# Introducción

Actualmente, una de las actividades de ocio más importantes es viajar. Teniendo en cuenta, la cantidad de turismo que tenemos tanto nacional como internacionalmente, se ha optado por el desarrollo de una aplicación que nos gestione un sistema de reservas.

Teniendo en cuenta las necesidades de nuestros clientes, se necesita una aplicación completa, donde podamos buscar vuelos, hoteles, seleccionar el que más nos guste y almacenarlo para ver las reservas que hemos realizado.

A lo largo de este documento, haremos referencia a los recursos necesarios que se han implementado para el desarrollo propio de la aplicación.

AÑADIR LOGO APP

# Objetivos

El principal objetivo de esta aplicación es darles a nuestros usuarios la mejor experiencia posible, gestionando las consultas y visualizaciones de las mejores ofertas y servicios que podamos proporcionar.

## Primarios

* Diseño de la interfaz gráfica de usuario.
* Integración de la API de Amadeus.
* Funcionalidades clave, como la búsqueda de vuelos, hoteles, etc.
* Gestión de las bases de datos.
* Autenticación de usuarios a través de Firebase.

## Secundarios:

* Optimización de la aplicación.
* Diseño funcional.
* Realización de pruebas de usuario.
* Gestión de itinerarios.

# Motivación

Debido a la alta demanda de servicios de viajes, se ha propuesto como objetivo la creación de una aplicación que haga más sencilla esta gestión. Gracias a las nuevas tecnologías nuestros usuarios van a obtener un mejor rendimiento si ofrecemos una aplicación móvil con todos los servicios que podamos ofrecer.

Es importante para el usuario, tener recogido en un mismo elemento todo lo necesario para poder así, cubrir las necesidades más importantes. Además de llevar un recuento de todas las reservas que ha realizado el usuario.

# Problemas a solucionar

Como se ha mencionado anteriormente, la alta demanda de viajes, hace que los usuarios a veces no encuentren todo lo necesario en una sola aplicación y tengan que desplazarse entre distintas aplicaciones para poder acceder a todos los datos necesarios.

Se pretende integrar la mayor oferta que podamos ofrecer al usuario para dar el servicio más cercano al usuario posible.

# Estudio de mercado

# Arquitectura del proyecto

Vamos a describir a continuación las herramientas que hemos utilizado para este proyecto:

## Tecnologías a utilizar

### Kotlin

Es un lenguaje de programación de código abierto y tipado estático. Es un lenguaje principalmente usado para la programación de aplicaciones de Android.

Algunas de las características y ventajas que nos puede proporcionar Kotlin son (Surra, 2023):

* Interoperabilidad con Java, una de las ventajas que nos proporciona este lenguaje es que es compatible con código escrito en Java e interactúan de manera correcta entre ambas.
* Curva de aprendizaje sencilla, como usuario que ha cambiado de Java a Kotlin, puedo decir que es sencillo de aprender, la sintaxis del código es muy lógica y es más fluida e intuitiva que Java, por ejemplo.
* Menor tiempo de programación, al ser un lenguaje mas compacto, una de las ventajas es que el código es más corto, no se repite tanto y no se crean funciones innecesarias.
* Corrutinas, además de ser un recurso que nos da Kotlin como lenguaje, a la hora de hacer llamadas en red o para bases de datos, es mucho más sencillo que otros lenguajes, con un par de líneas tienes la llamada necesaria.
* Desarrollo multiplataforma, podemos orientar nuestra app tanto a Android como a IOS o compartir código con otras plataformas.

### Firebase

Es una plataforma en la nube que nos permite el desarrollo de aplicaciones, nos facilita su creación y nos proporciona funciones sencillas e intuitivas para el desarrollo.

Funciones de Firebase utilizadas:

* Firestore como base de datos, los datos los almacenamos en tiempo real en la nube, el tipo de relación es NoSQL y la estructura que nos proporciona es la conocida como JSON.
* AuthFirebase como autenticación de usuarios. En la aplicación se ha utilizado tanto la autenticación con Google como con usuario y contraseña.

### Android Studio

Es el entorno de desarrollo integrado o IDE oficial de Android para el desarrollo de aplicaciones.

Utiliza herramientas como:

* Sistema de compilación flexible basado en el Gradle.
* Emuladores.
* Integración con GitHub.
* Compatibilidad integrada con Google Cloud Platform.

### Postman

Es una herramienta que nos permite realizar peticiones para las APIs de tipo REST.

Los métodos más utilizados son GET, POST, PUT y DELETE.

En nuestro caso solo se le ha dado uso a GET porque solo estamos realizando llamadas a las API necesarias.

## Librerías utilizadas

Firebase

Google Identity y Credential Manager

Retrofit

OkHttp

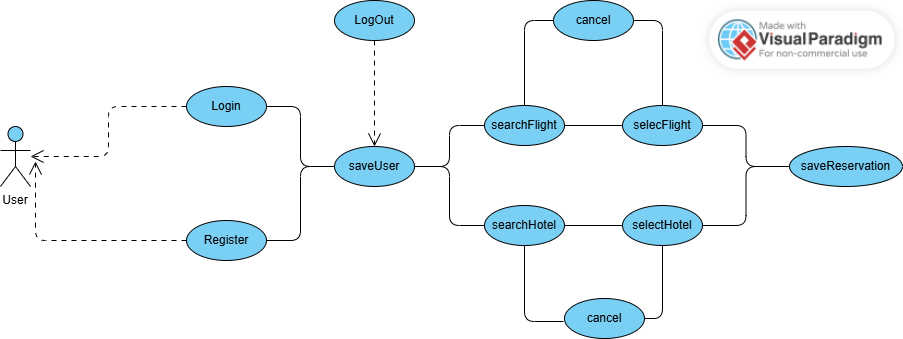
Corountines

Glide

Jetpack

# Requisitos funcionales

# Casos de uso



DESARROLLAR

# Base de datos

Como hemos comentado anteriormente, utilizamos Firebase como base de datos para almacenar lo necesario de nuestra aplicación y usuarios.

En Firebase utilizamos colecciones, subcolecciones y documentos.

En nuestro caso hemos dividido la base de datos en:

* Users, es la colección principal, donde se va a almacenar los datos del usuario en lo que llamamos documento, asi como sus reservas (subcolección). En esta se va a almacenar los hoteles como los vuelos que el usuario ha reservado.

NO SE COMO HACER LA REPRESENTACION!!!!!!

# Estructura del software

Login

MainActivity

Vuelos

* Home fragment
* FlightItems
* Summary
* Save

Hotel

* Gallery fragment
* HotelItems
* Summary
* Save

ITINERARIOS??????

# Soluciones de diseño adoptadas

Model – View – ViewModel

Model: modelos de datos y repositorios

View: actividades y fragmentos

ViewModel: shared view model

# Metodología de desarrollo

Scrum vs Kanban

# Planificación del proyecto

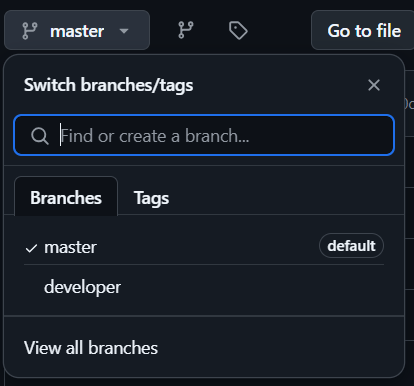
Gantt siii

# UML

# Herramientas de colaboración

Github

Es una plataforma basada en la nube donde podemos almacenar, compartir y trabajar junto con otros usuarios para el desarrollo de código.

Añadimos el código a lo que llamamos repositorios para poder presentar o compartir nuestro proyecto, administrar los cambios en el código y revisarlo. Y colaborar en un proyecto compartido con otros usuarios.

Para el desarrollo de nuestro proyecto, tenemos dos ramas, la principal y por defecto que es la master y una creada llamada developer. En la rama developer vamos a ir realizando los commits necesarios y subidas de proyecto mientras vamos integrando funcionalidades.

Mientras que en la rama master lo que vamos a hacer es tan solo tener las funcionalidades que están terminadas.

# Pruebas realizadas

Prueba de las api con Postman

# Manual de usuario

# Presupuesto??????

# Mejoras futuras

# Conclusión

# Bibliografía

<https://developer.android.com/identity/sign-in/credential-manager-siwg?hl=es-419>

<https://developers.amadeus.com/self-service>

<https://www.reddit.com/r/androiddev/comments/ip957p/best_way_to_pass_data_backwards_in_fragment/?tl=es-419&rdt=34455>

Surra, B. (2023, July 20). *El lenguaje de programación Kotlin: qué es y para qué sirve*. MyTaskPanel Consulting. https://www.mytaskpanel.com/lenguaje-de-programacion-kotlin/

Mora, S. L. (2022, October 4). *Firebase: qué es, para qué sirve, funcionalidades y ventajas*. DIGITAL55. https://digital55.com/blog/que-es-firebase-funcionalidades-ventajas-conclusiones/

Muradas, Y. (2022, May 12). Qué es Postman y primeros pasos. *OpenWebinars.net*. https://openwebinars.net/blog/que-es-postman/