

Memoria del proyecto

Route It: Aplicación Android colaborativa para la
realización de rutas

Trabajo de Fin de Grado

INGENIERÍA INFORMÁTICA



**VNIVERSIDAD
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

Enero 2023

Autor

Andrés Calvo Prieto

Tutores

André Filipe Sales Mendes

Gabriel Villarrubia González



VNiVERSiDAD
D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

Facultad D Ciencias
VNiVERSiDAD
D SALAMANCA



Certificado del/los tutor/es TFG

D./Dña. André Filipe Sales Mendes y Gabriel Villarrubia González,
profesores del Departamento de informática de la Universidad de Salamanca,

HACE/N CONSTAR:

Que el trabajo titulado

“Aplicación Android colaborativa para la realización de rutas”, que se presenta, ha sido realizado por D. Andrés Calvo Prieto, con DNI 70841193C y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado en ingeniería informática en esta Universidad.

Salamanca, 12 de enero de 2023

Fdo.: _____

Fdo.: _____

Agradecimientos

A mis grupos de amigos durante la carrera que me han acompañado y ayudado durante este camino, os llevo conmigo.

Sin olvidar a los santiagueses que mejor me trataron, gracias.

*Y en especial a mi padre, sin tu apoyo jamás lo hubiera
conseguido.*

Sé que estarías orgulloso de mí.

Resumen

Actualmente, cuando hablamos de rutas o usamos una aplicación de ruteo en nuestro dispositivo, seguramente se nos venga a la cabeza una ruta de senderismo a pie o con bicicleta, pues bien, aquí es donde hay que dar un golpe sobre la mesa y hacer constar que ruta es simplemente un camino que se toma para algún propósito y no tiene que ser necesariamente para el deporte.

El principal objetivo del proyecto es la creación de un sistema que de la libertad al usuario a crear rutas de todo tipo que el usuario crea oportunas, desde rutas gastronómicas hasta rutas simplemente para dar un paseo por las ubicaciones más históricas de nuestra ciudad.

Otro de los puntos a destacar es la eficiencia la hora de que los usuarios puedan planificar y hacer un buen uso de su tiempo, ya que siguiendo la ruta que ellos mismos u otros usuarios han creado, se evita la posibilidad de no malgastar nuestro tiempo.

O simplemente de dejarse llevar por el consejo de otros usuarios y seguir su ruta a la hora de visitar un nuevo lugar, por ejemplo, en el que nunca han estado así formando, una comunidad respetuosa en el que todo el mundo puede dar su conocimiento a otros y este puede ser de gran ayuda.

Es una buena forma de compartir tus rutinas, viajes o rutas al mundo en el cual podría haber otras personas que estarían interesadas, aplicación muy recomendada para los amantes de los viajes.

Palabras clave: rutas, planificación, viajes, ubicación, aplicación móvil, red social, compartir.

Summary

Currently, when we talk about routes or use a routing application on our device, surely a hiking or biking route comes to mind, well, here is where it is necessary to give a blow on the table and make it clear that a route is simply a path that is taken for some purpose and doesn't necessarily have to be for sport.

The main objective of the project is the creation of a system that gives the user the freedom to create routes of all kinds that the user deems appropriate, from gastronomic routes to routes just to have a walk through the most historic locations of our city.

Another point to highlight is the efficiency when users can plan and make good use of their time, since following the route that they themselves or other users have created, avoids the possibility of not wasting our time.

Or simply to be carried away by the advice of other users and follow their route when visiting a new place, for example, in which they have never been so forming a respectful community in which everyone can give their knowledge to others, and this can be of great help.

It is a good way to share your routines, trips, or routes to the world in which there could be other people who would be interested, highly recommended application for travel lovers.

Keywords: routes, planning, travel, location, mobile app, social network, sharing.

Tabla de contenidos

1.	Introducción.....	1
2.	Objetivos del proyecto.....	2
2.1.	Objetivos del sistema.....	2
2.2.	Objetivos personales.....	3
3.	Conceptos teóricos.....	4
3.1.	API.....	4
3.2.	SDK.....	5
3.3.	IDE.....	6
3.4.	Ubicación GPS.....	7
3.5.	Herramientas CASE.....	7
4.	Técnicas y herramientas.....	8
4.1.	Kotlin.....	8
4.2.	XML.....	9
4.3.	Android Studio.....	10
4.4.	Firebase.....	11
4.5.	Google Cloud Platform.....	11
4.6.	Herramientas CASE.....	12
4.6.1.	EZEstimate.....	12
4.6.2.	Microsoft Project.....	13
4.6.3.	Visual Paradigm.....	13
4.7.	Herramientas de generación de documentación.....	14
4.7.1.	KDoc.....	14
4.7.2.	Dokka.....	14
5.	Aspectos relevantes del desarrollo del proyecto.....	15
5.1.	Marco de trabajo.....	15
5.2.	Estimación de costes y esfuerzo.....	16
5.3.	Planificación temporal.....	17
5.4.	Especificación de requisitos.....	19
5.4.1.	Participantes.....	19
5.4.2.	Objetivos del sistema.....	19
5.4.3.	Requisitos de información.....	20
5.4.4.	Requisitos de restricción.....	21
5.4.5.	Definición de actores.....	21
5.4.6.	Requisitos funcionales.....	22

5.4.7.	Requisitos no funcionales	24
5.5.	Análisis de requisitos	25
5.5.1.	Modelo de dominio.....	25
5.5.2.	Realización de casos de uso de análisis	26
5.5.3.	Clases de análisis	27
5.5.4.	Vista arquitectura del modelo de análisis	27
5.6.	Diseño del sistema	28
5.6.1.	Patrón arquitectónico MVC.....	28
5.6.2.	Subsistemas de diseño	29
5.6.3.	Clases de diseño	30
5.6.4.	Realización de casos de uso de diseño	31
5.6.5.	Diseño de la base de datos	32
5.6.6.	Modelo de despliegue	33
5.7.	Implementación	34
5.8.	Pruebas	35
5.9.	Funcionalidad del sistema.....	36
5.9.1.	Acceso a la aplicación	36
5.9.2.	Perfil de usuario y configuración.....	39
5.9.3.	Búsqueda e interacción de usuarios.....	43
5.9.4.	Creación e interacción de rutas.....	46
6.	Conclusiones y líneas de trabajo futuras	53
6.1.	Conclusiones	53
6.2.	Líneas de trabajo futuras	54
7.	Referencias	55

Lista de tablas

Tabla 1: Participante Calvo Prieto, Andrés	19
Tabla 2: OBJ-0003 Gestión de rutas	20
Tabla 3: IRQ-0001 Datos de usuario	20
Tabla 4: ACT-0002 Usuario identificado	21
Tabla 5: UC-0015 Añadir ruta.....	23
Tabla 6: NFR-0003 Escalabilidad	24

Lista de figuras

Figura 1: Funcionamiento de una API.....	4
Figura 2: SDK.....	5
Figura 3: IDE.....	6
Figura 4: Funcionamiento del sistema GPS	7
Figura 5: XML.....	9
Figura 6: Android Studio.....	10
Figura 7: EZEstimate.....	12
Figura 8: Microsoft Project.....	13
Figura 9: Visual Paradigm.....	13
Figura 10: Comentario de bloque en KDoc.....	14
Figura 11: Dokka.....	14
Figura 12: Fases del Proceso Unificado	15
Figura 13: Estimación de esfuerzo en EZEstimate.....	16
Figura 14: Planificación temporal del proyecto	17
Figura 15: División de tareas y diagrama de Gantt	18
Figura 16: Diagrama de paquetes	22
Figura 17: Diagrama de gestión de usuarios	22
Figura 18: Diagrama de clases modelo de dominio	25
Figura 19: Diagrama de secuencia de análisis Añadir ruta	26
Figura 20: Arquitectura del modelo de análisis.....	27
Figura 21: Patrón arquitectónico MVC	28
Figura 22: Subsistemas de diseño.....	29
Figura 23: Clase de diseño del paquete Activities.....	30
Figura 24: Diagrama de secuencia de diseño Añadir ruta.....	31
Figura 25: Diseño de la base de datos	32
Figura 26: Diagrama de despliegue	33
Figura 27: Pantalla de inicio de sesión	36
Figura 28: Pantalla registro.....	37
Figura 29: Pantalla restablecer contraseña	38
Figura 30: Formulario cambiar contraseña.....	38
Figura 31: Pantalla Perfil.....	39
Figura 32: Pantalla editar perfil	40
Figura 33: Menú del perfil de usuario	41
Figura 34: Pantalla cambiar contraseña.....	42
Figura 35: Pantalla Búsqueda.....	43
Figura 36: Búsqueda de usuarios.....	44
Figura 37: Perfil público de un usuario	45
Figura 38: Creación de ruta	46
Figura 39: Pantalla principal Home.....	47
Figura 40: Comentarios de la ruta	48
Figura 41: Pantalla rutas personales	49
Figura 42: Compartir ruta.....	50
Figura 43: Ubicaciones de la ruta	51
Figura 44: Detalles de la ubicación	52

1. Introducción

Actualmente, hay innumerables aplicaciones de rutas, pero una vez que entras en detalle, verás que la mayoría de ellas, por no decir la totalidad, las cuales tienen éxito, son las centradas en senderismo. Pues lo que se busca con este proyecto es la creación de un sistema software, el cual da la libertad al usuario de poder crear rutas distintas o guías de cualquier tipo, desde viajes a rutas gastronómicas, pasando por guías turísticas, y así poder tener una planificación de lo que desee y no pierda tiempo por el camino.

En este documento se van a detallar los aspectos más relevantes en el desarrollo del proyecto software. La estructura será la siguiente:

- **Objetivos del proyecto:** Se explican los objetivos principales que se desean lograr con la realización del proyecto.
- **Conceptos teóricos:** Se detallan los conceptos teóricos necesarios para comprender correctamente el proyecto y su funcionamiento.
- **Técnicas y herramientas:** Se detallan las técnicas y herramientas utilizadas para la realización del proyecto software.
- **Aspectos relevantes del desarrollo del proyecto:** Se recogen los aspectos más importantes durante el desarrollo del proyecto software.
- **Conclusiones y líneas de trabajo futuras:** Se da una conclusión final tras la finalización del proyecto software y se habla sobre las posibles líneas de trabajo futuro que se podrían hacer sobre el mismo.
- **Bibliografía:** Se referencian las fuentes empleadas.

Este documento se ve complementado por los siguientes anexos:

- **Anexo I - Plan del proyecto software:** Se detalla el plan del proyecto software junto con una estimación de duración y esfuerzo de este.
- **Anexo II - Especificación de requisitos del software:** Se detalla la especificación de requisitos software del sistema.
- **Anexo III - Especificación de diseño:** Se recoge la documentación sobre la fase de análisis y diseño del sistema.
- **Anexo IV - Documentación técnica de programación:** Se facilita una serie de técnicas para la correcta comprensión del código del sistema.
- **Anexo V - Manual de usuario:** Se explica a forma de guía el funcionamiento del sistema al usuario para que pueda utilizarlo correctamente.

2. Objetivos del proyecto

A continuación, se detallarán los principales objetivos que el proyecto debe cumplir para su correcto desarrollo.

2.1. Objetivos del sistema

El principal objetivo del sistema es el desarrollo de una aplicación, la cual actúe como una red social de publicaciones en forma de rutas, guiándose por distintas ubicaciones, con su respectivo sistema de valoraciones y comentarios, como de un filtro de búsqueda.

- **Gestión de usuarios:** El sistema deberá gestionar las peticiones de los usuarios al querer darse de alta, darse de baja, visualizar perfiles y modificar datos personales, entre otras opciones relacionadas con los usuarios.
- **Gestión de ubicaciones:** El sistema deberá gestionar las ubicaciones que conforman una ruta junto a un buscador de estas y ser capaz de mostrar los detalles de cada ubicación.
- **Gestión de rutas:** El sistema deberá gestionar las rutas creadas por los usuarios almacenándolas en la base de datos y mostrándolas correctamente en las pantallas correspondientes.
- **Gestión de filtrado y búsqueda:** El sistema deberá gestionar correctamente las peticiones de los usuarios a la hora de que estos pidan filtrar rutas o buscar otros usuarios.
- **Gestión de sistemas:** El sistema deberá de ser capaz de gestionar con eficiencia y utilizar correctamente los distintos sistemas que hará uso la aplicación, como el sistema GPS.

2.2. Objetivos personales

Aparte de los objetivos del proyecto y todo su desarrollo, también hay unos objetivos personales que me gustaría remarcar desde el primer paso que fue escoger tema del proyecto hasta la finalización de este:

- Desde el principio siempre quise desarrollar desde cero una aplicación, ya que supondría un reto para mí, dado que hasta el momento nunca lo había hecho y sería de gran ayuda para mi formación.
- Como también lo ha sido aprender un entorno de desarrollo nuevo para mí como Android Studio y un nuevo lenguaje de programación, Kotlin. Sin olvidarme de aprender y utilizar tecnología de Google como lo han sido la SDK de Google Maps, API de direccionamiento y ubicaciones, la plataforma de desarrollo Firebase, etc.
- El desafío que supone plantear, analizar, diseñar y desarrollar un sistema completo totalmente solo, sin ayuda de compañeros con división de trabajo como en asignaturas de la carrera, teniendo que hacer una planificación de todos los procesos del proyecto para así dividir el tiempo.
- Utilizar todos mis conocimientos previos de mi formación para el trabajo de un nuevo proyecto que finaliza mi etapa para obtener el título.

3. Conceptos teóricos

En este apartado se van a explicar los conceptos utilizados durante el desarrollo del proyecto para lograr una fácil comprensión de este.

3.1. API

El término API se corresponde con la abreviatura, en inglés, application programming interface, lo que significa interfaz de programación de aplicaciones. Se definen como un conjunto de definiciones y protocolos que permiten desarrollar software de las aplicaciones.

Funcionan como una comunicación entre servicios o componentes de software sin necesidad de como estos estén implementados, esto ofrece una gran flexibilidad y libertad haciendo más fácil el desarrollo del software.

En la *Figura 1* se puede observar, de una forma con un ejemplo real, el funcionamiento de una API.

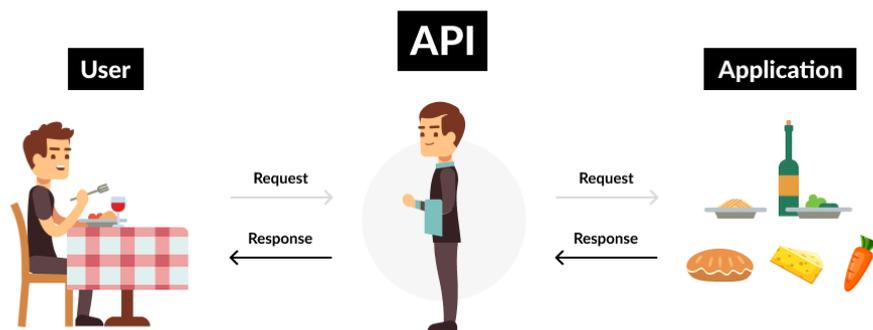


Figura 1: Funcionamiento de una API

3.2. SDK

SDK es el acrónimo de “Software Development Kit”, en español kit de desarrollo de software. Se define como un conjunto de herramientas software ofrecidas para el desarrollo de aplicaciones para cierta plataforma, sistema o lenguaje específicos.

Este puede incluir algo tan sencillo como una API o también algo más completo como muestras de código de ejemplo para que los desarrolladores tengan una guía de la que partir.

En la *Figura 2* se puede observar los componentes de un SDK.

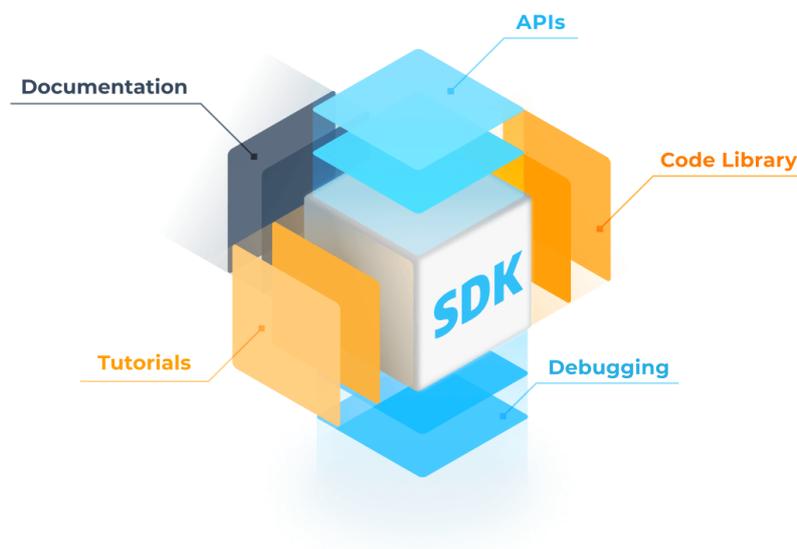


Figura 2: SDK

3.3. IDE

Un entorno de desarrollo integrado (IDE) es un sistema o aplicación informática que ofrece una serie de herramientas o servicios para facilitar el desarrollo de software. Suele consistir de las siguientes características:

- **Editor de código fuente:** Editor de texto diseñado para escribir el código de software, puede contar con herramientas y ayudas al desarrollador para facilitar el proceso.
- **Automatización de las compilaciones locales:** Herramientas de construcción automáticas de tareas sencillas y repetitivas.
- **Depurador:** Programa que sirve para identificar y mostrar la ubicación de los errores en software.

En la *Figura 3* se puede observar el diseño de un IDE.

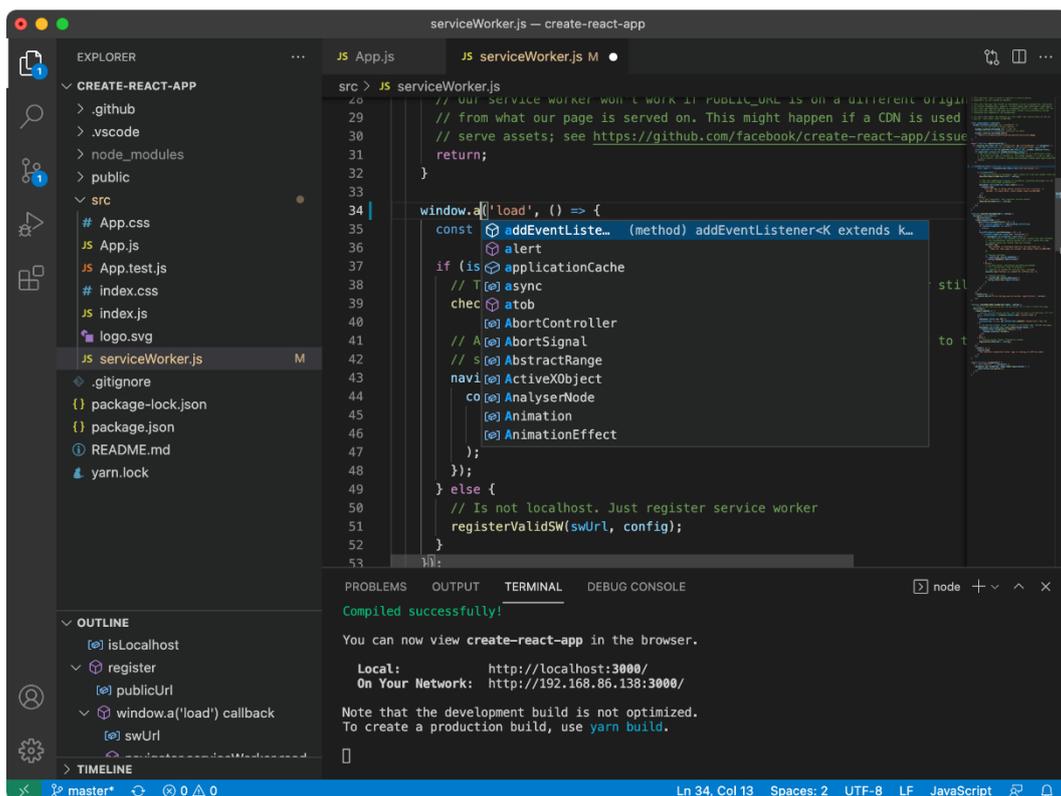


Figura 3: IDE

3.4. Ubicación GPS

El Sistema de Posicionamiento Global (GPS) es un sistema el cual permite localizar y ubicar cualquier objeto o persona sobre la Tierra. Esto puede ser desde casos básicos ya comentados hasta casos muy amplios, desde buscar una ubicación hasta el transporte y navegación de cualquier elemento.

Para calcular la ubicación se necesitan tres satélites mínimo y un equipo dotado de receptor y transmisor GPS, como también de unas estaciones de control terrestres.

El sistema hace uso de una técnica llamada trilateración, el satélite emite una señal que llega al receptor a una hora específica y esto se traduce a una distancia del satélite en concreto formando un círculo con las posibles ubicaciones del receptor. Esto mismo se realiza en dos satélites más y con el resultado de estos tres, se genera un solo punto que es la ubicación exacta del receptor, aunque se suele añadir un cuarto satélite para calcular la tercera dimensión del receptor y la altitud de este, a la par que también valida la información de los otros satélites empleados.

En la *Figura 4* se puede observar el funcionamiento del sistema GPS.

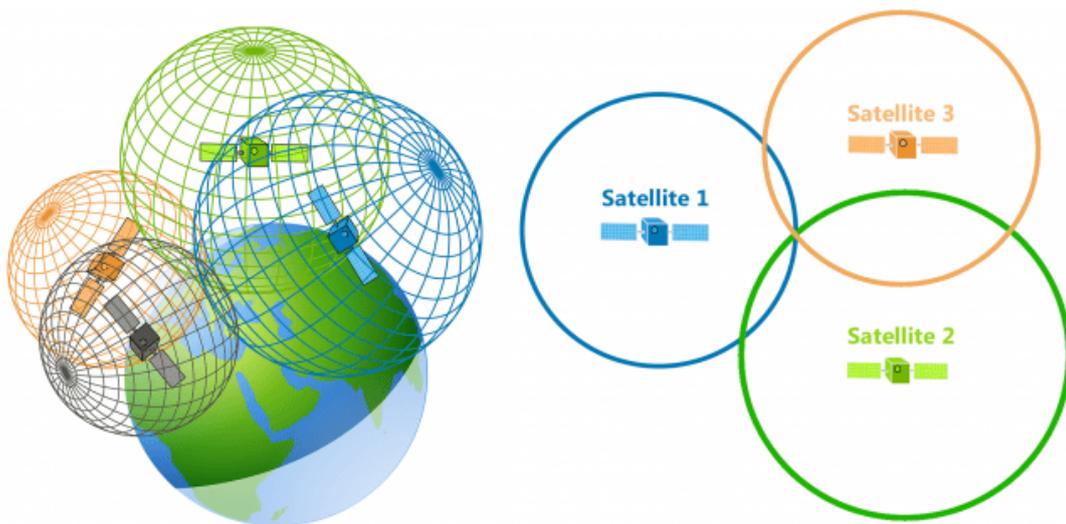


Figura 4: Funcionamiento del sistema GPS

3.5. Herramientas CASE

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora) son aplicaciones informáticas cuya finalidad es la de facilitar el desarrollo de aplicaciones software en términos de tiempo y costo.

4. Técnicas y herramientas

En este apartado se van a explicar las técnicas y herramientas utilizadas durante el desarrollo del proyecto.

4.1. Kotlin

Kotlin es un lenguaje de programación orientada a objetos que corre sobre la máquina virtual Java. En los últimos años, el lenguaje por excelencia para programar aplicaciones en Android era Java, pero surgió Kotlin con bastantes mejoras y Google no dudó en apostar en él y convertirlo en su lenguaje oficial en Android, recomendando usarlo a los desarrolladores en su propio IDE, Android Studio.

Tanto es así que ya supera el 60% de las aplicaciones creadas con este nuevo lenguaje de programación, algunos ejemplos son Netflix, Twitter, Duolingo, Adobe Acrobat, Airbnb, WordPress, etc. Muchas de las aplicaciones están mudando el cambio de lenguaje de programación en su código base de Java a Kotlin.

Algunas de las características de Kotlin son:

- Es interoperable al 100% con Java, esto quiere decir que en un proyecto Java puede utilizar Kotlin perfectamente, pueden coexistir ambos lenguajes.
- Es un lenguaje más seguro, todos sabemos la de quebraderos de cabeza que nos ha dado Java con el puntero a nulo y no encontrábamos donde este se ubica, pues con Kotlin podemos definir una variable 'Nullable' sin importar su tipo, lo cual quiere decir que esta puede tener un valor nulo perfectamente si lo deseamos.
- Es más rápido y conciso, pues necesita de un 40% menos de líneas de código que Java para la misma función, por lo que nos ahorramos mucho trabajo.
- Es fácil de aprender, si tienes experiencia con Java, no te resultará un problema adaptarte a Kotlin, ya que es similar.

4.2. XML

XML es un metalenguaje de marcado extensible, como indican sus siglas, que define un conjunto de reglas para la codificación de documentos. Suelen contener mucha información de cualquier tipo y se componen de etiquetas individuales o anidadas.

En la *Figura 5* se puede observar un ejemplo del metalenguaje XML utilizado en el desarrollo del proyecto.

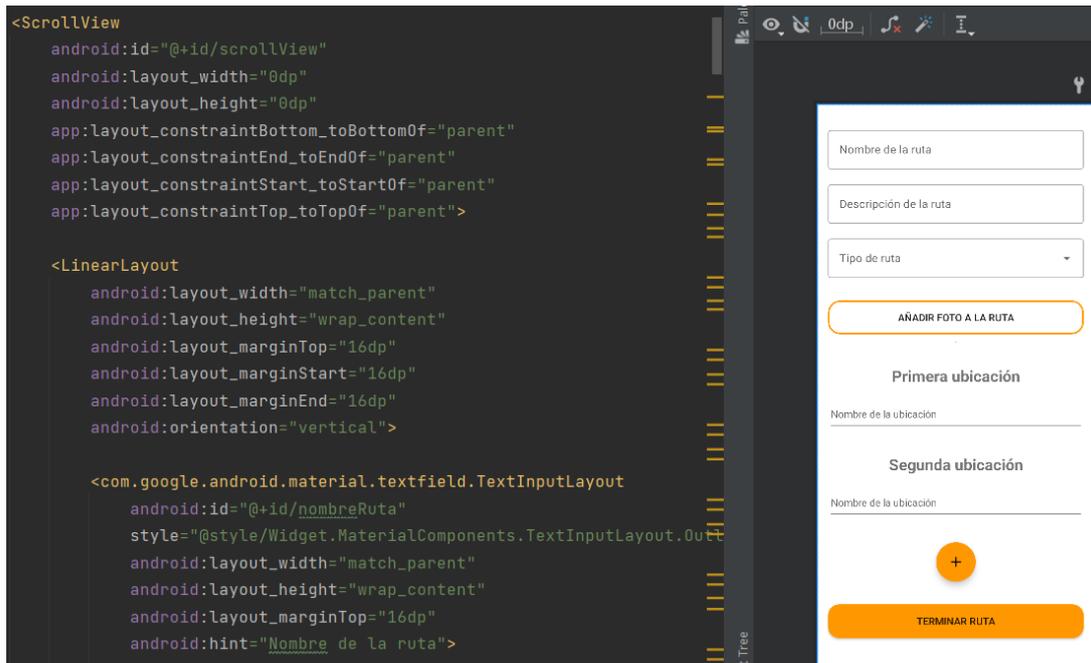


Figura 5: XML

4.3. Android Studio

Android Studio es el IDE, entorno de desarrollo integrado, oficial para la plataforma Android desarrollado por Google. Dispone de un emulador rápido para probar la aplicación tanto en tu dispositivo móvil físico como en uno virtual que ofrece el IDE, también posee de un editor de código inteligente con el que puedes escribir código de manera más eficaz y rápida.

Al ser propiedad de Google, ofrece una gran compatibilidad e integración con plataformas de esta, como bases de datos, SDK y muchas más que ayudan al usuario a desarrollar su sistema. Por no hablar de la optimización para los dispositivos Android que posee, desde dispositivos móviles, televisores Android, relojes inteligentes y mucho más.

En la *Figura 6* se puede observar el diseño de la IDE Android Studio.

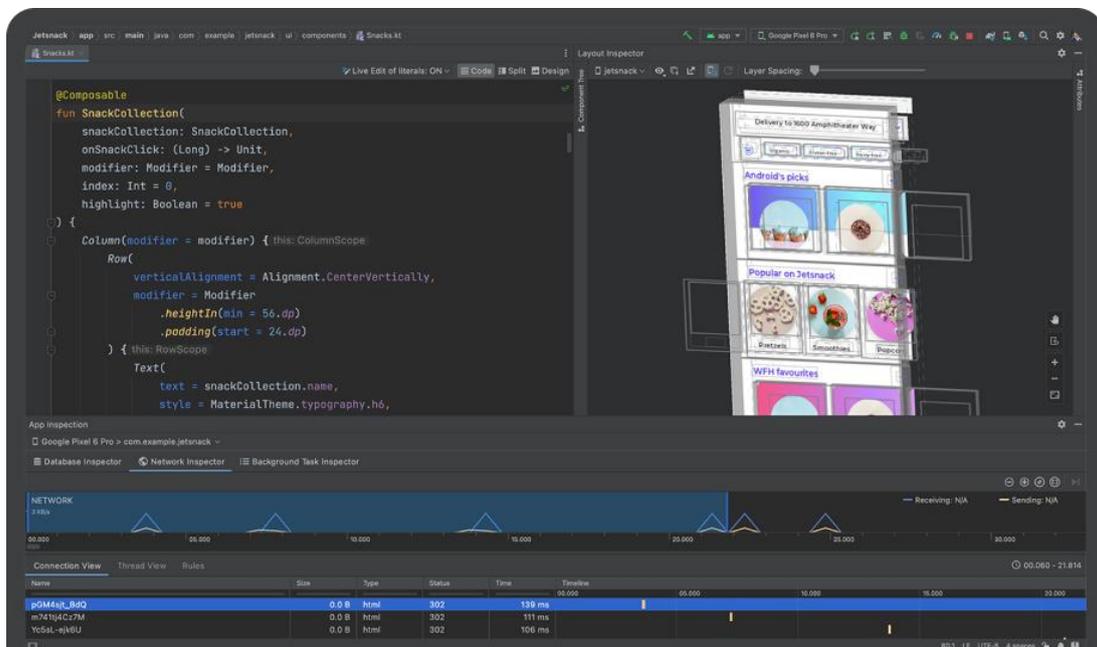


Figura 6: Android Studio

4.4. Firebase

Firebase es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles dirigida por Google. Las funciones que nos ofrece la plataforma que hará uso el proyecto son las siguientes:

- **Authentication:** Permite la creación de usuarios y la autenticación de estos mediante correo y contraseña, los cuales se vinculan a su cuenta que luego podrán editar si lo desean, almacenando los datos con la mayor seguridad que ofrece Google.
- **Cloud Firestore Database:** Base de datos no relacional con almacenamiento de datos simples en la nube donde se guardarán los datos de los usuarios y de las rutas.
- **Storage:** Base de datos la cual servirá para almacenar las imágenes que los usuarios suban como foto de perfil o foto de la ruta creada.

4.5. Google Cloud Platform

Google Cloud es una plataforma que reúne todas las aplicaciones de desarrollo web de Google, todas almacenadas en la nube. Destaca por su rapidez y escalabilidad, serán unos aspectos importantes por lo que se hará uso de dicha plataforma en el proyecto.

Esta plataforma nos brindará las bibliotecas necesarias para poder manejar las ubicaciones correctamente, los servicios que utilizaremos serán los siguientes:

- **Directions API:** Servicio web utilizado para mostrar instrucciones en formato JSON entre ubicaciones y así poder crear una trayectoria entre estos.
- **Places API:** Servicio utilizado para obtener los datos de las ubicaciones y agregar el autocompletado al buscar las mismas al crear las rutas.
- **Maps SDK for Android:** Servicio utilizado para agregar mapas a nuestro proyecto y facilitar la interacción con el usuario al agregar marcadores, entre otras cosas.

4.6. Herramientas CASE

Se van a explicar las herramientas CASE que se harán uso durante el desarrollo del proyecto.

4.6.1. EZEstimate

Herramienta para la estimación del esfuerzo de desarrollo software utilizando la metodología de puntos de caso de uso para la creación de los informes de estimación.

En la *Figura 7* se puede observar la interfaz de la herramienta EZEstimate.

File Settings Help

Module

Actores

Summary

Total Modules **Excel Report**

Use cases Simple Average Complex

Actors Simple Average Complex

Add Actor / Use case

Actor / Use case Name Select Type Complexity

Tech / Env Factors

Estimation Summary

UAW

UUCW

UUCP = UAW + UUCW

TFactor

EFactor

TCF = 0.6 + (.01*TFactor)

EF = 1.4 + (-0.03*EFactor)

UCP = UUCP*TCF*EF

Total Effort@ Hrs/UCP

Use case / Actor List (Double click to delete)

Id	Module	Type	Name	complexity
1	Actores	Actor	ACT-0001 Usu...	Complex
10	Gestión de usu...	Usecase	UC-0007 Cambi...	Simple
11	Gestión de usu...	Usecase	UC-0008 Cerrar...	Simple
12	Gestión de usu...	Usecase	UC-0009 Cambi...	Simple
13	Gestión de usu...	Usecase	UC-0010 Elim...	Simple
14	Gestión de usu...	Usecase	UC-0011 Ayuda	Simple
15	Gestión de usu...	Usecase	UC-0012 Listar ...	Simple
16	Gestión de usu...	Usecase	UC-0013 Búsqu...	Simple
17	Gestión de usu...	Usecase	UC-0014 Ver p...	Simple
18	Gestión de rutas	Usecase	UC-0015 Añadir...	Average
19	Gestión de rutas	Usecase	UC-0016 Añadir...	Simple
2	Actores	Actor	ACT-0002 Usu...	Complex
20	Gestión de rutas	Usecase	UC-0017 Añadir...	Average
21	Gestión de rutas	Usecase	UC-0018 Elim...	Simple
22	Gestión de rutas	Usecase	UC-0019 Comp...	Simple
23	Gestión de rutas	Usecase	UC-0020 Valora...	Simple
24	Gestión de rutas	Usecase	UC-0021 Ver c...	Simple
25	Gestión de rutas	Usecase	UC-0022 Añadir...	Simple

Figura 7: EZEstimate

4.6.2. Microsoft Project

Software de gestión de proyectos que permite tener un control sobre todos los procesos y tareas que conforman el desarrollo de un proyecto para conseguir una planificación óptima.

En la *Figura 8* se puede observar la interfaz de la herramienta Microsoft Project.

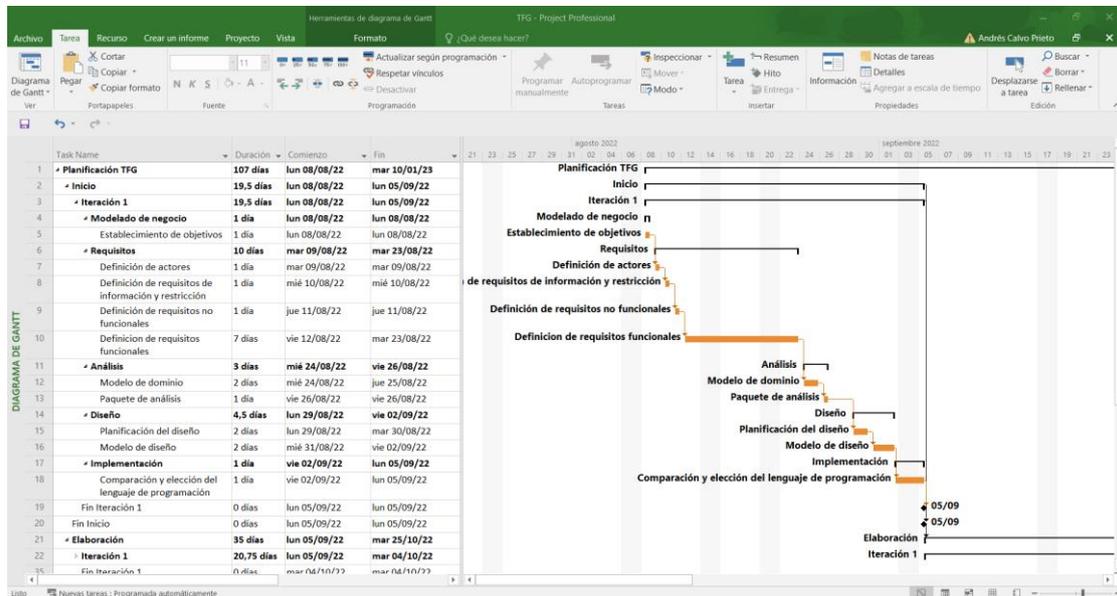


Figura 8: Microsoft Project

4.6.3. Visual Paradigm

Herramienta UML CASE empleada para representar mediante todo tipo de diagramas el ciclo de vida completo del proceso de desarrollo del software, desde la planificación, hasta el análisis y el diseño.

En la *Figura 9* se puede observar la interfaz de la herramienta Visual Paradigm.

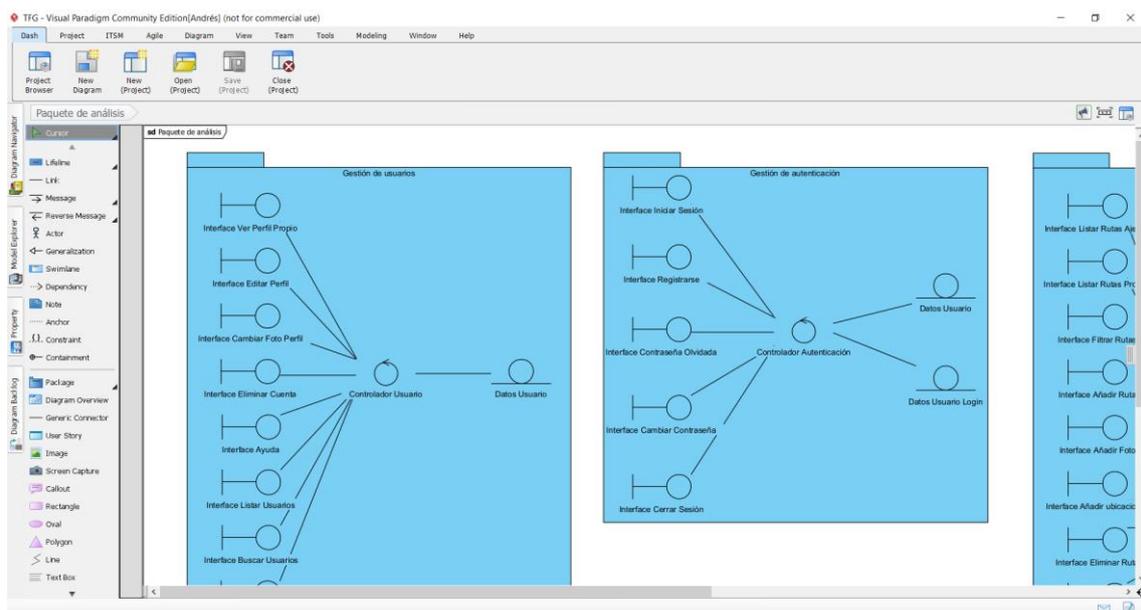


Figura 9: Visual Paradigm

4.7. Herramientas de generación de documentación

Para recoger toda la documentación generada durante el desarrollo de programación mediante Kotlin, se han usado dos herramientas complementarias.

4.7.1. KDoc

KDoc es un lenguaje para documentar el código Kotlin en proyectos software mediante una sintaxis de etiquetas de bloque y Markdown para marcado en línea.

En la *Figura 10* se puede observar la sintaxis que usa KDoc.

```
/**
 * Actividad que permite al usuario crear una nueva ruta
 * @author Andrés Calvo Prieto
 * @since 1.0
 * @constructor Constructor principal que establece las variables de referencia
 */
```

Figura 10: Comentario de bloque en KDoc

4.7.2. Dokka

Dokka es la herramienta de generación de documentación que recoge los comentarios de bloque y línea que son documentados mediante KDoc y genera la documentación en varios formatos, como HTML, para una visualización y comprensión más sencilla y organizada.

En la *Figura 11* se puede observar la documentación que crea la herramienta Dokka.

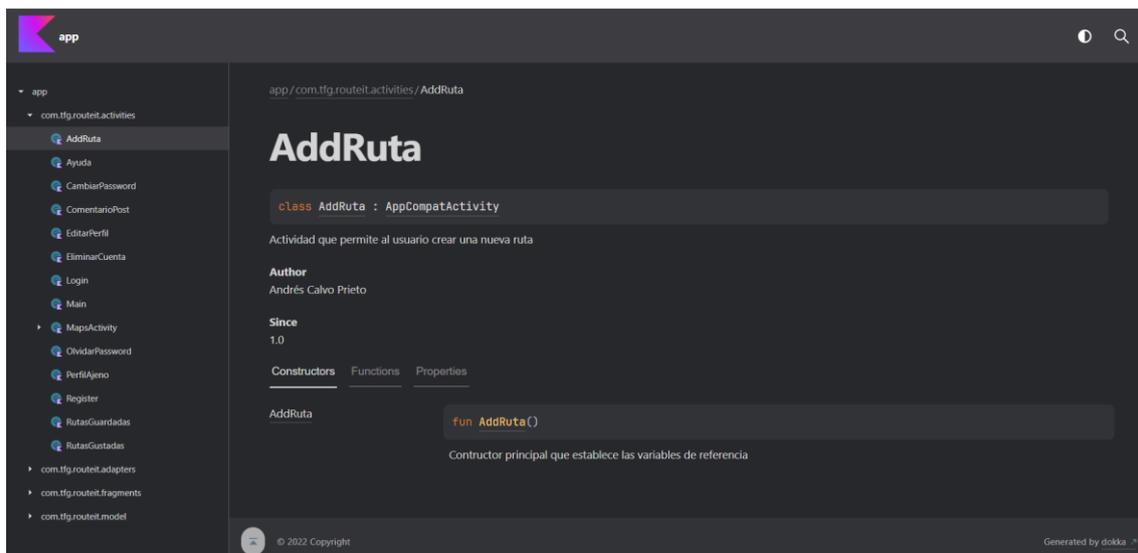


Figura 11: Dokka

5. Aspectos relevantes del desarrollo del proyecto

En esta sección se va a recoger los aspectos más relevantes del desarrollo del proyecto software.

5.1. Marco de trabajo

Durante el desarrollo del proyecto se ha seguido un marco de trabajo del Proceso Unificado caracterizado por los siguientes puntos:

- **Dirigido por casos de uso:** Se utilizan los casos de uso para capturar los requisitos funcionales y definir los contenidos de cada iteración.
- **Centrado en arquitectura:** Este marco de trabajo asume que no existe un único modelo que cubre todos los aspectos del sistema, sino que coexisten múltiples modelos y vistas que definen la arquitectura de software del sistema.
- **Iterativo e incremental:** El Proceso Unificado está compuesto por cuatro fases, y estas están divididas en una serie de iteraciones que ofrecen como resultado un incremento del producto desarrollado, añadiendo o mejorando las funcionalidades de este.

Y estas cuatro fases son las siguientes:

- **Inicio:** Se define el alcance del proyecto, los objetivos y se desarrollan los casos de negocio.
- **Elaboración:** Se elabora la planificación del proyecto, se entra en detalle y se especifican los casos de uso y se define la arquitectura del sistema.
- **Construcción:** Se construye el producto definido.
- **Transición:** Se prueba, instala y el usuario usa el producto final. Se corrigen errores y se piensan futuras novedades para el producto.

En la *Figura 12* se pueden observar las fases del Proceso Unificado.

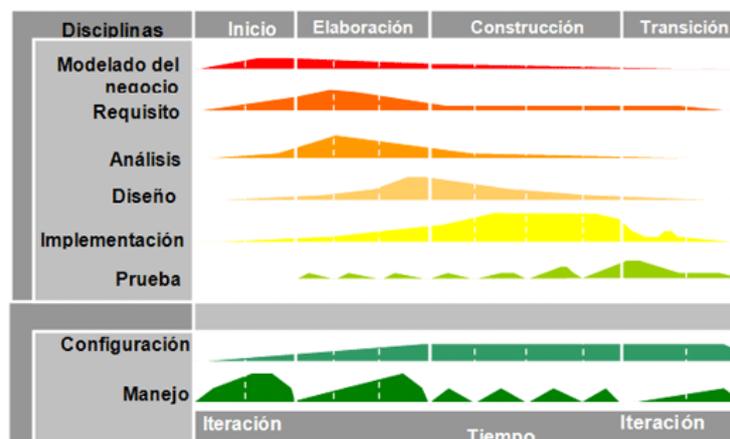


Figura 12: Fases del Proceso Unificado

5.2. Estimación de costes y esfuerzo

Una de las primeras cosas que hacer en el desarrollo del proyecto es la estimación de costes y esfuerzo, ya que esto te ofrece una idea de la duración de este. Basada en la métrica UCP, puntos de caso de uso, en la cual consideraremos actores, escenarios o casos de uso y factores tanto técnicos como de entorno. Para dicha estimación se hará uso de la herramienta EZEstimate.

Más información referente a esta fase puede encontrarse en el *Anexo I – Plan del proyecto software*.

En la *Figura 13* se puede observar la estimación de costes que estima la herramienta EZEstimate.

The screenshot displays the EZEstimate software interface with the following sections:

- Module:** A dropdown menu set to 'Actores', with 'Add Module' and 'Delete' buttons.
- Summary:**
 - Total Modules: 3
 - Excel Report: Generate Report button
 - Use cases: Simple (33), Average (3), Complex (0)
 - Actors: Simple (1), Average (0), Complex (2)
- Add Actor / Use case:** Fields for Actor / Use case Name, Select Type, Complexity, and an Add button.
- Tech / Env Factors:** Set Tech Factor and Set Env Factors buttons.
- Estimation Summary:**
 - UAW: 7
 - UUCW: 195
 - UUCP = UAW + UUCW: 202
 - TFactor: 26
 - EFactor: 14
 - TCF = 0.6 + (.01*TFactor): 0.86
 - EF = 1.4 + (.03*EFactor): 0.98
 - UCP = UUCP*TCF*EF: 170,2456
 - Total Effort@ 5 Hrs/UCP: 851,228
- Use case / Actor List:** A table with columns: Id, Module, Type, Name, complexity.

Id	Module	Type	Name	complexity
1	Actores	Actor	ACT-0001 Usu...	Complex
10	Gestión de usu...	Usecase	UC-0007 Cambi...	Simple
11	Gestión de usu...	Usecase	UC-0008 Cerrar...	Simple
12	Gestión de usu...	Usecase	UC-0009 Cambi...	Simple
13	Gestión de usu...	Usecase	UC-0010 Elimin...	Simple
14	Gestión de usu...	Usecase	UC-0011 Ayuda	Simple
15	Gestión de usu...	Usecase	UC-0012 Listar ...	Simple
16	Gestión de usu...	Usecase	UC-0013 Búsqu...	Simple
17	Gestión de usu...	Usecase	UC-0014 Ver p...	Simple
18	Gestión de rutas	Usecase	UC-0015 Añadir...	Average
19	Gestión de rutas	Usecase	UC-0016 Añadir...	Simple
2	Actores	Actor	ACT-0002 Usu...	Complex
20	Gestión de rutas	Usecase	UC-0017 Añadir...	Average
21	Gestión de rutas	Usecase	UC-0018 Elim...	Simple
22	Gestión de rutas	Usecase	UC-0019 Comp...	Simple
23	Gestión de rutas	Usecase	UC-0020 Valora...	Simple
24	Gestión de rutas	Usecase	UC-0021 Ver c...	Simple
25	Gestión de rutas	Usecase	UC-0022 Añadir	Simple

Figura 13: Estimación de esfuerzo en EZEstimate

Apreciando un valor de 5 horas por cada caso de uso, ya que la mayoría de ellos son sencillos, la estimación nos da como resultado un valor de 851 horas de trabajo para el desarrollo del proyecto.

5.3. Planificación temporal

Una vez realizada la estimación de costes y esfuerzo, toca llevar a cabo una planificación temporal para planear las tareas necesarias para el desarrollo del proyecto, así como de su duración y sus recursos. Para la realización de la planificación temporal se ha utilizado la herramienta Microsoft Project.

Más información referente a esta fase puede encontrarse en el *Anexo I – Plan del proyecto software*.

En la *Figura 14* se puede observar la planificación temporal del proyecto mediante la herramienta Microsoft Project.

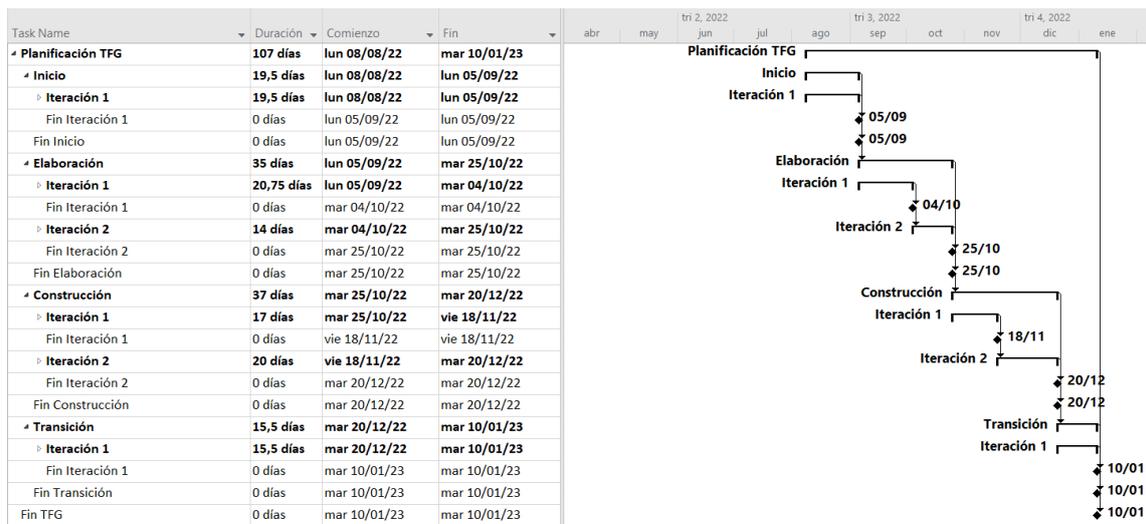


Figura 14: Planificación temporal del proyecto

Como dicta el Proceso Unificado, será un proceso iterativo e incremental que constará de las siguientes etapas en cada iteración:

- **Modelado de negocio:** Se realiza una búsqueda e investigación de aspectos relevantes al proyecto, como recursos o información, para dotar de suficientes conocimientos para la realización del proyecto.
- **Requisitos:** Se fijan los requisitos y objetivos del sistema, como también de revisarlos y modificarlos si lo requieren.
- **Análisis:** Se analizan y refinan los casos de uso con mayor profundidad.
- **Diseño:** Se diseñan y describen como actuaran los componentes del sistema.
- **Implementación:** Se implementa el sistema en términos de componentes, como la programación de código fuente.
- **Pruebas:** Se realizan las pruebas necesarias para comprobar que su funcionamiento es el esperado

En la *Figura 15* se puede observar la división de tareas con su respectivo diagrama de Gantt del proyecto.

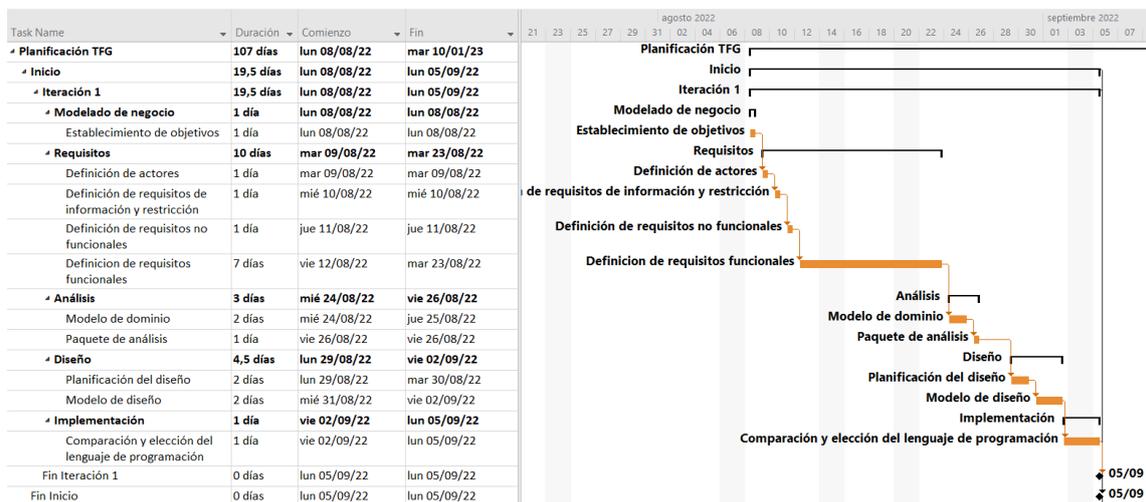


Figura 15: División de tareas y diagrama de Gantt

5.4. Especificación de requisitos

A continuación, se va a recoger la especificación de requisitos software del sistema a desarrollar mediante la metodología de Durán y Bernández para la elicitación de requisitos.

Más información referente a esta fase puede encontrarse en el *Anexo II – Especificación de requisitos del software*.

5.4.1. Participantes

Los participantes del proyecto son los siguientes:

- Calvo Prieto, Andrés.
- Villarrubia González, Gabriel.
- Sales Mendes, André Filipe.

Seguidamente, se muestra un ejemplo de especificación de los participantes del proyecto:

Participante	Calvo Prieto, Andrés
Organización	Universidad de Salamanca
Rol	Autor y desarrollador del proyecto
Comentarios	Ninguno

Tabla 1: Participante Calvo Prieto, Andrés

5.4.2. Objetivos del sistema

Los objetivos que se han definido y el sistema debe cumplir para satisfacer los requisitos son los siguientes:

- Gestión de usuarios.
- Gestión de ubicaciones.
- Gestión de rutas.
- Gestión de filtrado y búsqueda.
- Gestión de sistemas.

Un ejemplo de especificación de objetivos del proyecto será:

OBJ-0003	Gestión de rutas
Versión	1.0
Autores	Calvo Prieto, Andrés
Fuentes	Villarrubia González, Gabriel Sales Mendes, André Filipe
Descripción	El sistema deberá gestionar todo aquello relacionado con las rutas, como la creación y almacenamiento de las rutas que los usuarios hayan creado y mostrar dichas rutas en las pantallas correspondientes.
Subobjetivos	Ninguno

Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 2: OBJ-0003 Gestión de rutas

5.4.3. Requisitos de información

Los requisitos de información son aquellos que indican los datos que van a ser almacenados en el sistema, y son los siguientes:

- Datos de usuario.
- Ruta.
- Valoración.
- Comentario.
- Guardado.

Se va a mostrar un ejemplo de especificación de requisito de información:

IRQ-0001 Datos de usuario	
Versión	1.0
Autores	Calvo Prieto, Andrés
Fuentes	Villarrubia González, Gabriel Sales Mendes, André Filipe
Dependencias	[OBJ-0001] Gestión de usuarios
Descripción	El sistema deberá almacenar la información correspondiente a los datos de los usuarios registrados en el sistema. En concreto:
Datos específicos	ID Nombre Nombre de usuario Email Foto de perfil
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 3: IRQ-0001 Datos de usuario

5.4.4. Requisitos de restricción

Los requisitos de restricción son aquellos requisitos que están sujetos a unas reglas de negocio o unas restricciones, son los siguientes:

- Correo registrado.
- Nombre de usuario idéntico.

5.4.5. Definición de actores

Los actores que participaran en el sistema son los siguientes:

- Usuario anónimo.
- Usuario identificado.
- Sistema.

Un ejemplo de especificación de actor es el siguiente:

ACT-0002	Usuario identificado
Versión	1.0
Autores	Calvo Prieto, Andrés
Fuentes	Villarrubia González, Gabriel Sales Mendes, André Filipe
Descripción	Este actor representa a un usuario que accede al sistema y se ha identificado correctamente.
Comentarios	Ninguno

Tabla 4: ACT-0002 Usuario identificado

5.4.6. Requisitos funcionales

Los requisitos funcionales son aquellos que definen una actividad que el sistema debe realizar.

Se ha realizado un diagrama de paquetes para organizar los requisitos del sistema según su funcionalidad.

En la *Figura 16* se puede observar el diagrama de paquetes.

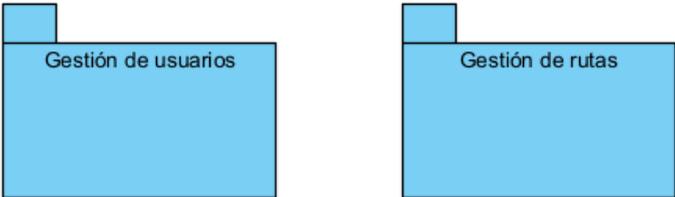


Figura 16: Diagrama de paquetes

Se definen los casos de uso y se realizan los diagramas de casos de uso por los paquetes anteriormente definidos, más tarde se detallarán los casos de uso.

En la *Figura 17* se puede observar el diagrama de gestión de usuarios del sistema.

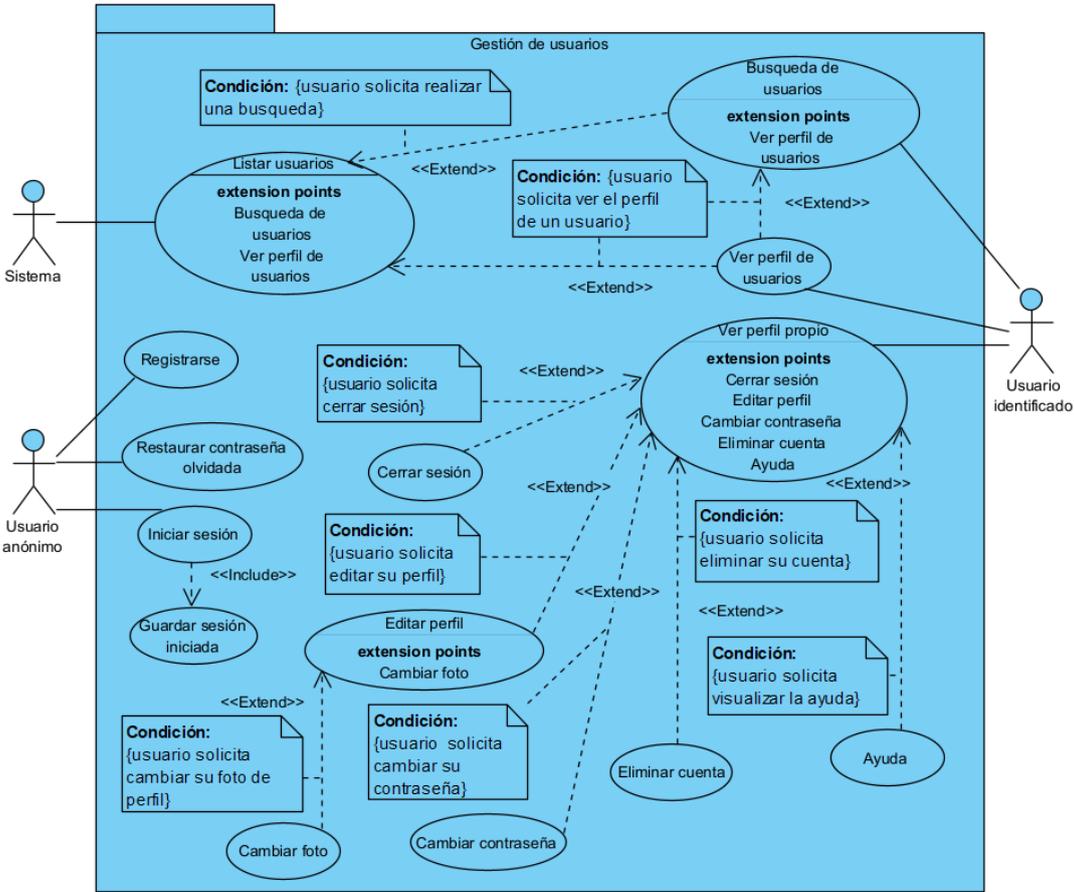


Figura 17: Diagrama de gestión de usuarios

Se va a mostrar un ejemplo de especificación de requisito funcional:

UC-0015		Añadir ruta	
Versión	1.0		
Autores	Calvo Prieto, Andrés		
Fuentes	Villarrubia González, Gabriel Sales Mendes, André Filipe		
Dependencias	[OBJ-0002] Gestión de ubicaciones [OBJ-0003] Gestión de rutas [IRQ-0002] Ruta		
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario quiera crear una ruta nueva.		
Precondición	Debe haber un actor <i>Usuario identificado (ACT-0002)</i> con una sesión activa en el dispositivo.		
Secuencia normal	Paso	Acción	
	1	El actor <i>Usuario identificado (ACT-0002)</i> solicita crear una ruta nueva.	
	2	El sistema solicita al usuario un nombre de la ruta, descripción y tipo.	
	3	El actor <i>Usuario identificado (ACT-0002)</i> introduce los datos.	
	4	Si el usuario solicita añadir foto a la ruta, se realiza el caso de uso <i>Añadir foto a la ruta (UC-0016)</i> .	
	5	Se realiza el caso de uso <i>Añadir ubicaciones (UC-0017)</i> .	
	6	El sistema almacena la ruta correctamente.	
Postcondición	-		
Excepciones	Paso	Acción	
	6	Si los datos introducidos no son válidos o están incompletos, el sistema muestra al usuario un aviso con dicho error y se regresa al paso 3 hasta que se corrijan dichos errores.	
Importancia	Vital		
Urgencia	Inmediatamente		
Estado	Validado		
Estabilidad	Alta		
Comentarios	Ninguno		

Tabla 5: UC-0015 Añadir ruta

5.4.7. Requisitos no funcionales

Los requisitos no funcionales son aquellos que describen servicios que prestará el sistema, asimismo de las restricciones en el diseño e implementación. Son los siguientes:

- Concurrencia.
- Usabilidad.
- Escalabilidad.
- Eficiencia.
- Seguridad.

A continuación, se muestra un ejemplo de especificación de requisito no funcional:

NFR-0003 Escalabilidad	
Versión	1.0
Autores	Calvo Prieto, Andrés
Fuentes	Villarrubia González, Gabriel Sales Mendes, André Filipe
Dependencias	Ninguna
Descripción	El sistema deberá ser capaz de adaptarse correctamente sin comprometer el rendimiento y calidad a medida que aumenten los usuarios que hagan uso del sistema.
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta
Comentarios	Ninguno

Tabla 6: NFR-0003 Escalabilidad

5.5. Análisis de requisitos

En este apartado se van a analizar los requisitos recogidos en el punto anterior de forma más exhaustiva.

Más información referente a esta fase puede encontrarse en el *Anexo III – Especificación de diseño*.

5.5.1. Modelo de dominio

El modelo de dominio representa las clases conceptuales significativas para el dominio del problema. Se describen entidades, atributos y relaciones entre estos.

En la *Figura 18* se puede observar el diagrama de clases para representar el modelo de dominio.

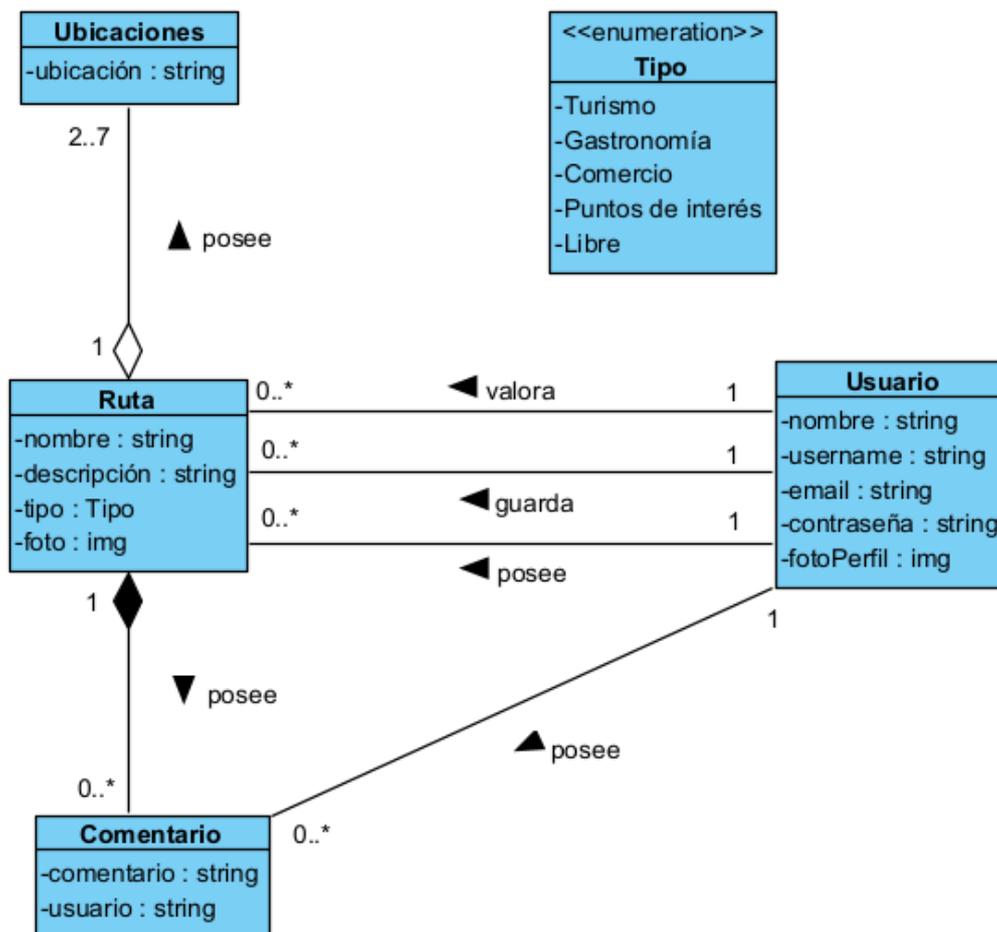


Figura 18: Diagrama de clases modelo de dominio

5.5.2. Realización de casos de uso de análisis

A continuación, se va a realizar un análisis de los casos de uso mediante diagramas de secuencias para observar cómo conectan entre sí los distintos elementos y el intercambio de mensajes entre estos.

En la *Figura 19* se puede observar un diagrama de secuencia de análisis.

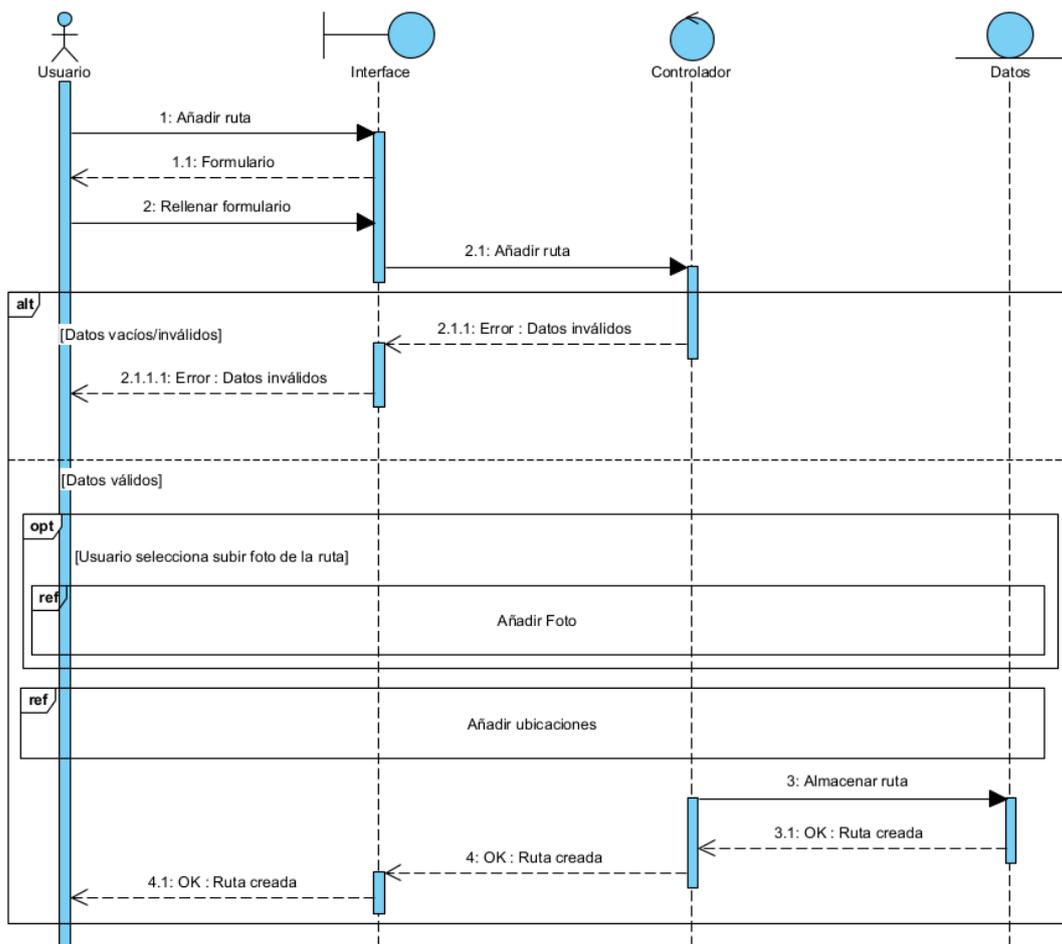


Figura 19: Diagrama de secuencia de análisis Añadir ruta

5.5.3. Clases de análisis

A continuación, se van a tratar los diagramas de comunicación, los cuales se encargan de mostrar la distribución de las clases de análisis y las comunicaciones entre estos.

5.5.4. Vista arquitectura del modelo de análisis

La vista arquitectura del modelo análisis sitúa las clases de análisis dentro de sus capas correspondientes del patrón arquitectónico que usaremos en la fase de diseño.

En la *Figura 20* se puede observar la vista arquitectura del modelo de análisis.

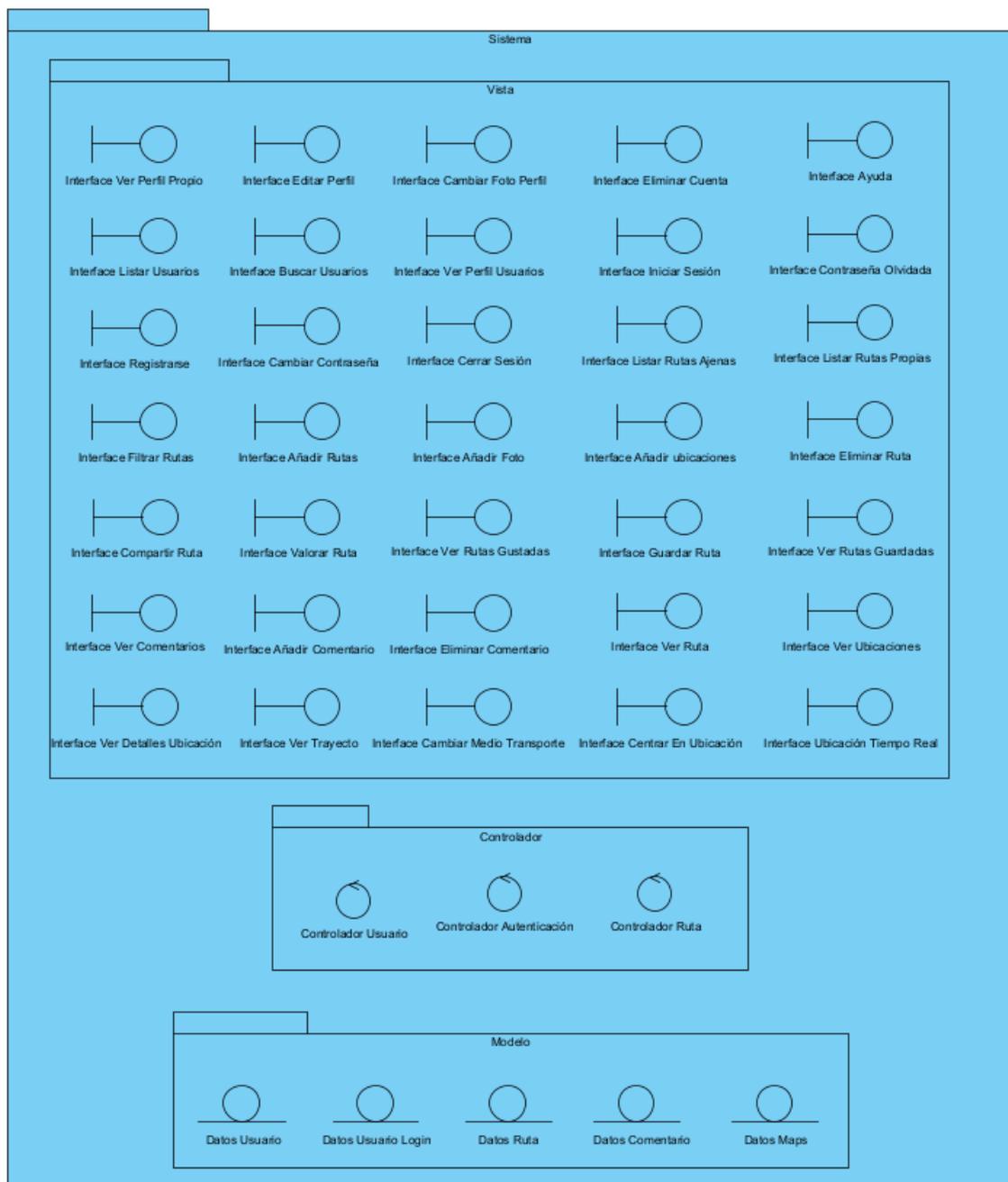


Figura 20: Arquitectura del modelo de análisis

5.6. Diseño del sistema

El diseño del sistema se centra en el dominio de la solución, esto es, se va a describir con mayor detalle la estructura del sistema y como será implementado acercándose más al producto final.

5.6.1. Patrón arquitectónico MVC

Para llevar a cabo una separación entre los datos del sistema, la interfaz de usuario y la lógica de control entre estas, se utilizará un patrón arquitectónico MVC (Modelo-Vista-Controlador).

En el siguiente esquema se puede visualizar la relación entre ellos y como se comunican:

- **Modelo:** Representa los datos del sistema y su lógica de negocio.
- **Vista:** Representa de forma visual los datos del modelo gracias a interfaces y elementos visuales.
- **Controlador:** Representa la dirección de órdenes del usuario y se encarga de realizar los controles apropiados a los datos para cumplir con dichas órdenes. Actúa como el intermediario del modelo y la vista, es decir, entre los datos del sistema y la representación de estos.

En la *Figura 21* se puede observar el esquema del patrón arquitectónico MVC.

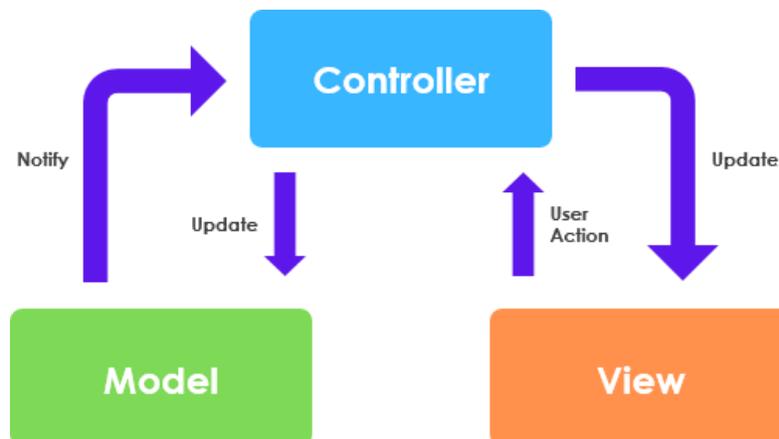


Figura 21: Patrón arquitectónico MVC

5.6.2. Subsistemas de diseño

Dados los grupos del patrón MVC, se ha decidido dividir el sistema en paquetes y estos a su vez en subpaquetes para una correcta y clara organización del sistema.

En la *Figura 22* se puede observar los paquetes de diseño y las relaciones entre estos.

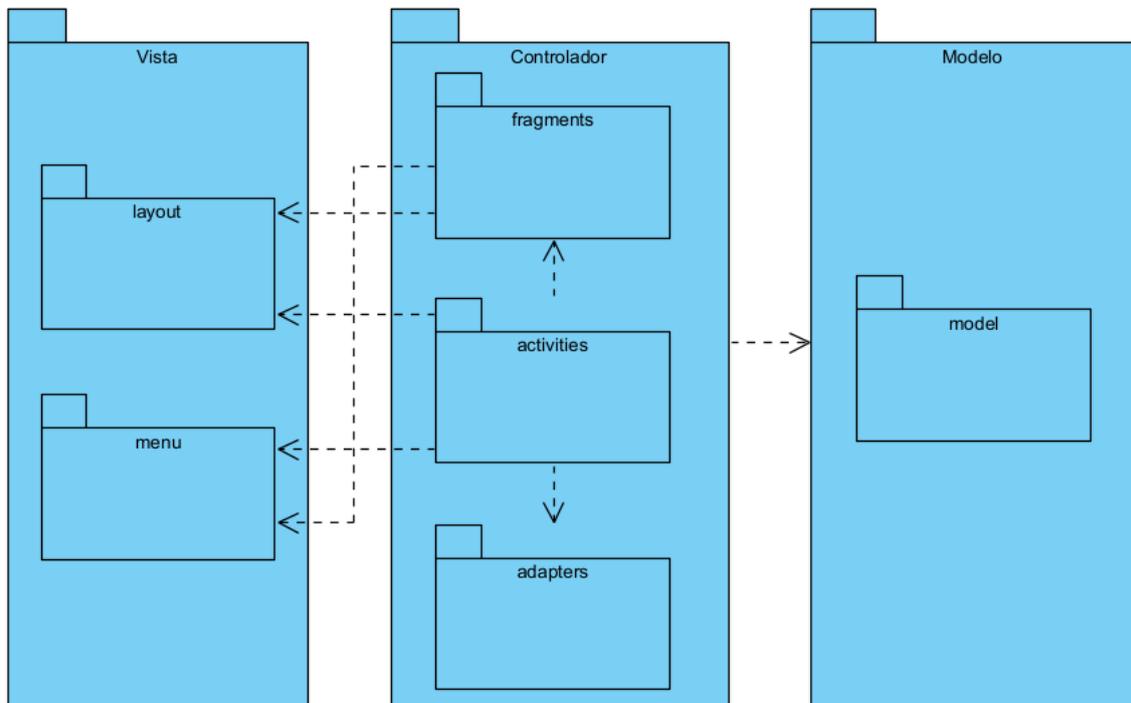


Figura 22: Subsistemas de diseño

5.6.3. Clases de diseño

Seguidamente se detalla el contenido de cada paquete definido en los anteriores puntos, así como las clases y métodos de estas.

En la *Figura 23* se puede observar una clase de diseño.

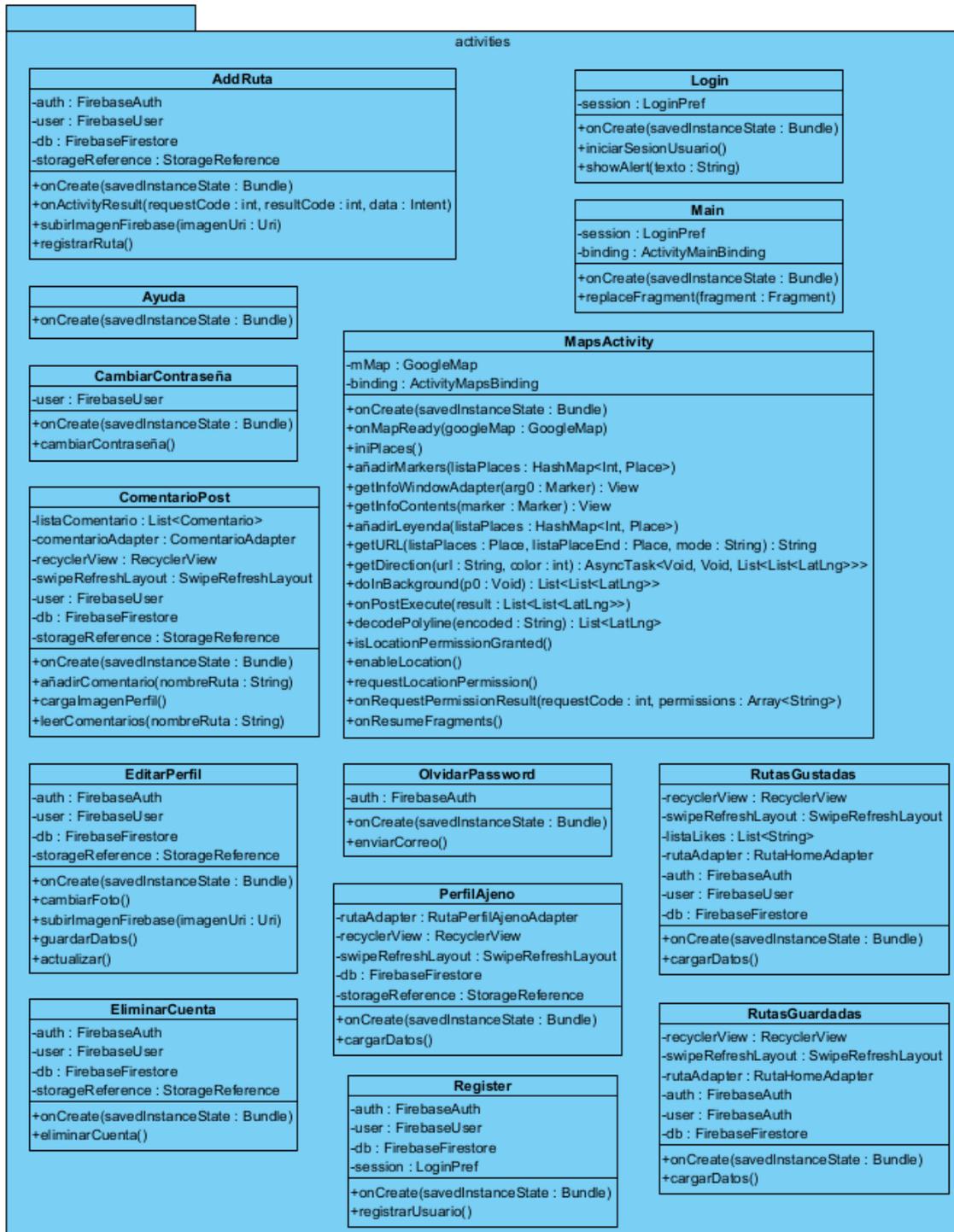


Figura 23: Clase de diseño del paquete Activities

5.6.4. Realización de casos de uso de diseño

Actualmente, se va a realizar el diseño de los casos de uso, los cuales previamente fueron sujeto del análisis. También se emplearán diagramas de secuencia, los cuales mostrarán como los distintos elementos previamente definidos van a conectarse entre sí y comunicarse por medio de mensajes.

En la *Figura 24* se puede observar un diagrama de secuencia de diseño.

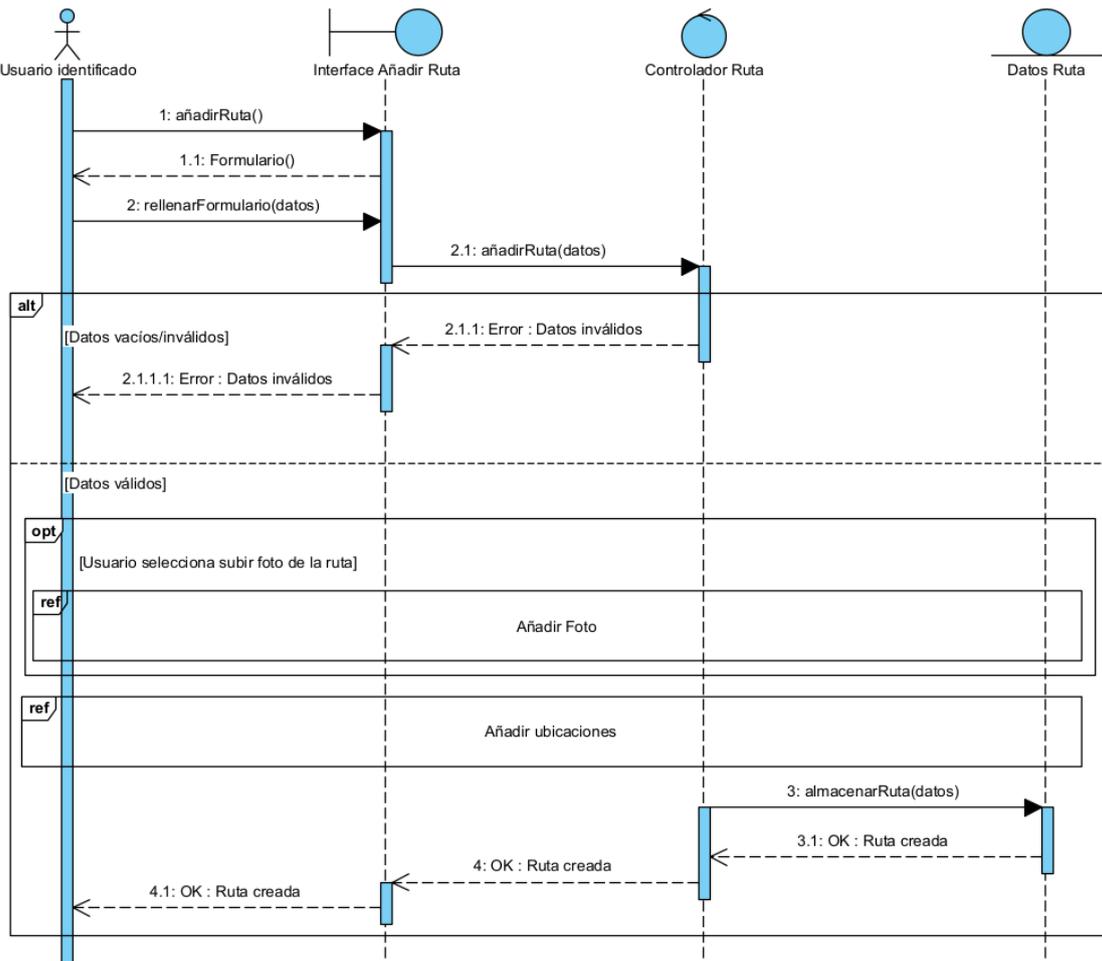


Figura 24: Diagrama de secuencia de diseño Añadir ruta

5.6.5. Diseño de la base de datos

En cuanto a la base de datos, se emplearán dos bases de datos, ambas alojadas en la plataforma Firebase de Google. Cloud Firestore Database constará de todos los datos simples, mientras que Storage almacenará las imágenes.

En la *Figura 25* se puede observar el diseño de las bases de datos que utiliza el proyecto.

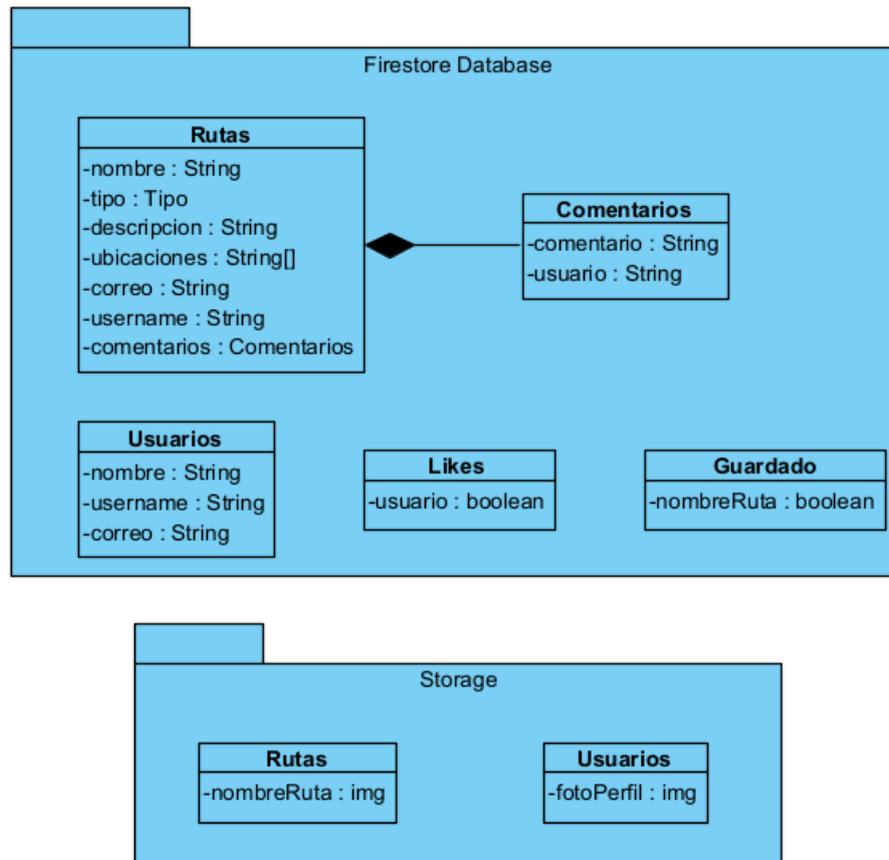


Figura 25: Diseño de la base de datos

5.6.6. Modelo de despliegue

El modelo de despliegue muestra de forma gráfica los distintos artefactos software del sistema en forma de nodos y como estos están organizados.

En la *Figura 26* se puede observar el modelo de despliegue.

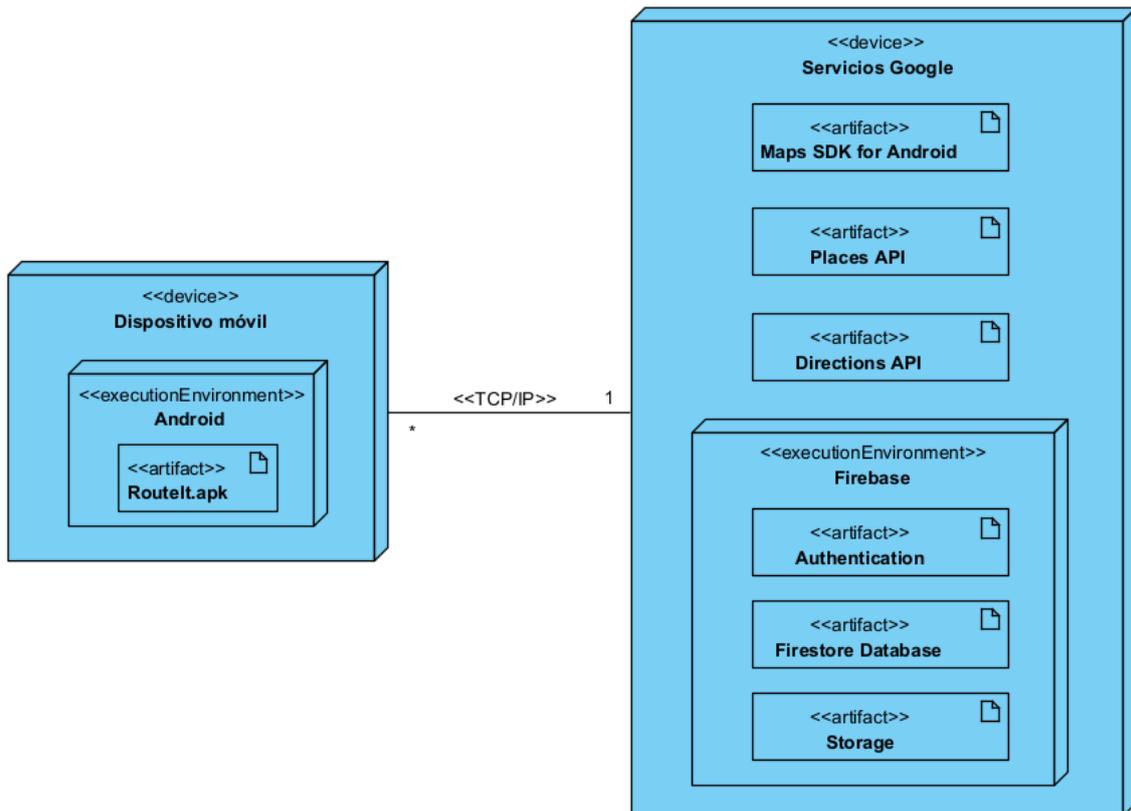


Figura 26: Diagrama de despliegue

5.7. Implementación

Durante la fase de implementación se ha realizado la programación del sistema tomando como referencia los resultados obtenidos de la fase anteriormente descrita del diseño.

Dicha programación se ha realizado para el sistema operativo Android haciendo uso del entorno de desarrollo integrado oficial de Google, Android Studio. Para el lenguaje de programación en este IDE se ha optado por Kotlin, el cual es muy recomendado utilizar por parte de Google, ya que ofrece varias mejoras respecto a Java y muchas aplicaciones ya están haciendo el cambio hacia Kotlin.

Respecto a las bases se usará la plataforma Firebase, como esta pertenece a Google, hay mucha conexión y facilidades para usar, también ofrece un servicio de autenticación muy útil y sencillo de implementar. Google Cloud Platform ofrece el SDK y varias API con las que reduce la complejidad de usar y manejar mapas en la aplicación.

Se comenzó implementando el sistema de autenticación para poder acceder a la aplicación y salir de ella, más tarde se creó el perfil de los usuarios y el poder editarlo a gusto del usuario, con esto todas las acciones relacionadas como cambiar contraseña, borrar cuenta, etc.

Una vez que esto tenía una base y funcionaba correctamente, se pasó a una fase inicial de la creación de rutas, pero esto requería del buscador de ubicaciones de Google y con eso el utilizar una de las API de Places, que permitía el autocompletado de lugares. Una vez creada la posibilidad de crear rutas, había que mostrarlas, pues se diseñó la pantalla principal de mostrado de rutas y la de rutas personales.

Con las rutas ya creadas, y con las IDs de las ubicaciones que estas la forman, se desarrolló el mapa donde ocuparían lugar, creando peticiones a los servicios de Google y con esto crear los marcadores y los detalles de cada ubicación.

Con una versión inicial de la funcionalidad principal funcionando, se pasó a añadir el sistema de valoración y comentarios de las rutas. Seguidamente, el sistema de filtrado de las rutas en la pantalla principal y rutas personales para una sencilla búsqueda por tipo de rutas.

Posteriormente, se realiza el sistema de búsqueda de usuarios, filtro de estos y visualización de perfiles ajenos. Una vez finalizado, se terminó añadiendo unas funciones más a las rutas como el poder compartir rutas propias y el guardar rutas, así también como las pantallas correspondientes para visualizar las rutas guardadas y gustadas. Mencionando también la posibilidad de poder recuperar la cuenta si el usuario no recuerda la contraseña y el guardado de inicio de sesión para que el usuario no tenga que estar todo el rato poniendo su correo y contraseña cada vez que entra a la aplicación.

Con toda funcionalidad externa a los mapas de las rutas terminada, se trabajó sobre el mapa y las ubicaciones, como el añadido del cuadro de leyenda, la trayectoria, la ubicación en tiempo real para poder seguir la trayectoria sin uso de otros mapas, los botones para cambiar las vistas y medio de transporte, etc.

Finalmente, se mejoró el aspecto visual de la aplicación como de muchos de sus elementos, añadiendo nuevas funcionalidades, como por ejemplo el visualizado de contraseñas en los cuadros de texto antes de realizar una acción, y el soporte tanto en modo oscuro de la aplicación como de la vista con el dispositivo en horizontal.

La documentación técnica de programación se ha generado en formato HTML para que todo aquel que desee informarse pueda hacerlo de una forma más organizada y sencilla de entender. Más información referente a esta fase puede encontrarse en el *Anexo IV – Documentación técnica de programación*.

5.8. Pruebas

La fase de pruebas es muy importante en el desarrollo del proyecto, pues permite comprobar que toda funcionalidad implementada actúa correctamente y cumple con los objetivos.

Durante el desarrollo se realizaron pruebas unitarias de cada funcionalidad nueva que se iba incorporando al sistema para asegurarse de que su funcionamiento era el correcto.

También se realizaron pruebas globales cuando el sistema ya poseía una cantidad notable de funcionalidades para comprobar que el sistema se comportaba de forma correcta y que los componentes estaban correctamente integrados.

Cabe mencionar que dichas pruebas fueron realizadas en varios dispositivos físicos distintos, comprobando también que el sistema funciona en una gama variada de dispositivos con distinta versión del sistema operativo Android.

5.9. Funcionalidad del sistema

A continuación, se va a explicar de forma resumida los aspectos más relevantes de la funcionalidad del sistema. Para una concisa organización y comprensión del funcionamiento, la sección se dividirá en varios apartados claramente definidos.

Más información referente a esta fase puede encontrarse en el *Anexo V – Manual de usuario*, donde pueden consultarse todas las acciones que se pueden realizar en el sistema.

5.9.1. Acceso a la aplicación

En la pantalla de inicio encontramos la posibilidad de iniciar sesión, registrarse y recuperar la contraseña si el usuario la ha olvidado.

En la *Figura 27* se puede observar la pantalla de inicio de sesión a la aplicación.

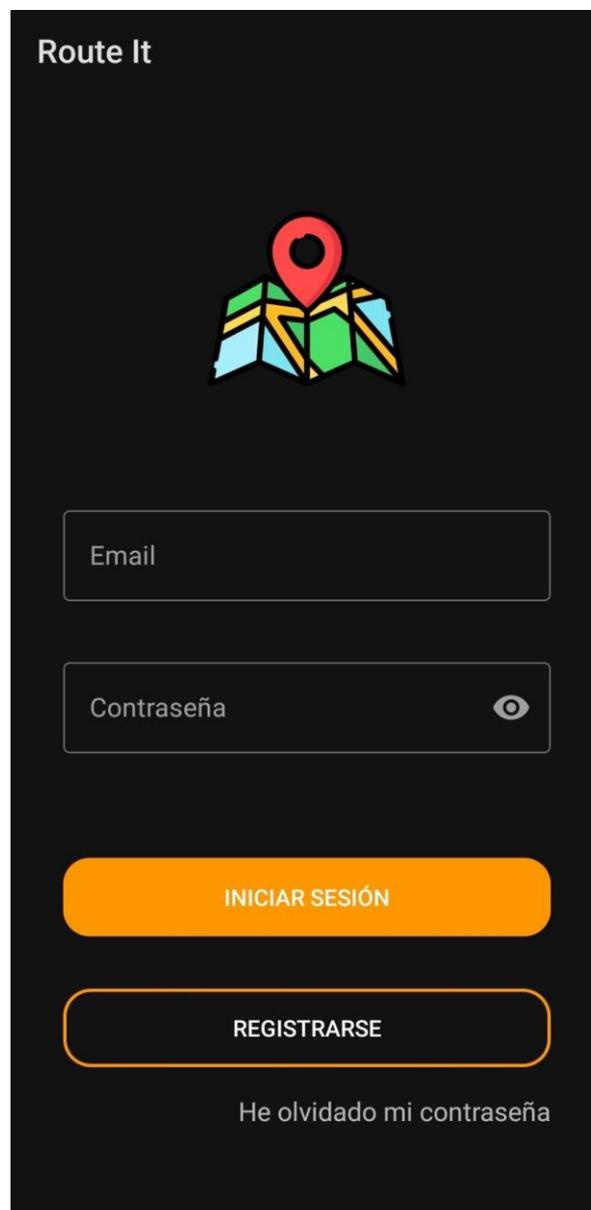


Figura 27: Pantalla de inicio de sesión

El usuario podrá crear una cuenta mientras el username o el email ya esté vinculado con un usuario existente del sistema.

En la *Figura 28* se puede observar la pantalla de registro de la aplicación.

The image shows a mobile application registration screen titled "Route It". The background is black. At the top, the title "Route It" is displayed in white. Below the title, there are five input fields, each with a white border and a light gray placeholder text. The fields are labeled "Nombre", "Username", "Email", "Contraseña", and "Repite Contraseña". The "Contraseña" and "Repite Contraseña" fields have a small white eye icon to the right of the input area, indicating a toggle for password visibility. At the bottom of the screen, there is a prominent orange button with rounded corners and the text "REGISTRARSE" in white, centered on the button.

Figura 28: Pantalla registro

Para recuperar la contraseña solo habrá que introducir la dirección de correo y se enviará un link con el que poder asignar una nueva contraseña para la cuenta del usuario.

En la *Figura 29* se puede observar la pantalla de restablecer la contraseña de la aplicación.

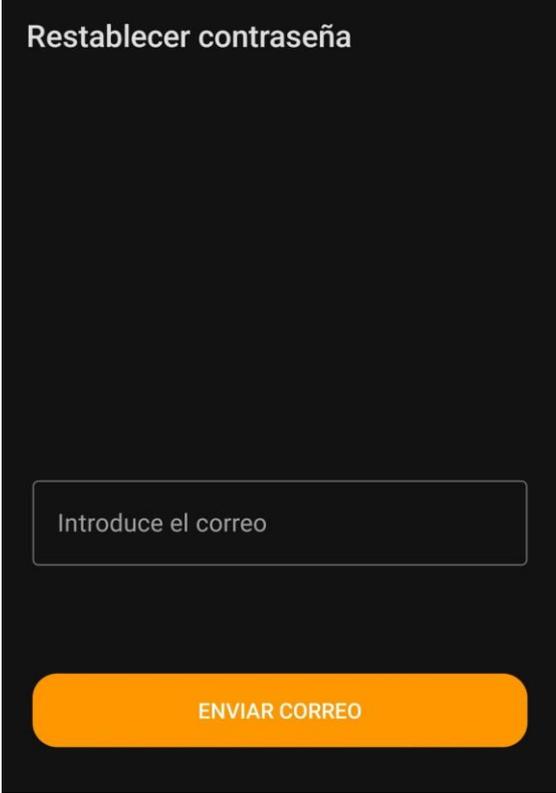
La imagen muestra una pantalla de restablecimiento de contraseña con un fondo negro. En la parte superior, el título "Restablecer contraseña" está escrito en blanco. Debajo, hay un campo de entrada rectangular con un borde gris claro que contiene el texto "Introduce el correo". En la parte inferior de la pantalla, hay un botón rectangular de color naranja con el texto "ENVIAR CORREO" en blanco.

Figura 29: Pantalla restablecer contraseña

En la *Figura 30* se puede observar la pantalla para introducir la nueva contraseña para la cuenta del usuario.

Cambiar la contraseña

de **andrescalvo@usal.es**

Nueva contraseña



GUARDAR

Figura 30: Formulario cambiar contraseña

5.9.2. Perfil de usuario y configuración

Una vez se haya iniciado sesión, se mostrará la pantalla principal de la aplicación donde se visualizan todas las rutas ajenas de los usuarios, para dirigirnos a la sección del perfil personal hay que pulsar sobre el botón de perfil del menú inferior.

Una vez en el perfil, el usuario podrá editar su perfil, visualizar las rutas guardadas y también las rutas gustadas que ya explicaremos más adelante.

En la *Figura 31* se puede observar la pantalla del perfil personal del usuario.

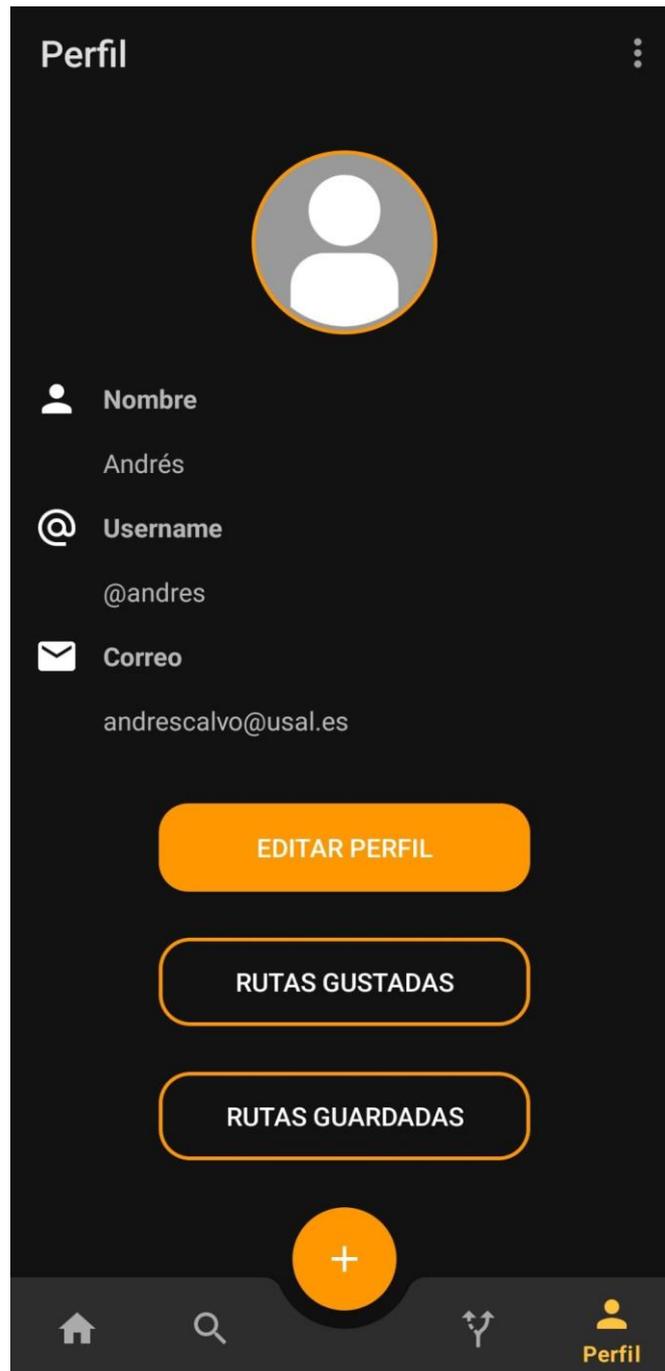


Figura 31: Pantalla Perfil

En el editor de perfil se podrá cambiar la foto y los datos del usuario, asegurando que no se puede escoger un username o correo que esté ya vinculado a otro usuario del sistema.

En la *Figura 32* se puede observar la pantalla de edición del perfil del usuario.

Editar perfil

CAMBIAR FOTO

Nombre
Andrés

Username
andres

Correo
andrescalvo@usal.es

Confirma la contraseña

GUARDAR CAMBIOS

Figura 32: Pantalla editar perfil

También se podrá cerrar sesión, cambiar contraseña, eliminar cuenta e ir a la pantalla de ayuda mediante el menú del perfil del usuario.

En la *Figura 33* se puede observar el menú de opciones de la aplicación.

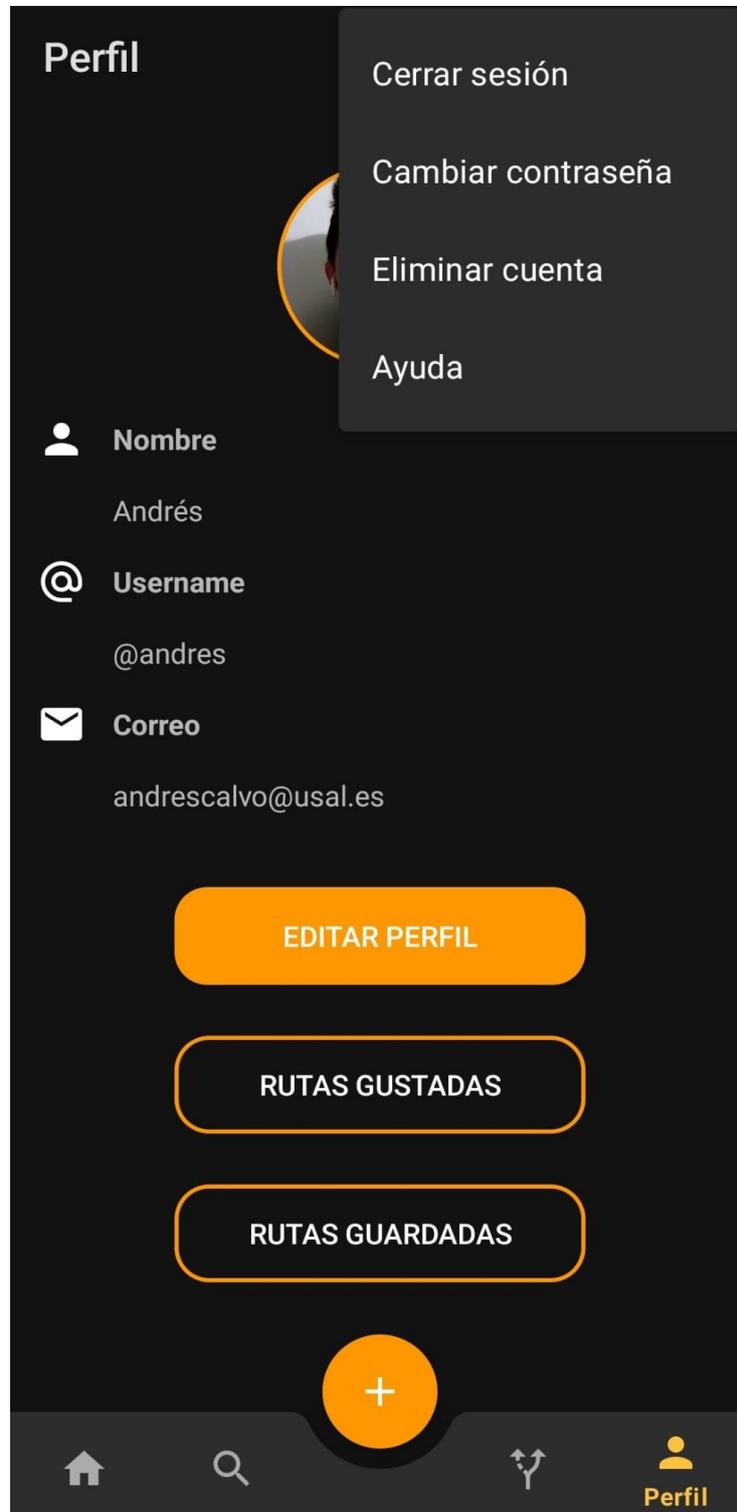


Figura 33: Menú del perfil de usuario

En la *Figura 34* se puede observar la pantalla de cambio de contraseña para la cuenta del usuario.

Cambiar contraseña

Contraseña actual 

Contraseña nueva 

Repita la nueva contraseña 

CAMBIAR CONTRASEÑA

Figura 34: Pantalla cambiar contraseña

5.9.3. Búsqueda e interacción de usuarios

Para ir a la búsqueda de usuarios, hay que dirigirse a la pantalla de búsqueda desde el menú inferior. Aquí se muestran todos los usuarios del sistema para que el usuario pueda visitar su perfil y visualizar sus rutas.

En la *Figura 35* se puede observar la pantalla de búsqueda de usuarios de la aplicación.

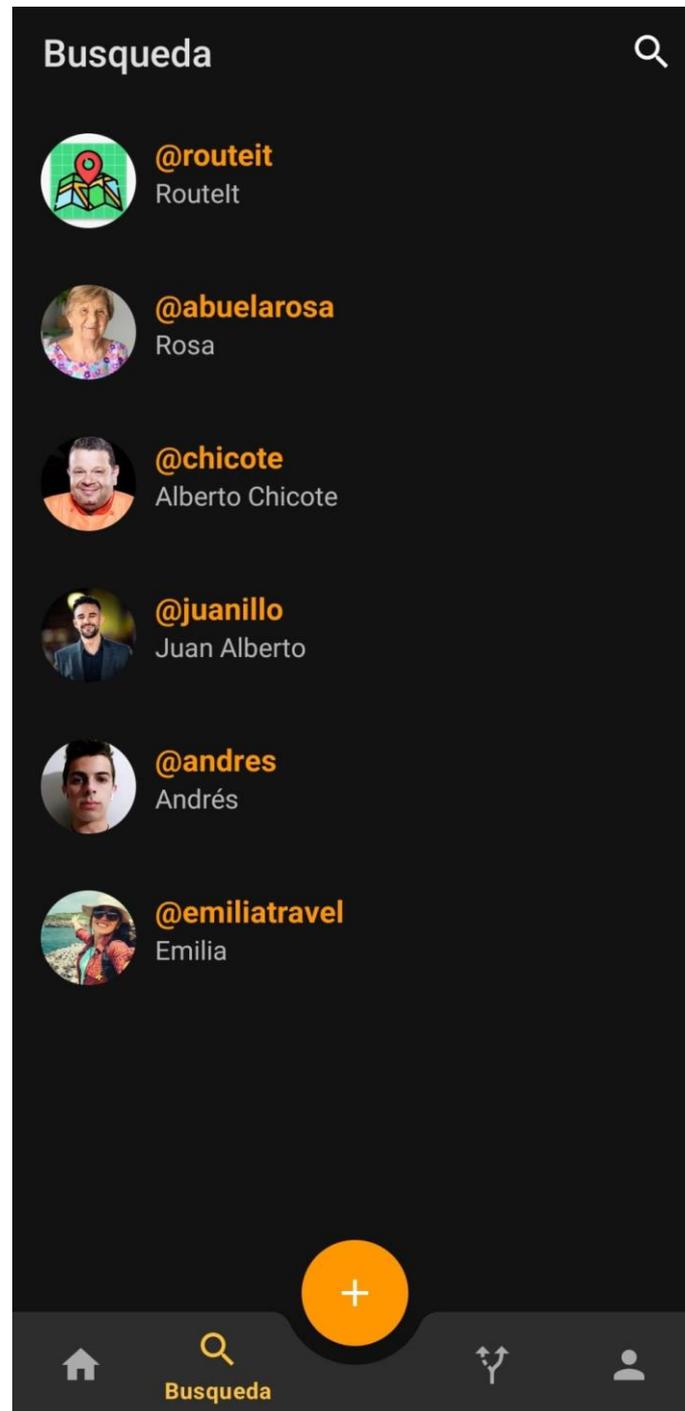


Figura 35: Pantalla Búsqueda

Se pueden buscar usuarios con la lupa situada en la esquina superior derecha e introduciendo el nombre o username del usuario que se desee buscar. Se mostrará un listado filtrado siguiendo la condición del nombre que ha impuesto el usuario.

En la *Figura 36* se puede observar la lista de usuarios filtrados.

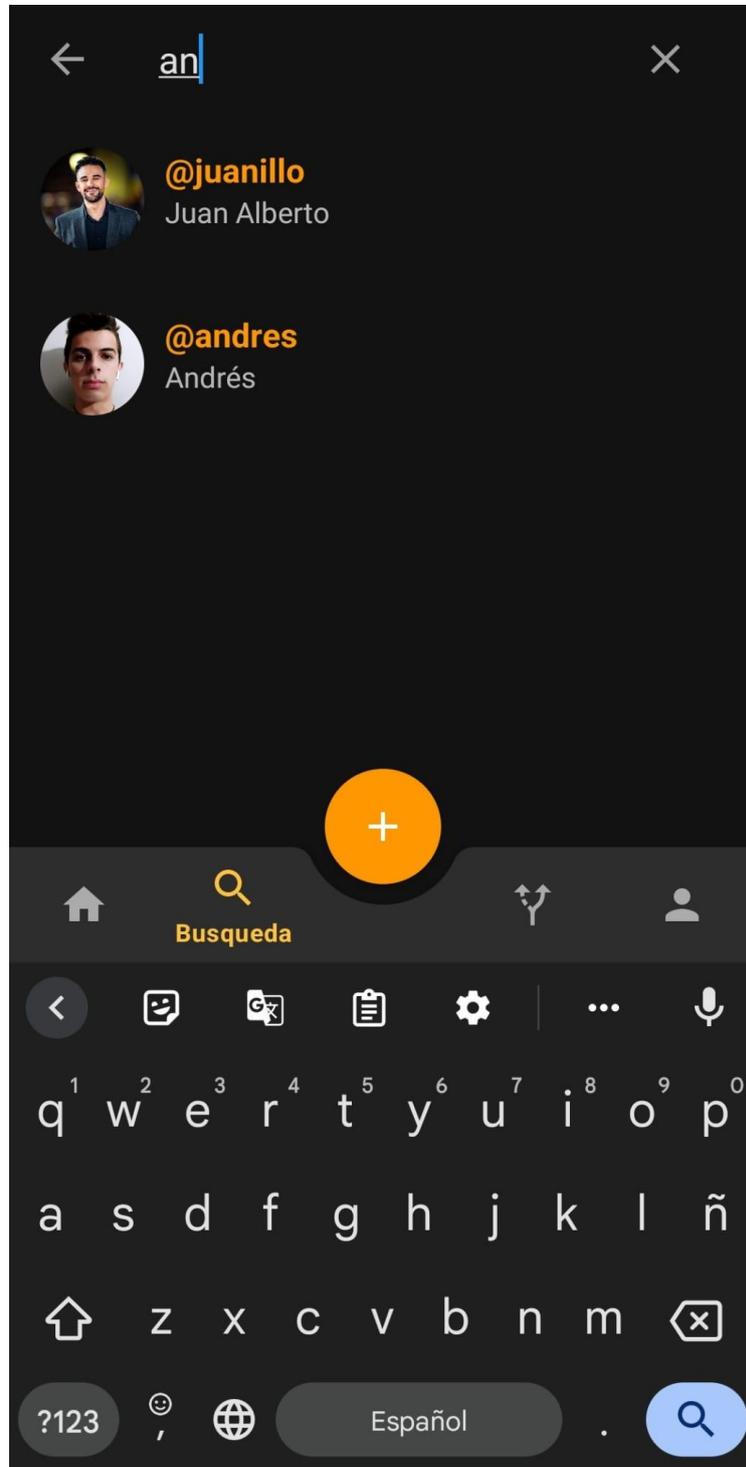


Figura 36: Búsqueda de usuarios

Se podrá visitar los perfiles de los usuarios, desde cualquier pantalla que los relacione, y con ello se muestran los datos de estos, juntos las rutas que han creado.

En la *Figura 37* se puede observar la pantalla del perfil público de un usuario.

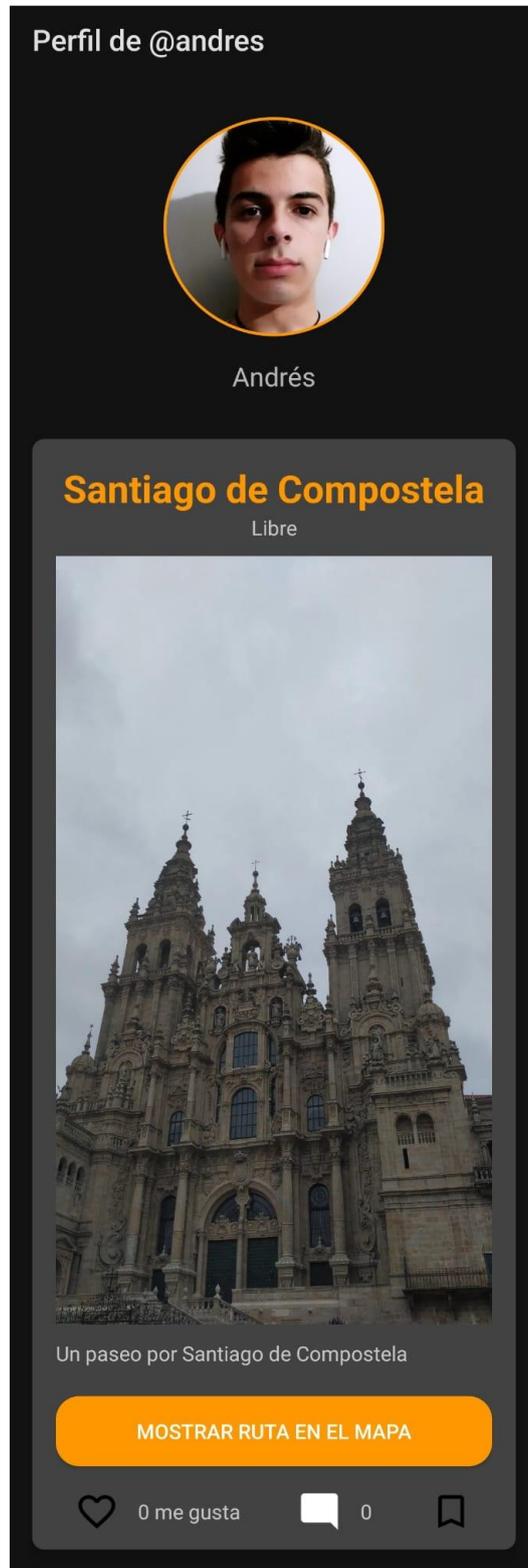
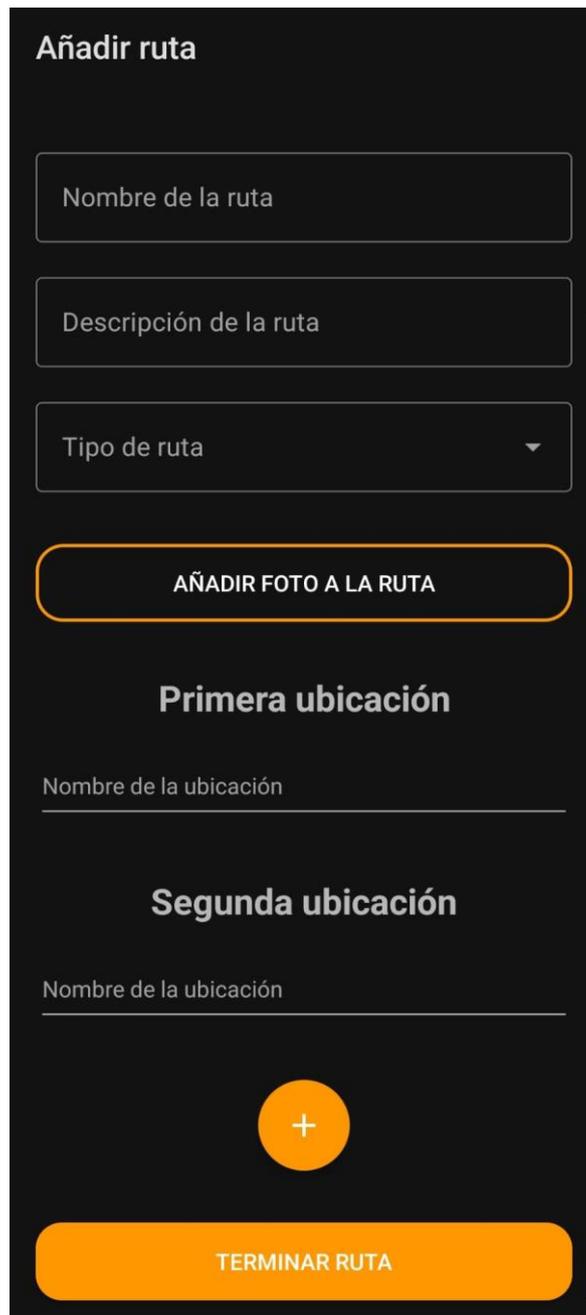


Figura 37: Perfil público de un usuario

5.9.4. Creación e interacción de rutas

En la creación de las rutas se solicita al usuario añadir un nombre representativo de la ruta, el tipo de ruta que más se asemeje a su idea y dos ubicaciones que la formen, después puede añadir de forma opcional una descripción y una foto a la ruta, como también más ubicaciones.

En la *Figura 38* se puede observar la pantalla de creación de una ruta.



La imagen muestra una interfaz de usuario para la creación de una ruta. El título es "Añadir ruta". Hay tres campos de entrada: "Nombre de la ruta", "Descripción de la ruta" y "Tipo de ruta" (con un menú desplegable). Debajo de estos campos hay un botón naranja con el texto "AÑADIR FOTO A LA RUTA". Luego, se ven dos secciones de ubicación: "Primera ubicación" y "Segunda ubicación", cada una con un campo de entrada "Nombre de la ubicación". Al final, hay un botón circular naranja con un signo "+" y un botón naranja grande con el texto "TERMINAR RUTA".

Figura 38: Creación de ruta

El sistema lista todas las rutas ajenas al usuario en la pantalla principal Home, donde el usuario podrá interactuar tanto con ellas como visitando el perfil de los usuarios.

En la *Figura 39* se puede observar la pantalla principal de la aplicación.

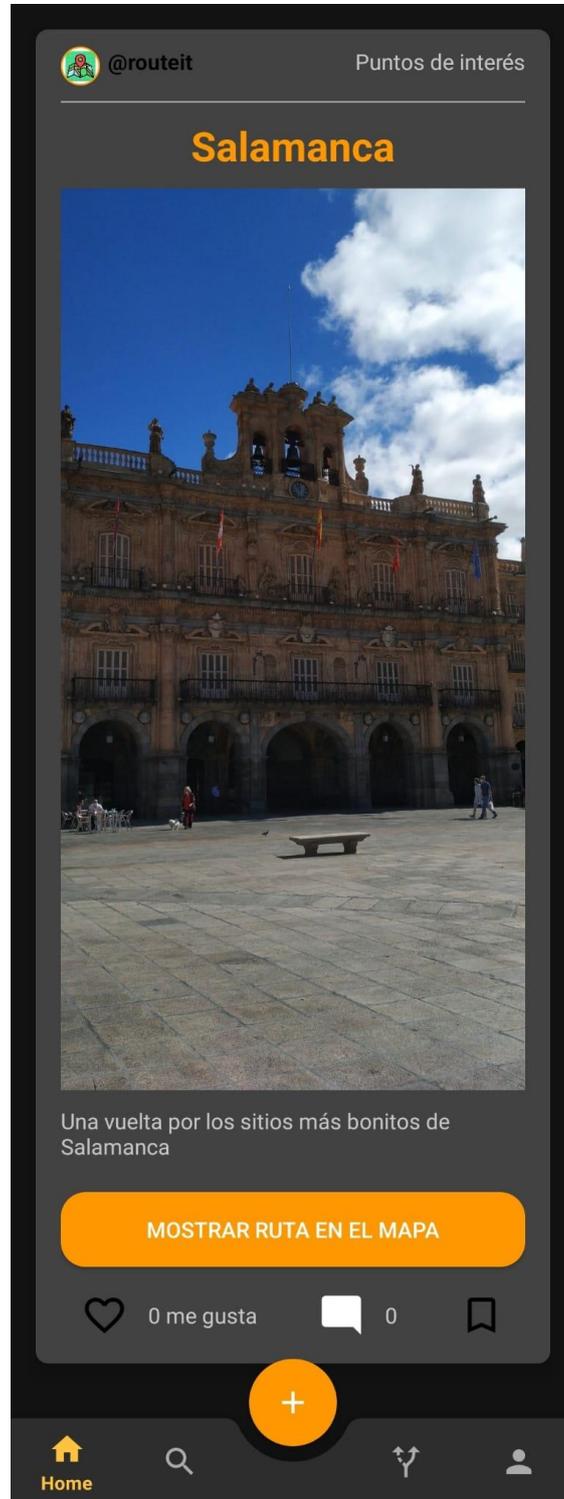


Figura 39: Pantalla principal Home

Se podrán filtrar las rutas por el tipo que son con el filtrado de la esquina superior derecha, también se podrán valorar, guardar y acceder al apartado de comentarios que posee dicha ruta donde se pueden añadir comentarios y eliminar comentarios propios.

En la *Figura 40* se puede observar la pantalla de comentarios de una ruta.

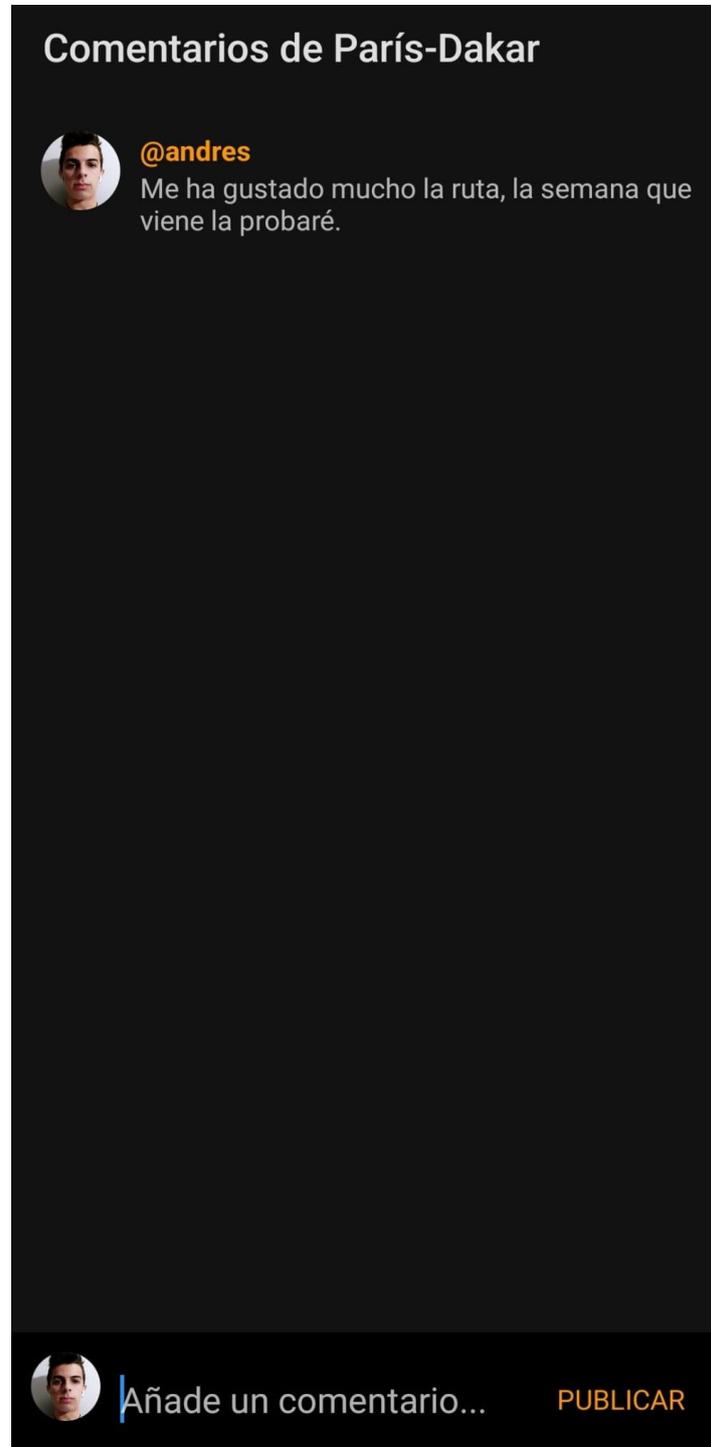


Figura 40: Comentarios de la ruta

El usuario podrá consultar las rutas personales que haya creado en la pantalla de rutas propias, las cuales, aparte de poder realizar sobre ellas lo mismo que en las demás rutas ajenas, tendrá la posibilidad de eliminarlas y compartirlas.

En la *Figura 41* se puede observar la pantalla de rutas personales del usuario.

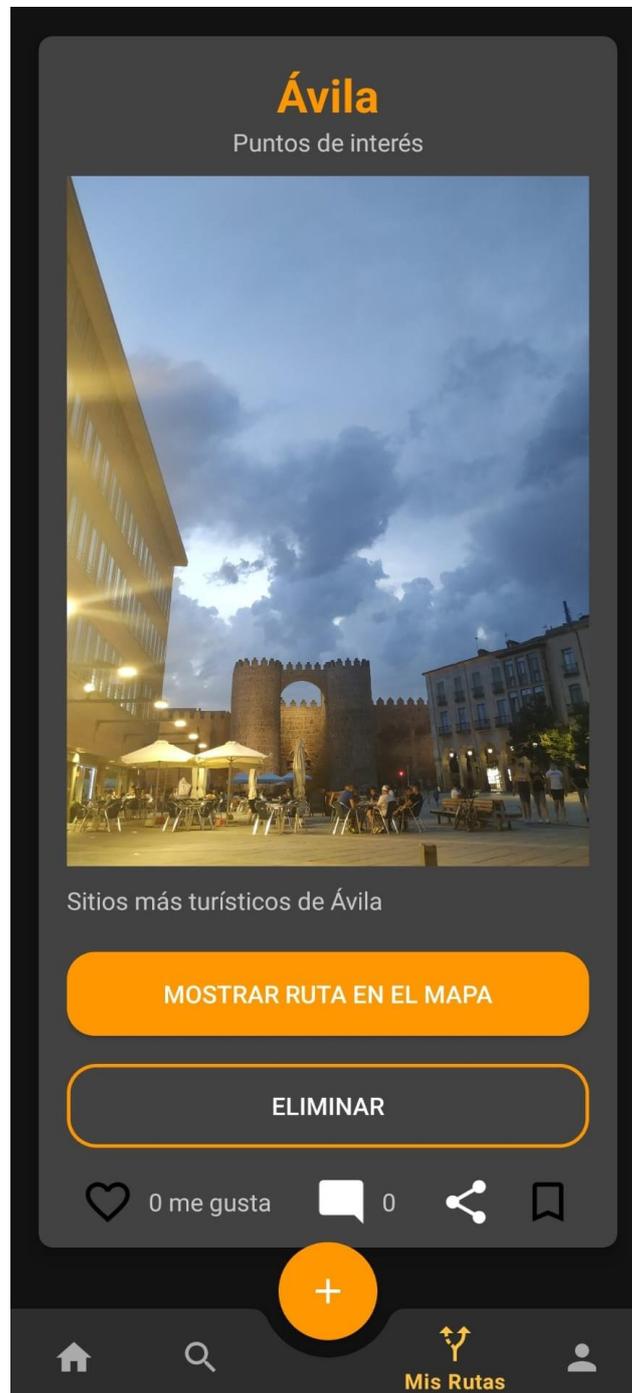


Figura 41: Pantalla rutas personales

En la *Figura 42* se pueden observar los datos que son compartidos de una ruta.

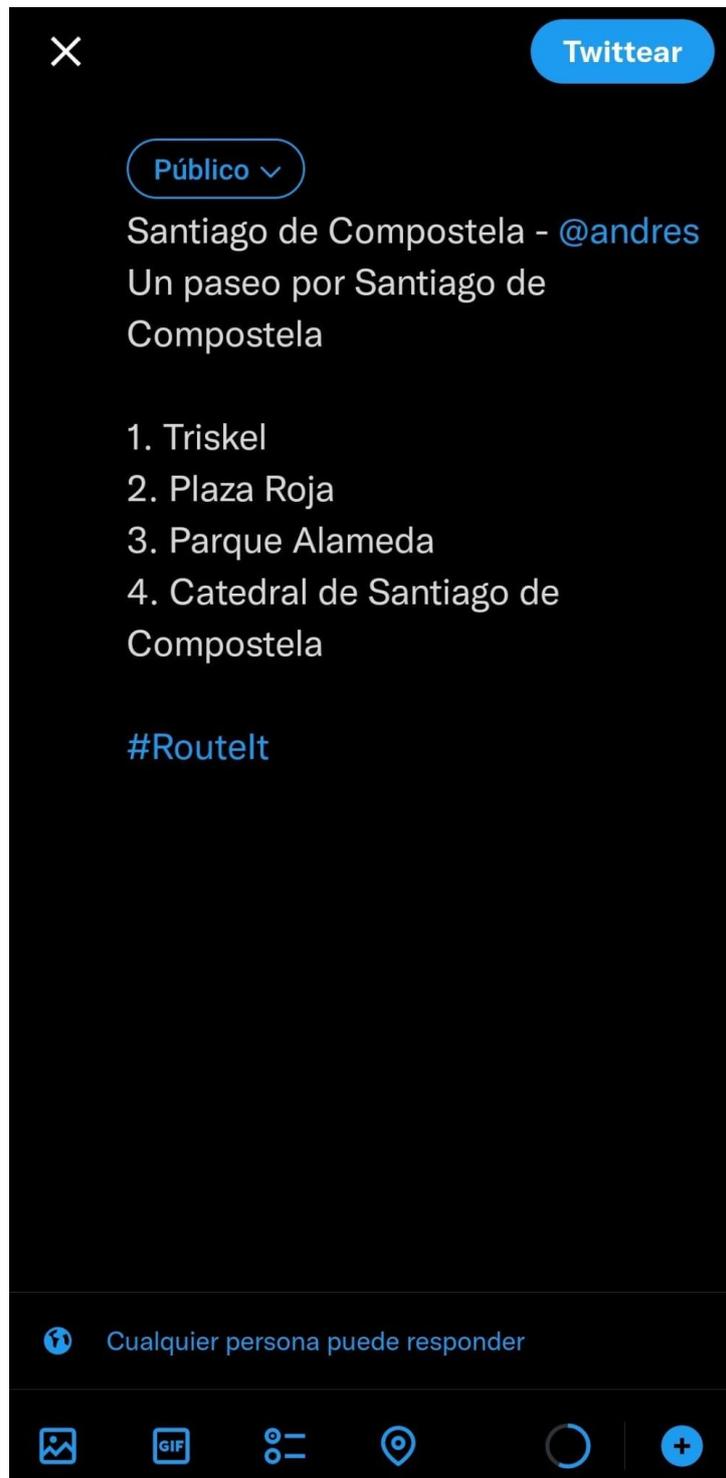


Figura 42: Compartir ruta

A la hora de visualizar las rutas, se mostrarán las ubicaciones de la misma como marcadores en el mapa y una pequeña leyenda a forma de ayuda que transportará a los usuarios hasta la ubicación seleccionada, dicha leyenda se puede ocultar si el usuario lo desea.

Se muestra la trayectoria de que hay que seguir para realizar la ruta siguiendo los pasos que ha seguido el creador de esta, esta trayectoria se puede ocultar y también se puede cambiar el tipo de transporte entre andando y en coche.

Asimismo, se puede visualizar la ubicación en tiempo real del usuario para que pueda seguir la trayectoria de la ruta desde el mismo mapa ubicándose en todo momento.

En la *Figura 43* se puede observar la pantalla del mapa de una ruta.



Figura 43: Ubicaciones de la ruta

Cabe destacar que es posible acceder a los detalles de cada ubicación como su valoración, dirección, horario de apertura, etc.

En la *Figura 44* se puede observar la pantalla del mapa mostrando los detalles de una ubicación.



Figura 44: Detalles de la ubicación

6. Conclusiones y líneas de trabajo futuras

Finalmente, se van a recoger las conclusiones obtenidas con la finalización del desarrollo del proyecto, añadiendo por último unas líneas de trabajo futuras con las que se podría continuar el desarrollo del proyecto.

6.1. Conclusiones

Con la finalización del proyecto se puede asegurar que este ha cumplido tanto los objetivos como los requisitos que se establecieron al principio del proyecto.

Entre los logros conseguidos se pueden destacar:

- Se ha cumplido el objetivo de gestión de usuarios, ya que permite realizar toda acción de entrada y salida al sistema con seguridad junto con una gestión óptima de relación entre usuarios.
- Se ha cumplido el objetivo de gestión de ubicaciones administrando correctamente las ubicaciones que se añaden a las rutas y obteniendo su información debidamente.
- Se ha cumplido el objetivo de gestión de rutas, el sistema almacena todas las rutas creadas en la base de datos y las muestra correctamente en las pantallas, pudiendo visualizarlas.
- Se ha logrado el objetivo de gestión de filtrado y búsqueda, por lo que los usuarios son capaces de llevar a cabo búsquedas de rutas y usuarios.
- Se ha logrado el objetivo de gestión de sistemas, ya que el sistema es capaz de administrar de forma eficaz los distintos tipos de sistemas que usa la aplicación.
- Se ha logrado desarrollar un sistema que es capaz de plasmar desde un hábito hasta una ruta con un sinnúmero de finalidades para esta y dando esta información a otras personas del mundo a las cuales les puede servir de gran utilidad.
- Se ha logrado la adaptación tanto de un nuevo entorno de desarrollo, como de un nuevo lenguaje de desarrollo con proyecciones a futuro.
- Se ha puesto a prueba el esfuerzo que supone el planteamiento, el desarrollo y la realización de un proyecto de forma individual, trabajando sobre cada área y aprendiendo al mismo tiempo la forma de trabajo.
- Se ha logrado trabajar con una de las herramientas que están tan normalizadas actualmente como lo es los servicios de Google, empleando sus bibliotecas y su forma de desarrollar productos, lo que supone un conocimiento muy valioso a la hora de formación y futuro trabajo.
- Por último, se ha conseguido aplicar todos los conocimientos y habilidades adquiridas durante todo el desarrollo del grado, aplicándolo a un proyecto final de grandes dimensiones.

6.2. Líneas de trabajo futuras

Los aspectos que podrían ser un siguiente desarrollo o se podrían mejorar del proyecto serían los siguientes:

- Ampliar las opciones a la hora de crear rutas, que se puedan crear rutas con ubicaciones que no sean un lugar ya existente, por ejemplo.
- Ampliar el sistema de filtrado de las rutas por ubicación, así poder buscar rutas en ciertas áreas.
- La posibilidad de seguir a personas y poder intercambiar mensajes entre usuarios, creando así una red social más potente.
- Mejorar el aspecto visual de la aplicación a uno con diseños más bonitos y actuales.
- Un sistema de ayuda y feedback mejorado para que el usuario este seguro de que todo un equipo detrás se preocupa por él.
- Implementar una base de datos propia, sin tener que usar una de terceros.
- Desarrollar la aplicación en otros sistemas operativos como iOS o una página web para todos los usuarios.

7. Referencias

- [1] Google. Documentación de Firebase. Disponible en: <https://firebase.google.com/docs/build>
- [2] Firebase. Google Firebase. Disponible en: <https://firebase.google.com/>
- [3] Android Studio. Android para desarrolladores. Disponible en: <https://developer.android.com/>
- [4] Kotlin. Kotlin Docs. Disponible en: <https://kotlinlang.org/docs/home.html>
- [5] Kotlin. Desarrolla aplicaciones Android con Kotlin. Disponible en: <https://developer.android.com/kotlin?hl=es-419>
- [6] Google. Documentación de Google Cloud. Disponible en: <https://cloud.google.com/docs>
- [7] Google. API de Places. Disponible en: <https://developers.google.com/maps/documentation/places/web-service>
- [8] Google. Api de Directions. Disponible en: <https://developers.google.com/maps/documentation/directions/overview>
- [9] Google. SDK de Maps para Android. Disponible en: <https://developers.google.com/maps/documentation/android-sdk/overview>
- [10] JetBrains. Document Kotlin code: KDoc and Dokka. [En línea]. Disponible en: <https://kotlinlang.org/docs/kotlin-doc.html>
- [11] Visual Paradigm. Guía del usuario de Visual Paradigm. Disponible en: <https://www.visual-paradigm.com/support/documents/vpuserguide.jsp>
- [12] “Gestión de Proyectos”, class notes for 101130, Departamento de Informática y Automática, Universidad de Salamanca, 2021.
- [13] “Gestión de Proyectos Informáticos”, class notes for G4011422, Departamento de Electrónica y Computación, Universidad de Santiago de Compostela, 2021.
- [14] PMI (Project Management Institute), A Guide to the Project Management Body of Knowledge (PMBOK Guide), 7th ed., 2021. <http://www.pmi.org>
- [15] “Ingeniería del Software I”, class notes for 101118, Departamento de Informática y Automática, Universidad de Salamanca, 2019.
- [16] A. Durán Toro and B. Bernández Jiménez, “Metodología para la Elicitación de Requisitos de Sistemas Software”, Universidad de Sevilla, España, Informe Técnico, 2000. [En línea]. Disponible en: <http://www.lsi.us.es/docs/informes/lsi-2000-10.pdf>
- [17] “Ingeniería del Software II”, class notes for 101122, Departamento de Informática y Automática, Universidad de Salamanca, 2019.