a un foro se ha planificado para futuras fases, lo que permitirá al usuario planificar aún más a fondo sus actividades y podrá compartir sus viajes con otros usuarios de su preferencia. Así mismo como el cambio de uso de Amadeus a otra API más eficiente con mejores tiempos de respuesta, sin tener un limitante de usos en sus endpoints, así como otras desventajas que vienen de la mano de su uso en el estado de Testing.

Manual de Usuario

En el siguiente capítulo se podrán observar las interfaces gráficas de la plataforma web, junto con una descripción de sus funcionalidades.

Página principal

Descripción de la Página Principal

La página de inicio está conformada por el formulario de inicio de sesión responsable de la autenticación de usuarios que controla el acceso a la plataforma. Se brinda la opción de iniciar sesión con mail y contraseña ya registrados o bien si aún no se ha registrado el usuario puede acceder al formulario de registro a través del botón “Registrarse” ubicado en la parte inferior derecha del formulario de inicio de sesión.

Formulario de Iniciar Sesión

Campos del formulario:

- Email: Campo de texto para que el usuario introduzca su dirección de correo electrónico.
- Contraseña: Campo de texto con formato de entrada oculta para que el usuario introduzca su contraseña.

- Botones:
  - Iniciar Sesión (Botón Azul): Botón principal que envía los datos de correo y contraseña para autenticar al usuario.

- Registrarse (Botón Gris): Botón secundario que redirige a los usuarios que aún no tienen cuenta al formulario de registro.

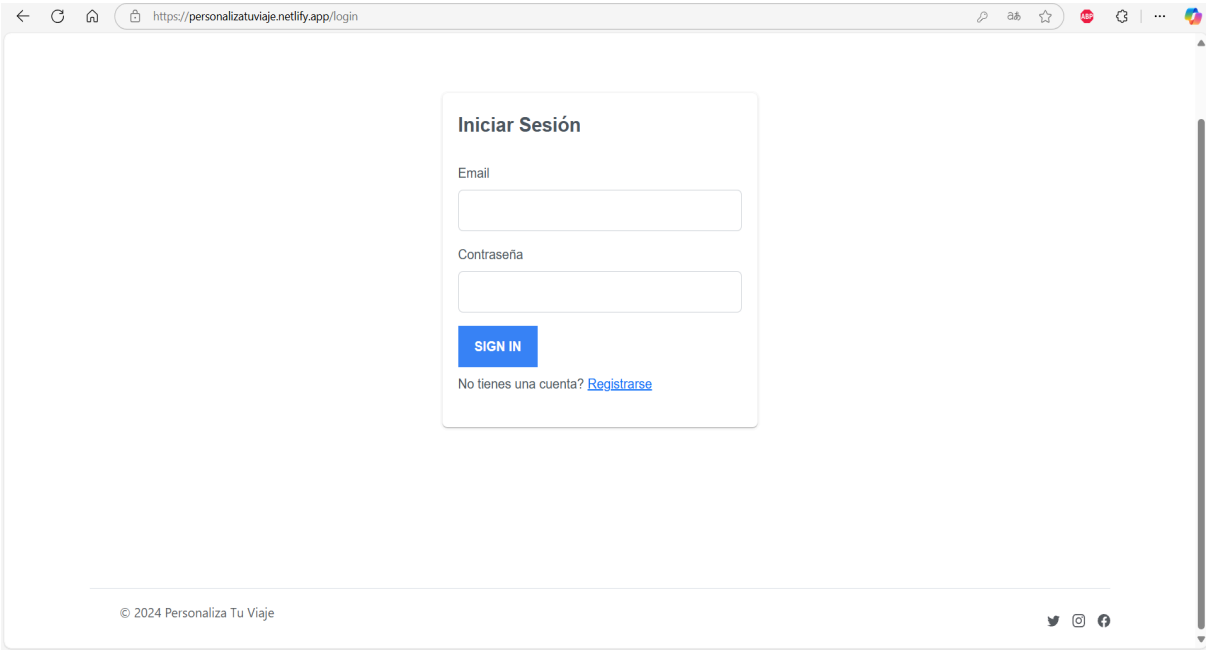
Formulario de Registro de Usuario

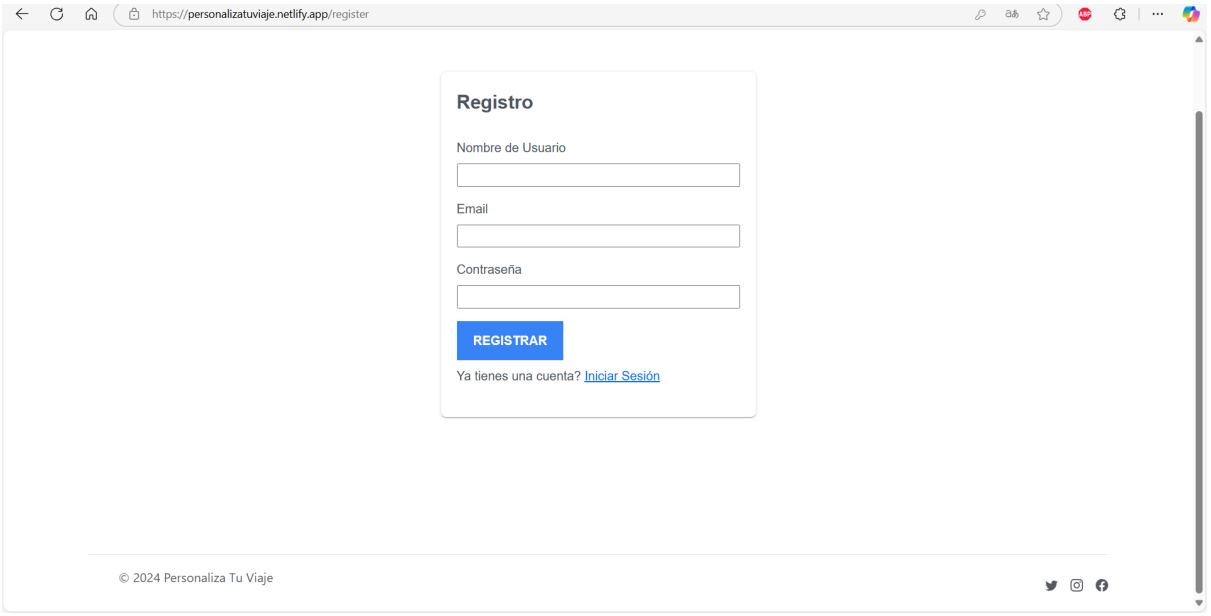
Campos del formulario:

- Email: Campo de texto para que el usuario introduzca su dirección de correo electrónico.
- Nombre de Usuario: Campo de texto para que el usuario introduzca el nombre por el cual quiere ser identificado dentro de la plataforma.
- Contraseña: Campo de texto con formato de entrada oculta para que el usuario introduzca su contraseña.

Botones:

- Registrarse (Botón Azul): Botón principal que envía los datos de correo y contraseña para autenticar al usuario.
- Iniciar Sesión (Botón Gris): Botón secundario que redirige a los usuarios que aún no tienen cuenta al formulario de registro.





Home

En la página de inicio se tiene dos formularios que se utilizan para la búsqueda tanto de ofertas de vuelo, desde un origen a un destino provisto por el usuario así como las fechas y

la cantidad de pasajes que desea, como también ofertas de alojamiento para la ciudad destino junto con las fechas correspondientes, la cantidad de personas y la cantidad de habitaciones. Luego de elegir las ofertas deseadas se muestra debajo un resumen de las ofertas para realizar el pago de la reserva y que se registre el cambio en la base de datos.

Formulario de búsqueda de ofertas de vuelos

Campos del formulario:

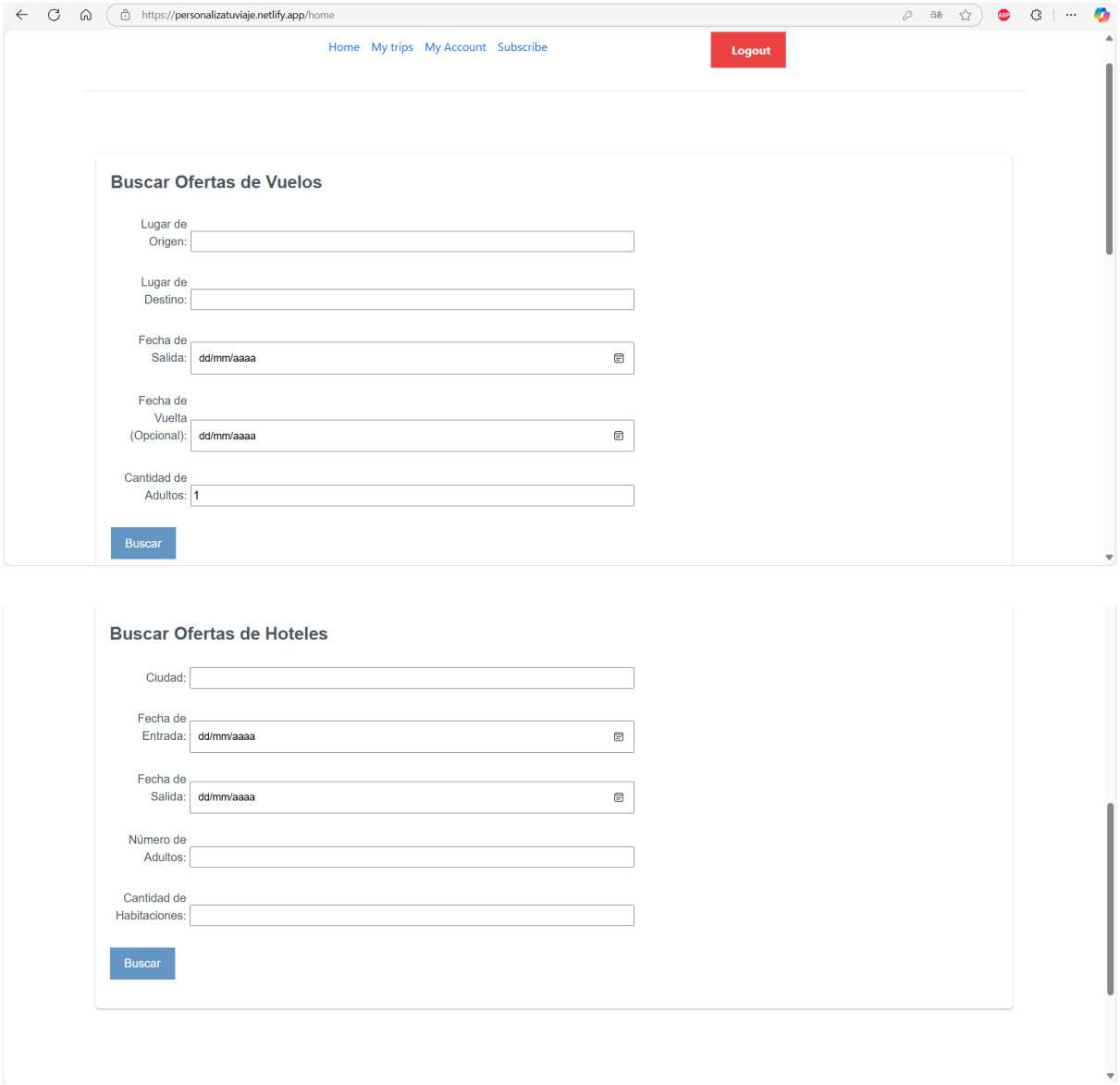
- Ciudad de Origen: Campo de texto donde se introduce el nombre de la ciudad desde donde saldrá el vuelo.
- Ciudad de Destino: Campo de texto donde se introduce el nombre de la ciudad a donde arribará el vuelo.
- Fecha de salida: Campo de texto para que el usuario introduzca la fecha de salida deseada para el vuelo
- Fecha de vuelta(Opcional): Campo de texto para que el usuario ingrese la fecha de regreso(si lo desea) para que se le muestren solamente ofertas que vienen en paquete con el pasaje de ida y vuelta, de omitir este campo solamente se le enseñarán ofertas con viajes de ida al destino ingresado.
- Adultos: Campo numérico para indicar la cantidad de pasajes que el usuario desea reservar(mínimo 1).

Botón: Buscar (Botón Azul): Botón que envía los datos ingresados al servicio que se ocupa de realizar la petición a la API de Amadeus para devolver las ofertas de vuelo disponibles con las condiciones ingresadas.

Formulario de búsqueda de ofertas de hoteles

Campos del formulario:

- Ciudad de destino: Campo de texto para que el usuario introduzca el nombre de la ciudad donde desea buscar un hotel para alojarse.
- Fecha de llegada: Campo de texto para que el usuario introduzca la fecha de llegada al hotel donde desea alojarse.
- Fecha de salida: Campo de texto para que el usuario introduzca la fecha de salida del hotel donde desea alojarse.
- Adultos: Campo numérico para que el usuario introduzca la cantidad de personas que se van a alojar
- Habitaciones: Campo numérico para que el usuario introduzca la cantidad de habitaciones deseadas para la cantidad de personas previamente ingresadas.



Lista de ofertas de Vuelos

Al ingresar los datos se manda la petición a la Api y se listan(si es que hay alguna oferta disponible con los parámetros ingresados) las ofertas de vuelo con un listado de ítems cómo el itinerario del viaje, la aerolínea, el número de vuelo, el precio del vuelo, características de la oferta como: número de valijas(sin cargo) y su peso, si viene con comida incluida, si se permite equipaje de mano, si el asiento es numerado, etc.

Lista de ofertas de Hoteles

Al ingresar los datos se manda la petición a la Api y se listan(si es que hay alguna oferta disponible con los parámetros ingresados) las ofertas de alojamiento con un listado de ítems cómo: nombre, ubicación, precio, características de la habitación.

Viajes

En la sección de viajes el usuario puede tanto visualizar cuántas reservas tiene activas de una forma clara y organizada a través de unos paneles que se contraen para poder visualizar solamente los viajes deseados. El usuario puede si quiere modificar los datos de sus viajes o bien borrarlos para lo cuál debería cancelar la reserva y estar atento a la política de cancelación de cada reserva ya que puede variar para cada compañía.

Lista de Viajes

Sección donde el usuario puede visualizar los viajes que ya ha reservado y tiene la opción de cancelar sus reservas.

Account

En esta sección, se tiene un formulario para que el usuario pueda completar sus datos personales y mantenerlos actualizados en todo momento. Además estos datos serán utilizados tanto para realizar las reservas así como también, en una mejora a futuro, para la sugerencia de ofertas que se ajusten a las preferencias que se indiquen en su perfil.

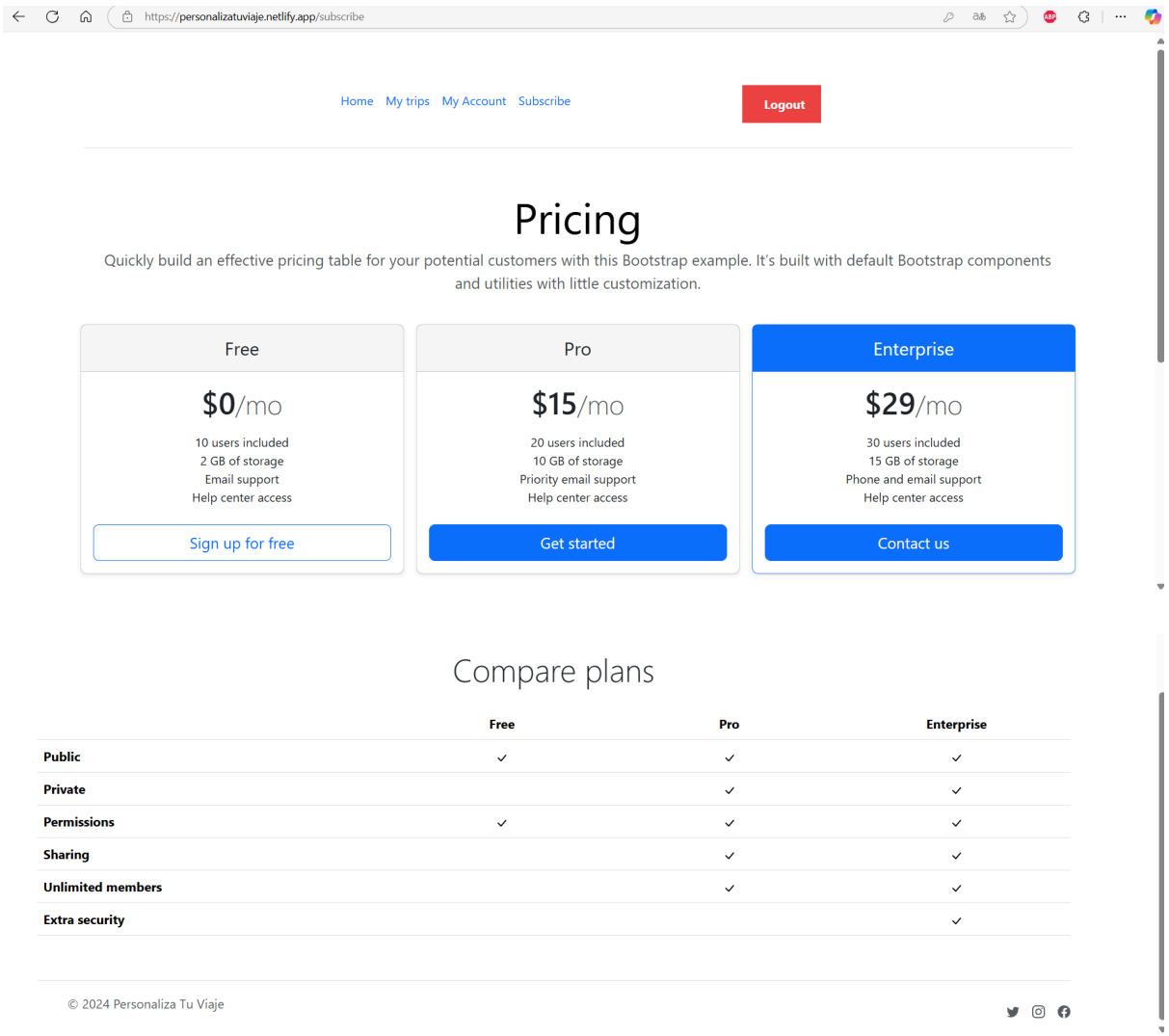
Formulario de Perfil de Usuario

Campos del formulario:

- Nombre de Usuario: Campo de texto para que el usuario modifique su nombre de usuario que será utilizado para identificarlo dentro de la plataforma.
- Email: Campo de texto para que el usuario modifique su email con el que realiza la autenticación para ingresar a la plataforma.
- Edad: Campo numérico para que el usuario ingrese o modifique su edad(mínimo 18).
- Género: Campo de texto para que el usuario ingrese o modifique su género.
- Presupuesto: Campo numérico para que el usuario ingrese o modifique un monto para utilizar de referencia para mostrar ofertas que se ajusten al mismo.
- Lista de Viajes: campo donde se visualizan los viajes que están activos en la cuenta del usuario en este momento.

Suscripción

En esta sección se visualiza una oferta que brinda la página de suscripción para acceder a beneficios que permitan a los usuarios utilizar la funcionalidad total de la plataforma que se planteó anteriormente, estas son por mencionar algunas: la opción de agregar un itinerario a los viajes, el uso de un foro donde publicar los viajes, un sistema que informe en tiempo real cambios en torno a la reserva realizada, etc.



Conclusiones

El desarrollo de la plataforma **"Personaliza Tu Viaje"** ha conseguido lograr un espacio dónde las personas puedan organizar de forma estructurada y ordenada sus viajes a través de la contratación de servicios de transportes y alojamientos. El diseño y la implementación han permitido implementar de forma exitosa las funciones esenciales, como el registro, la gestión de viajes y la personalización de perfiles.

En un futuro la plataforma servirá además para conectar a las personas a través de las publicaciones de sus viajes en un foro público.

Las repercusiones esperadas a partir de este trabajo incluyen no solo la creación de una comunidad activa sino también conectar personas a través de las publicaciones futuras. La plataforma está diseñada para mejorar la experiencia de la gestación del viaje y a partir de las publicaciones de itinerarios y actividades, poder conocer personas con intereses similares.

Desde una perspectiva empresarial, la plataforma ofrece un nuevo canal digital que facilita la promoción y venta de servicios relacionados con servicios turísticos. El sistema de gestión implementado garantiza la actualización y relevancia del contenido, lo que contribuye a mejorar la experiencia de los usuarios. De este modo, se espera que la plataforma impacte positivamente tanto a nivel individual como empresarial, promoviendo el crecimiento del mercado de actividades turísticas y afines.

En cuanto a la proyección a futuro, se prevé que la plataforma continúe evolucionando en función de las necesidades de sus usuarios. Se espera incorporar nuevas funcionalidades, como la capacidad de compartir itinerarios y actividades en foros. Estas mejoras, junto con una mayor personalización de los perfiles y recomendaciones basadas en los intereses de los usuarios, permitirán consolidar la plataforma.

"Personaliza Tu Viaje" fue pensada como una plataforma digital integral, capaz de adaptarse y crecer en el futuro. Con una mejora continua, este proyecto tiene un gran potencial de impacto en el mercado de actividades turísticas.

Referencias

- Documentación de angular: Home • Angular
- Documentación de PrimeNG: PrimeNG - Angular UI Component Library

- Documentación de PrimeFlex: Getting Started - PrimeFlex
- Documentación de Spring Boot: Spring | Home
- Documentación de Firebase: Firebase console
- Documentación de Amadeus: developers.amadeus.com
- Draw.io: https://app.diagrams.net/
- Netlify: https://app.netlify.com/
- GitHub: agusgioia/TrabajoFinal2024 agusgioia/TrabajoFinal2024-Java