

# Memoria del proyecto.

## DESARROLLO DE APLICACIÓN MÓVIL DE GESTIÓN DE VIAJES EN GRUPO

**Trabajo de Fin de Grado**

INGENIERÍA INFORMÁTICA



**VNiVERSIDAD  
D SALAMANCA**

CAMPUS DE EXCELENCIA INTERNACIONAL

Julio de 2022

Autor

Javier Sánchez Bernardo

Tutor/a

Héctor Sánchez San Blas  
Juan Francisco De Santana  
Gabriel Villarrubia González



## Certificado de los tutores

D. Gabriel Villarrubia González, D. Héctor Sánchez San Blas y D. Juan Francisco de Paz Santana, profesores del Departamento de Informática y Automática de la Universidad de Salamanca.

HACEN CONSTAR:

Que el trabajo titulado 'Desarrollo de aplicación móvil de gestión de viajes en grupo' ha sido realizada por D. Javier Sánchez Bernardo, con el número de documento de identidad 70912052Q y constituye la memoria del proyecto realizado para la superación de la asignatura 'Trabajo de fin de Grado' de la titulación 'Grado en Ingeniería Informática'.

Y para que así conste a todos los efectos oportunos.

En Salamanca, a 5 de septiembre de 2022

D. Gabriel Villarrubia González

D. Héctor Sánchez San Blas

D. Juan Francisco de Paz Santana



## Resumen

Los seres humanos son sociables por su propia naturaleza, y es a través de esta convivencia en comunidad, de compartir experiencias y emociones con el resto como se desarrollan como personas y obtienen la felicidad. Las etapas más importantes en el desarrollo psicosocial de los humanos son la adolescencia y la juventud ya que están llenas de curiosidades y aprendizajes.

A lo largo del tiempo, pero sobre todo en la actualidad, una de las formas más comunes utilizadas por jóvenes y estudiantes de todo el mundo para vivir experiencias en compañía son los viajes. La globalización y las nuevas tecnologías han permitido que los viajes internacionales sean mucho más accesibles para estos grupos de personas, y por tanto este tipo de turismo está en auge.

El objetivo de este proyecto es, por tanto, desarrollar una aplicación móvil que ofrezca a este sector de población una serie de herramientas y funcionalidades que faciliten diversas situaciones o problemas que puedan plantearse a lo largo de estos viajes en grupo, como puede ser el reparto de gastos, haciéndolos de esta forma más satisfactorios.

Además, teniendo en cuenta que los jóvenes suelen tener un poder adquisitivo escaso, y que organizan los viajes buscando ahorrar la mayor cantidad de costes posible, la aplicación contará con la opción de buscar otros grupos de jóvenes con viajes organizados en su mismo destino. De esta forma se abre la posibilidad tanto de conocer gente de todos los lugares del mundo, abriendo un abanico de posibilidades infinito en cuanto a experiencias se refiere, además de la posibilidad de compartir posibles gastos en el caso de excursiones, alojamientos o transportes.

**Palabras clave:** viaje, jóvenes, grupo, gastos, herramienta, aplicación.

## Summary

Human beings are sociable by their very nature, and they develop as people and obtain happiness sharing experiences and emotions with others through this community coexistence. Adolescence and youth are the most important stages in the psychosocial development of humans because these are full of curiosity and learning.

Throughout time, but especially nowadays, travelling is one of the most common ways used by young people and students around the world to live experiences in company. Globalization and new technologies have made international trips more accessible to these groups of people, and this type of tourism is growing because of it.

The objective of this project is to develop a mobile application that offers to this sector of the population some tools and functionalities that help in different situations or problems that may occur during these group trips, such as the sharing of expenses, making these trips more satisfactory.

In addition, having in mind that young people don't usually have a high acquisitive power, and that they look to save as many costs as possible when organizing trips, the application will bring the option to search for other young people groups with organized trips at their same destination. This opens up the possibility of meeting people from all over the world, creating an infinite range of opportunities in terms of experiences, as well as the possibility of sharing expenses in the case of excursions, housing or transportation.

**Keywords:** trip, youth, group, expenses, tool, application.



## Índice de contenidos

1.	Introducción.....	- 1 -
2.	Estado del arte .....	- 3 -
3.	Objetivos.....	- 8 -
3.1.	Objetivos del sistema .....	- 8 -
3.2.	Objetivos personales .....	- 9 -
4.	Conceptos teóricos.....	- 10 -
4.1.	REST API .....	- 10 -
4.2.	Frontend .....	- 10 -
4.3.	Backend.....	- 10 -
4.4.	Framework y biblioteca .....	- 11 -
4.5.	Herramientas CASE.....	- 11 -
5.	Técnicas y herramientas .....	- 12 -
5.1.	Entornos de desarrollo .....	- 12 -
5.1.1.	Android Studio .....	- 12 -
5.1.2.	Visual Studio Code.....	- 12 -
5.2.	Lenguajes de programación y derivados.....	- 13 -
5.2.1.	Java .....	- 13 -
5.2.2.	XML .....	- 13 -
5.2.3.	JavaScript.....	- 13 -
5.2.4.	JSON.....	- 13 -
5.2.5.	UML .....	- 13 -
5.3.	Entornos de ejecución y despliegue.....	- 14 -
5.3.1.	NodeJS.....	- 14 -
5.3.2.	NPM.....	- 14 -
5.3.3.	MongoDB .....	- 14 -
5.4.	Frameworks, API's, middlewares, librerías y módulos.....	- 14 -
5.4.1.	ExpressJS.....	- 14 -
5.4.2.	PassportJS .....	- 14 -
5.4.3.	Mongoose .....	- 15 -
5.4.4.	BcryptJS.....	- 15 -
5.4.5.	JSDoc.....	- 15 -
5.4.6.	JSDoc-http-plugin.....	- 15 -
5.4.7.	Volley .....	- 15 -
5.5.	Herramientas CASE.....	- 15 -

5.5.1.	REM.....	- 15 -
5.5.2.	Visual Paradigm .....	- 16 -
5.5.3.	NoSQL DB Schema Modeling .....	- 16 -
5.5.4.	EZEstimate .....	- 16 -
5.5.5.	Microsoft Project .....	- 16 -
5.6.	Otras herramientas .....	- 16 -
5.6.1.	Javadoc .....	- 16 -
5.6.2.	Postman .....	- 16 -
5.6.3.	Photoshop cs6.....	- 17 -
5.6.4.	Bgenerator .....	- 17 -
6.	Aspectos relevantes del desarrollo .....	- 18 -
6.1.	Marco de trabajo .....	- 18 -
6.2.	Estimación del esfuerzo .....	- 19 -
6.3.	Planificación temporal .....	- 20 -
6.4.	Especificación de requisitos.....	- 23 -
6.4.1.	Participantes.....	- 23 -
6.4.2.	Objetivos del sistema.....	- 23 -
6.4.3.	Requisitos de almacenamiento de información .....	- 24 -
6.4.4.	Actores.....	- 25 -
6.4.5.	Casos de uso .....	- 25 -
6.4.6.	Requisitos no funcionales.....	- 29 -
6.5.	Análisis de requisitos .....	- 29 -
6.5.1.	Modelo del dominio.....	- 29 -
6.5.2.	Realización de casos de uso del análisis.....	- 30 -
6.5.3.	Clases del análisis.....	- 31 -
6.5.4.	Vista de arquitectura.....	- 32 -
6.6.	Diseño del sistema .....	- 33 -
6.6.1.	Patrón arquitectónico MVC.....	- 33 -
6.6.2.	Subsistemas de diseño .....	- 34 -
6.6.3.	Clases de diseño.....	- 35 -
6.6.4.	Vista arquitectónica de diseño.....	- 39 -
6.6.5.	Realización de casos de uso del diseño .....	- 39 -
6.6.6.	Diseño de la base de datos .....	- 41 -
6.6.7.	Modelo de despliegue.....	- 41 -
6.7.	Implementación.....	- 42 -
6.7.1.	Aplicación Android .....	- 42 -
6.7.2.	Servidor NodeJS .....	- 43 -

6.8.	Pruebas.....	- 43 -
6.9.	Funcionalidad del sistema.....	- 44 -
6.9.1.	Registro, inicio y cierre de sesión.....	- 44 -
6.9.2.	Agenda de viajes.....	- 46 -
6.9.3.	Menú del viaje .....	- 47 -
6.9.4.	Gestor de dinero.....	- 49 -
6.9.5.	Notas grupales .....	- 52 -
6.9.6.	Mensajes entre usuarios.....	- 53 -
6.9.7.	Buscador de viajes .....	- 55 -
6.9.8.	Opciones del administrador .....	- 56 -
7.	Limitaciones de la aplicación .....	- 58 -
8.	Conclusiones y líneas de trabajo futuras .....	- 59 -
8.1.	Conclusiones .....	- 59 -
8.2.	Líneas de trabajo futuras .....	- 60 -
9.	Referencias .....	- 61 -

## Índice de figuras

Figura 1. Aplicación Tricount .....	- 3 -
Figura 2. Aplicación Google Keep .....	- 4 -
Figura 3. Web Joinmytrip 1. Búsqueda de viajes .....	- 4 -
Figura 4. Web Joinmytrip 2. Creación de viajes .....	- 5 -
Figura 5. Aplicación BlaBlaCar .....	- 6 -
Figura 6. Aplicación TripIt.....	- 7 -
Figura 7. Fases del Proceso Unificado .....	- 19 -
Figura 8. Estimación del esfuerzo. EZEstimate .....	- 20 -
Figura 9. Calendario del proyecto. Microsoft Project .....	- 21 -
Figura 10. Diagrama de Gantt del desarrollo. Microsoft Project .....	- 22 -
Figura 11. Diagrama de actores .....	- 25 -
Figura 12. Diagrama de paquetes .....	- 26 -
Figura 13. Diagrama de casos de uso del paquete Gestión de usuarios .....	- 27 -
Figura 14. Diagrama de clases del modelo del dominio.....	- 30 -
Figura 15. Diagrama de secuencia del análisis. Registro nuevo usuario.....	- 31 -
Figura 16. Diagrama de comunicación del paquete Gestión de usuarios .....	- 32 -
Figura 17. Vista de arquitectura del modelo de análisis.....	- 33 -
Figura 18. Patrón MVC .....	- 34 -
Figura 19. Subsistemas de diseño.....	- 35 -
Figura 20. Clases de diseño 1. Vista aplicación .....	- 36 -
Figura 21. Clases de diseño 2. Controlador aplicación .....	- 37 -
Figura 22. Clases de diseño 3. Controlador servidor.....	- 38 -
Figura 23. Clases de diseño 4. Modelo servidor.....	- 38 -
Figura 24. Vista arquitectónica de diseño .....	- 39 -
Figura 25. Diagrama de secuencia del diseño. Registro nuevo usuario .....	- 40 -
Figura 26. Diagrama de diseño de la base de datos.....	- 41 -
Figura 27. Diagrama de despliegue del sistema .....	- 42 -
Figura 28. Pantallas de registro e inicio de sesión.....	- 45 -
Figura 29. Información del usuario y cierre de sesión .....	- 45 -
Figura 30. Agenda de viajes .....	- 46 -
Figura 31. Menús de creación y unión de viajes .....	- 47 -
Figura 32. Menú del viaje.....	- 47 -
Figura 33. Consultar información de viaje .....	- 48 -
Figura 34. Consultar usuarios del viaje .....	- 48 -

Figura 35. Gestor de gastos .....	- 49 -
Figura 36. Balance de usuarios del viaje .....	- 50 -
Figura 37. Historial de pagos.....	- 50 -
Figura 38. Modificación de pago.....	- 51 -
Figura 39. Eliminación de pago.....	- 51 -
Figura 40. Adición de pago .....	- 51 -
Figura 41. Notas grupales.....	- 52 -
Figura 42. Añadir nota.....	- 52 -
Figura 43. Desplegar nota.....	- 53 -
Figura 44. Editar nota .....	- 53 -
Figura 45. Enviar mensaje .....	- 54 -
Figura 46. Buzón de mensajes.....	- 54 -
Figura 47. Mensaje desplegado .....	- 55 -
Figura 48. Buscador de viajes.....	- 55 -
Figura 49. Información de viaje público .....	- 56 -
Figura 50. Búsqueda de viajes por filtros .....	- 56 -
Figura 51. Acceso a viaje – Administrador .....	- 57 -
Figura 52. Eliminar viaje - Administrador .....	- 57 -

## Índice de tablas

Tabla 1. Participante Javier Sánchez Bernardo .....	- 23 -
Tabla 2. Objetivo Gestión de viajes.....	- 24 -
Tabla 3. Requisito de información de usuario registrado.....	- 24 -
Tabla 4. Actor usuario no autenticado.....	- 25 -
Tabla 5. Caso de uso Registro nuevo usuario.....	- 28 -
Tabla 6. Requisito no funcional Escalabilidad .....	- 29 -

## 1. Introducción.

Los seres humanos son seres sociables por naturaleza, que necesitan pasar tiempo en grupo y compartir experiencias con los demás para su propio desarrollo como personas, en especial en la etapa de la juventud, que está llena de ambiciones.

Una de las formas más utilizadas por las personas que transitan esa etapa de la vida para saciar las ganas de aventuras es viajar, conocer todos los lugares y personas posibles alrededor del mundo acompañado de amigos.

El objetivo principal de este proyecto es desarrollar una aplicación móvil que ofrezca a jóvenes y estudiantes una serie de herramientas útiles en diversas situaciones durante la realización de estos viajes en grupo. Además, como este sector de población suele contar con poco poder adquisitivo, la aplicación ofrecerá la posibilidad de buscar grupos de personas que tengan planificados viajes similares, abriendo la posibilidad de conectar a gente de todos los lugares del mundo, y permitiendo compartir gastos en actividades, alojamientos o transportes.

Este documento recoge los puntos principales del desarrollo del proyecto, y consta de los siguientes apartados:

- **Estado del arte:** recoge el estudio de mercado hecho acerca de aplicaciones con funcionalidades similares disponibles.
- **Objetivos:** recoge tanto los objetivos del proyecto como los personales.
- **Conceptos teóricos:** contiene las explicaciones teóricas de los conceptos técnicos utilizados.
- **Técnicas y herramientas:** recoge todas las herramientas, técnicas y lenguajes utilizadas en el desarrollo.
- **Aspectos relevantes del desarrollo:** contiene un resumen de las partes más importantes de todo el proceso de desarrollo.
- **Limitaciones de la aplicación:** recoge los puntos que se consideran mejorables de la aplicación final.
- **Conclusiones y futuras líneas de trabajo:** contiene las conclusiones a las que se ha llegado tras la realización del proyecto, y ámbitos en los que se podría seguir trabajando para la mejora o mantenimiento del sistema.
- **Referencias.**

## Memoria del proyecto

A demás, se incluyen una serie de anexos que completan los apartados anteriores:

- **Anexo I, planificación temporal:** recoge la estimación del esfuerzo del proyecto y la planificación temporal del mismo
- **Anexo II, especificación de requisitos:** recoge la elicitación de requisitos software.
- **Anexo III, análisis de requisitos:** recoge la documentación relativa al proceso de análisis de requisitos.
- **Anexo IV, diseño del sistema:** recoge la documentación relativa al proceso de diseño del sistema final.
- **Anexo V, documentación técnica:** recoge las especificaciones del código fuente de la aplicación y el servidor.
- **Anexo VI, manual de usuario:** es un documento explicativo del funcionamiento de la aplicación para el usuario final.

## 2. Estado del arte

Uno de los pasos iniciales en el desarrollo del proyecto ha sido la investigación de plataformas y aplicaciones ya existentes en el mercado que cumplen algunas de las funciones que se quieren implementar, buscando así orientar de manera correcta su desarrollo. Las plataformas analizadas han sido las siguientes:

- **Tricount** (Tricount, 2022): es una aplicación móvil que permite crear grupos de personas para repartir gastos sin perder la cuenta global de las deudas entre ellos. Permite añadir pagos al grupo, que se reparten en función del número de personas implicadas en cada uno de esos pagos y la persona que los ha hecho. Permite consultar en todo momento los balances de los miembros y modificar los pagos. En la Figura 1 se observan los menús principales de la aplicación; el historial de pagos y los balances globales de los usuarios.

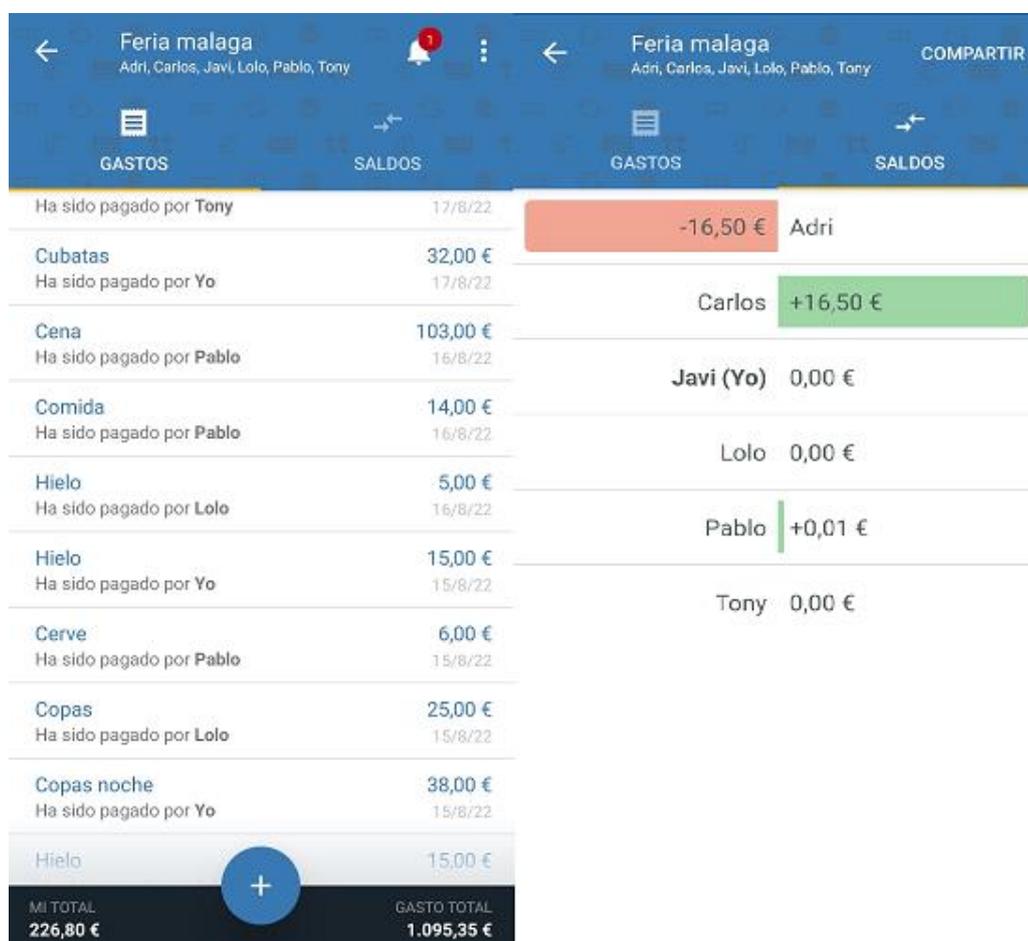


Figura 1. Aplicación Tricount

- **Keep** (Google, 2022): es la aplicación de notas de Google. Permite añadir usuarios como colaboradores de una nota para su edición en tiempo real. La Figura

2 muestra una nota colaborativa dentro de la aplicación.

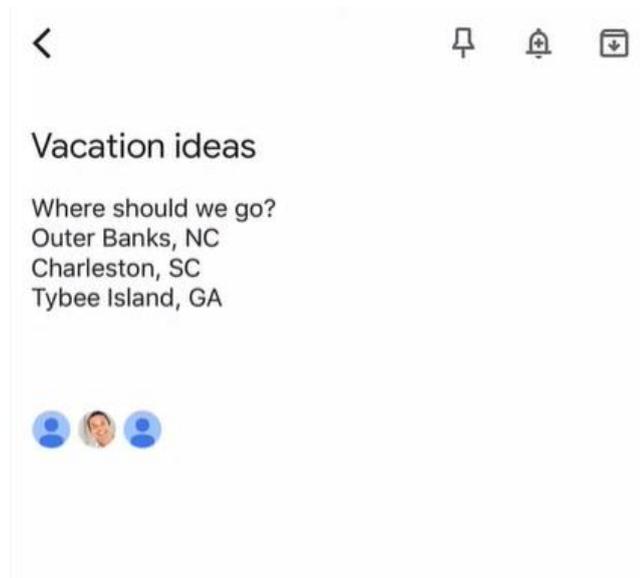


Figura 2. Aplicación Google Keep

- **Joinmytrip** (Joinmytrip, 2022): es una web dedicada a encontrar personas para hacer viajes. Permite crear un viaje especificando tus planes, fechas, destino y el tipo de viaje que vas a realizar y dejar que la gente se una a ti. También aporta la opción de buscar tú viajes que se ajusten a tus gustos y unirse a sus aventuras. La Figura 3 muestra el proceso de buscar un viaje, y la Figura 4 el de crear uno nuevo.

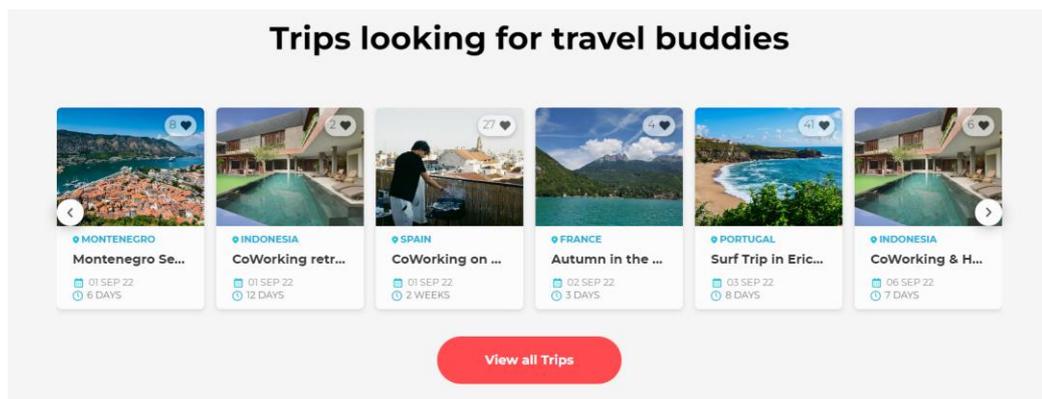


Figura 3. Web Joinmytrip 1. Búsqueda de viajes

**Hi Javier! Let's get started!**

**Departure date**

From\*  
31 Aug 2022

**Pick categories**

Food Wellness Beach Culture Party Sport Nature  
City Backpacking Female Only

**Type of trip**

**Guided**  
You're taking the lead and making all arrangements for those wanting to join.

**Coworking**  
Sharing accommodation and working space with other remote professionals.

BACK NEXT

Figura 4. Web Joinmytrip 2. Creación de viajes

- **BlaBlaCar** (BlaBlaCar, 2022): es una aplicación destinada a compartir los gastos de viajes en coche. Permite a sus usuarios compartir futuros desplazamientos que tengan programados, ofertando las plazas vacías de su vehículo por un precio. Ofrece la posibilidad de buscar estos viajes y contactar con su publicador mediante mensajes, facilitando la comunicación entre las partes para concretar la recogida, etc.

La Figura 5 muestra a la derecha el menú de resultado al buscar un viaje, con su precio y valoración del conductor, mientras que a la izquierda se muestra el menú principal, con tus viajes pendientes, en caso de que hayas reservado alguno, el menú de mensajes, perfil, etc.

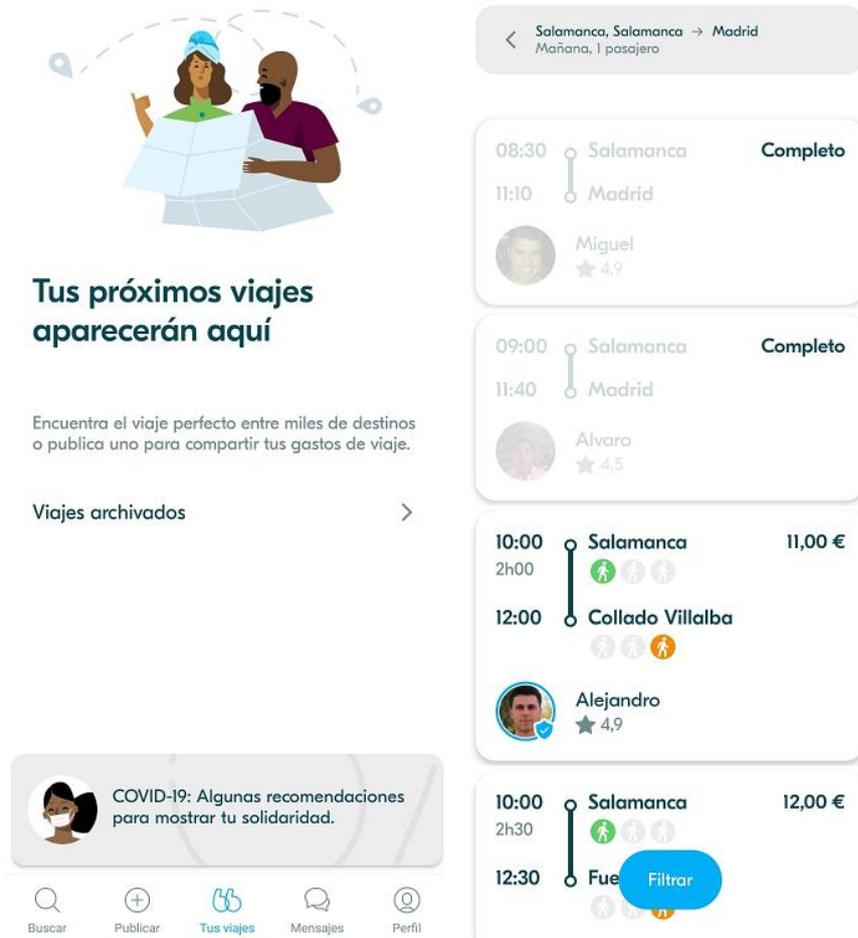


Figura 5. Aplicación BlaBlaCar

- **TripIt** (Concur Technologies, Inc, 2022): es una aplicación destinada a la organización de viajes en grupo. Permite a los usuarios crear viajes grupales, como se muestra en la parte izquierda de la Figura 6, una vez creado da la posibilidad de hacer una organización temporal con los distintos planes que se tengan a lo largo de dicho viaje, vuelos, actividades, lugares de interés, etc. Estas opciones, que se pueden ver en la parte derecha de la Figura 6, son añadidas y editadas por todos los miembros del viaje.

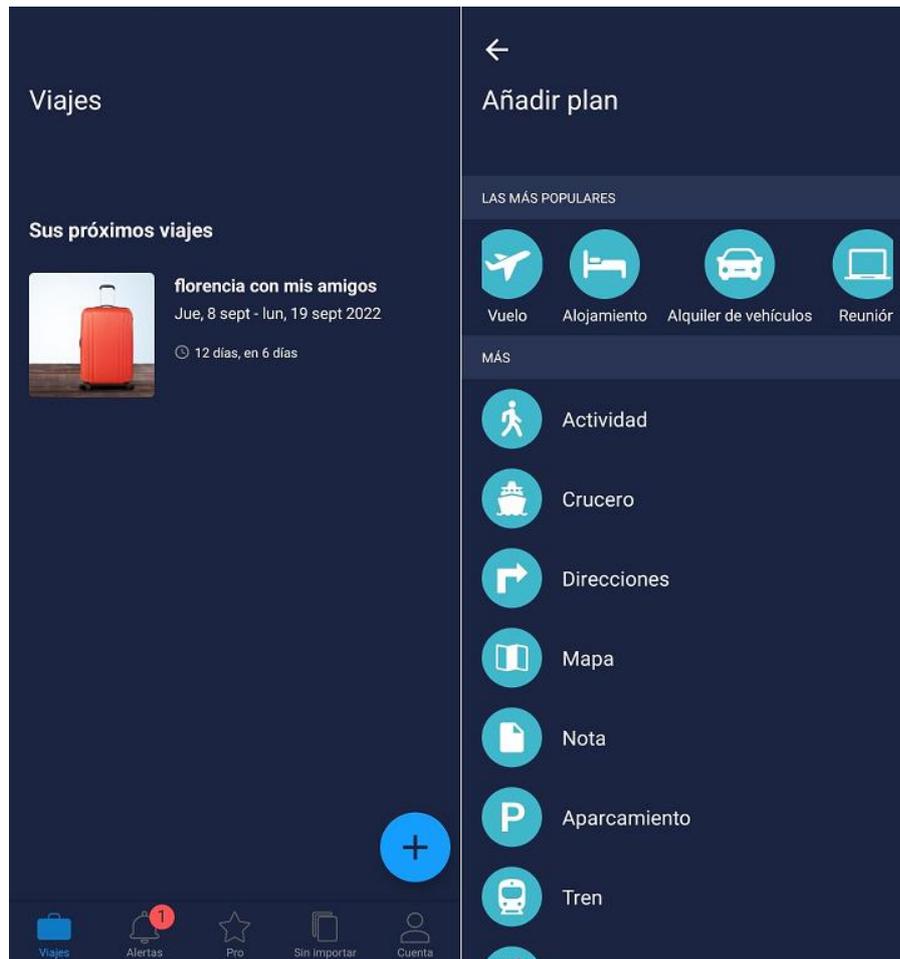


Figura 6. Aplicación TriplIt

### 3. Objetivos

Este apartado describe los objetivos que debe cumplir el sistema para la realización de este proyecto. A demás, se especifican los objetivos personales que se pretenden lograr.

#### 3.1. Objetivos del sistema

El objetivo principal del sistema es desarrollar una aplicación que permita a los usuarios crear grupos para viajes, proveyéndoles de herramientas para simplificar tareas durante el desarrollo de estos. Además, permitirá encontrar grupos de personas que hayan publicado sus viajes para poder conocer y compartir experiencias con ellos allá dónde vayas.

Los objetivos detallados del sistema son los siguientes:

- **Gestión de usuarios:** el sistema deberá gestionar el registro de nuevos usuarios en el servidor, así como la creación y eliminación de sesiones para dichos usuarios registrados.
- **Gestión de viajes:** el sistema deberá gestionar la creación de viajes por parte de los usuarios y la modificación de sus datos, así como su unión a viajes ya existentes.
- **Gestión de gastos:** el sistema deberá gestionar los gastos del viaje: adición, eliminación y modificación de pagos por los miembros, reparto equitativo de deudas, etc.
- **Gestión de notas:** el sistema deberá gestionar las notas grupales del viaje: adición, eliminación y modificación de estas por parte de cualquier miembro del grupo.
- **Gestión de búsqueda de viajes:** el sistema deberá permitir a los usuarios publicar sus viajes. También permitirá buscar dichos viajes mediante filtros al resto de usuarios registrados en la plataforma.
- **Gestión de mensajes:** el sistema deberá permitir a los usuarios enviar mensajes a los creadores de viajes públicos, así como a los miembros de sus viajes y responder a los mismos.

### 3.2. Objetivos personales

Este apartado explica los objetivos personales que me han motivado para realizar este proyecto como Trabajo de Fin de Grado, a forma de aunar todos los conocimientos obtenidos a lo largo de la carrera universitaria.

La idea inicial de este proyecto comenzó el verano antes del primer año de universidad, cuando planeaba un viaje con un grupo bastante grande de amigos. A lo largo de ese verano, y antes de que comenzase el viaje comencé a desarrollar una aplicación local bastante primitiva, que servía para repartir turnos de ducha durante el viaje. Ese proyecto se desechó rápidamente debido a la falta de conocimientos y a la poca utilidad que tenía realmente la aplicación, pero la idea de hacer una aplicación con herramientas que ayudasen a los viajes grupales con amigos se quedó dando vueltas por mi mente.

Cuando llegó la hora de plantear el TFG, esa idea salió de nuevo a la luz, pero mucho más madurada, las herramientas que tendría la aplicación serían más útiles, y, sobre todo, ahora que tenía los conocimientos necesarios para hacer que todos los usuarios pudiesen participar de forma online simultáneamente sobre el mismo viaje, la hacía tener mayor potencial. La parte de incluir en la aplicación un ‘foro’ o ‘red social’ que permite encontrar personas con planes similares alrededor del mundo completa la funcionalidad buscada a la perfección, conseguir que jóvenes y estudiantes puedan viajar de forma más cómoda y económica si contactan con otros grupos para compartir gastos.

El desarrollo de esta aplicación junto con su servidor agrupaba gran parte de los conocimientos obtenidos durante el grado, lo que la convirtió en la opción elegida para el proyecto.

## 4. Conceptos teóricos

En este apartado se definirán los distintos conceptos teóricos empleados en la memoria y anexos del trabajo, facilitando así su comprensión.

### 4.1. REST API

Es una interfaz de programación de aplicaciones (API) desarrollada en los márgenes de la arquitectura REST. Es decir, un conjunto de definiciones y protocolos que permiten a una aplicación solicitar información a un sistema o ejecutar una función, pero cumpliendo una serie de criterios necesarios para considerarse RESTful (Zepeda, 2022):

- Implementar una arquitectura cliente-servidor basada en peticiones HTTP.
- La información del cliente no se almacena entre peticiones, por tanto, debe ser una comunicación sin estado.
- Los datos son almacenables en caché para agilizar las comunicaciones.
- Debe ofrecer una interfaz uniforme para que la información se transmita de forma estandarizada.
- El servidor debe poder estructurarse en capas invisibles al cliente, de forma que pueda aumentar su eficiencia y escalabilidad.

### 4.2. Frontend

Es la parte de un programa o dispositivo que está del lado del cliente y que es accesible por el usuario final. Está compuesto por todas las tecnologías de diseño y desarrollo de interfaces que gestionan la interacción del usuario con el sistema (Maldeadora, 2022).

### 4.3. Backend

Es la capa que gestiona el acceso a los datos de un programa o dispositivo, y no es accesible por el usuario final. Contiene la lógica software encargada de manejar estos datos y proveerlos al frontend cuando sean solicitados, así como de realizar las posibles solicitudes al servidor (Maldeadora, 2022).

#### 4.4. Framework y biblioteca

Ambos son un conjunto de herramientas, módulos y funciones que permiten agilizar el trabajo en cuanto al desarrollo de software se refiere. La diferencia principal reside en el término inversión de control, esto significa que, mediante una biblioteca, el programador tiene el control de flujo de la aplicación, llamando a las funciones de esta cuando él lo requiera. Sin embargo, a la hora de utilizar un framework, este maneja el flujo de control de la aplicación, utilizando el código específico del programador cuando el propio framework considere necesario (Wozniewicz, 2022).

#### 4.5. Herramientas CASE

Las siglas CASE significan herramientas para ingeniería de software asistido por computadora. Permiten emplear de manera más fácil y eficiente las metodologías estructuradas de ingeniería del software, la realización de prototipos y mantenimiento de los programas. También ayudan a estandarizar la documentación, aumentando así su portabilidad y reutilización. En resumen, son sistemas software con herramientas que ayudan al diseño, desarrollo, administración y mantenimiento de proyectos durante todo el ciclo de vida de estos (Herramienta CASE, 2022).

## 5. Técnicas y herramientas

Este apartado describe las distintas herramientas de desarrollo, lenguajes de programación, *frameworks* y bibliotecas empleadas en el desarrollo del proyecto.

### 5.1. Entornos de desarrollo

#### 5.1.1. Android Studio

Es el entorno de desarrollo oficial para la producción de aplicaciones destinadas al sistema operativo Android. Este IDE está basado en IntelliJ IDEA (JetBrains, 2022), y permite el desarrollo tanto en Java como en Kotlin.

Ofrece gran cantidad de plantillas y herramientas que facilitan y agilizan la producción de las aplicaciones entre ellas depuradores y sistemas de análisis de errores (Google Developers, 2022).

Además, ofrece un emulador de Android integrado, lo que permite hacer pruebas de manera muy eficiente en toda clase de terminales, con distintas versiones del SO, resoluciones de pantalla, etc.

Se ha utilizado esta herramienta debido a que es la oficial para el desarrollo Android y a que es la que más posibilidades ofrece.

#### 5.1.2. Visual Studio Code

Es un editor de código fuente de software libre y multiplataforma. Debido a la infinidad de módulos y extensiones de los que dispone, está perfectamente adaptado para la programación y ejecución de código en casi cualquier lenguaje.

Dispone de una característica denominada *Intellisense* que facilita la edición, autocompletado y sintaxis del código, agilizando así el proceso de desarrollo.

También dispone de un depurador para encontrar y solucionar los posibles errores de código que se produzcan durante la producción del sistema (Microsoft, 2022).

Se ha utilizado esta herramienta debido a que es la más extendida en el desarrollo de aplicaciones en NodeJS, como es el caso del servidor REST, y a la cantidad de facilidades que ofrece.

## 5.2. Lenguajes de programación y derivados

### 5.2.1. Java

Es un lenguaje de programación orientado a objetos, y más simple que sus alternativas de más bajo nivel como C++. Es independiente del hardware, lo que ofrece una gran portabilidad, y se ejecuta de forma multihilo, aumentando así su rendimiento.

Es el lenguaje principal en el que se desarrollan las aplicaciones Android, que es el ámbito en el que se ha utilizado en este proyecto (Oracle, 2022).

### 5.2.2. XML

Es un lenguaje de marcado que define un conjunto de reglas para la codificación de documentos (Mozilla Foundation, 2022). Ha sido utilizado para la creación de todos los elementos visuales de la aplicación Android.

### 5.2.3. JavaScript

Es un lenguaje de programación de alto nivel, interpretado y poco tipado orientado a objetos. Su mayor uso se encuentra en páginas y aplicaciones web (Mozilla Foundation, 2022). En este proyecto ha sido utilizado para la creación del servidor REST que dota de gran parte de la funcionalidad a la aplicación móvil ofreciéndole métodos.

### 5.2.4. JSON

Es un formato de información estructurada muy utilizado en intercambio de datos entre cliente y servidor (ECMA International, 2017). Ha sido utilizado para el traspaso de información en las peticiones http que realiza la aplicación al servidor, tanto en el cuerpo de la consulta como en la respuesta.

### 5.2.5. UML

El lenguaje de modelado unificado es un lenguaje visual utilizado para documentar la arquitectura, diseño e implementación de todas las partes que conforman un sistema software. Se basa en diagramas que describen la estructura y el comportamiento del sistema, así como los objetos que lo componen (Booch, Rumbaugh, & Jacobson, 2006). Ha sido utilizado para llevar a cabo toda la parte de ingeniería de software del proyecto.

## 5.3. Entornos de ejecución y despliegue

### 5.3.1. NodeJS

Es un entorno de ejecución de JavaScript basado en Chrome v8 JavaScript, que aporta muchos beneficios y soluciona problemas propios del lenguaje original. Es utilizado para la mayoría de los *backends* implementados en JS ya que aporta gran fluidez bajo una alta demanda de peticiones (The OpenJS Foundation, 2022).

### 5.3.2. NPM

Es un administrador de paquetes y módulos en línea de comandos que funciona de la mano de NodeJS. Permite la instalación de bibliotecas y módulos en el proyecto y administra sus dependencias (NPM, Inc, 2022).

### 5.3.3. MongoDB

Es un sistema de almacenamiento de información o base de datos denominado NoSQL, lo que significa que, en lugar de guardar los datos en tablas y accederlos mediante SQL, se almacenan en documentos con formato BSON (similar a JSON) en un esquema dinámico. Tiene un gran potencial, ya que el acceso y las consultas se pueden hacer de todas las formas imaginables, incluso creadas por el usuario en forma de función de JavaScript (MongoDB, Inc., 2022). Es la base de datos utilizada por el servidor en el caso del sistema desarrollado en este proyecto, y la encargada de almacenar la información de todos los usuarios y viajes.

## 5.4. Frameworks, API's, middlewares, librerías y módulos

### 5.4.1. ExpressJS

Es un marco de desarrollo de aplicaciones web para NodeJS que proporciona una API con métodos útiles para el enrutamiento y de utilidad para HTTP, creación de sesiones, cookies y más (The OpenJS Foundation, 2022).

### 5.4.2. PassportJS

Es un middleware desarrollado para ExpressJS que permite implementar en tu aplicación web estrategias de autenticación y validación de usuarios de manera simple (Hanson, 2022).

#### 5.4.3. Mongoose

Es una biblioteca para NodeJS que permite crear modelos de datos para una base de datos MongoDB, así como hacer consultas, conversiones de tipos, etc. De esta forma automatiza y facilita muchas de las tareas relacionadas con la gestión de una base de datos (Automattic, 2022).

#### 5.4.4. BcryptJS

Es una biblioteca de encriptación de datos mediante hashing para NodeJS, aporta funciones que permiten encriptar, desencriptar y comparar datos mediante sus hashes de forma sencilla (Bartolomeo, Girish, & Wirtz, 2022).

#### 5.4.5. JSDoc

Es una herramienta implementada en un módulo de NPM que permite generar documentación técnica para aplicaciones JavaScript en forma de página web HTML mediante comentarios en el código fuente (Open Source project, 2022). Sirve para documentar métodos, clases, módulos, parámetros, etc.

#### 5.4.6. JSDoc-http-plugin

Es un plugin para JSDoc que permite documentar de forma más correcta las rutas HTTP de tu servidor REST, con sus parámetros en la URL o en el cuerpo de la petición y sus parámetros de respuesta (vmarchaud, 2022).

#### 5.4.7. Volley

Es una biblioteca HTTP desarrollada por Google, que facilita el uso de redes en aplicaciones Android. Incluye métodos para enviar, encolar y cancelar solicitudes a direcciones de internet, y una gran capacidad de personalización de estas (Google, 2022).

### 5.5. Herramientas CASE

#### 5.5.1. REM

Es una herramienta empleada en la elicitación de requisitos de software de un proyecto, siguiendo la metodología definida por Durán y Bernárdez para este propósito (Durán Toro, 2022).

### 5.5.2. Visual Paradigm

Es una herramienta de modelado de diagramas UML que permite capturar los requisitos y sus componentes de forma precisa en proyectos software (Visual Paradigm, 2022). Se ha utilizado para la creación de todos los diagramas de documentación de diseño, análisis e implementación del sistema especificados en los anexos correspondientes.

### 5.5.3. NoSQL DB Schema Modeling

Es una herramienta web que permite crear diagramas de la estructura de bases de datos NoSQL (NoSQL DB Schema Modeling, 2022). Ha sido utilizada para realizar el diseño de la base de datos MongoDB del sistema.

### 5.5.4. EZEstimate

Es una herramienta que genera un informe de la estimación del esfuerzo de un proyecto utilizando la metodología de puntos de casos de uso (Purushothaman, 2009).

### 5.5.5. Microsoft Project

Es una herramienta especializada en la administración de proyectos desarrollada por Microsoft. Permite establecer un plan para distribuir las tareas en el calendario del proyecto, optimizando la eficiencia del desarrollo (Microsoft, 2022). Se ha utilizado para crear el plan de proyecto y temporización del desarrollo del proyecto completo.

## 5.6. Otras herramientas

### 5.6.1. Javadoc

Es una utilidad desarrollada por Oracle e implementada de forma nativa en las últimas versiones de Android Studio. Permite generar documentación técnica en formato de página web HTML de forma automática a partir de comentarios en el código fuente. Sirve para documentar paquetes, clases, métodos, etc. Se ha utilizado para generar la documentación de la aplicación Android (Oracle, 2022).

### 5.6.2. Postman

Es una herramienta utilizada en el testeo de API's, ya que permite realizar peticiones HTTP personalizadas y capturar las respuestas, pudiendo así hacer las pruebas necesarias a servicios web (Postman, Inc, 2022). Se ha utilizado para hacer las pruebas relativas a la mayoría de los servicios REST del servidor del proyecto.

### 5.6.3. Photoshop cs6

Es una herramienta de diseño gráfico y edición de imágenes desarrollada por Adobe (Adobe, 2018). Se ha utilizado para la creación del logo de la aplicación y la edición de los fondos.

### 5.6.4. Bgenerator

Es una utilidad web que permite generar de forma automática fondos de pantalla en base a unos parámetros de color y formas establecidos (Background Generator, 2022). Se ha utilizado para generar los fondos de la aplicación móvil.

## 6. Aspectos relevantes del desarrollo

En este apartado se especifican los apartados más relevantes de cada una de las fases del desarrollo del proyecto.

### 6.1. Marco de trabajo

Para el desarrollo del proyecto se ha seguido el marco de trabajo del Proceso Unificado. Está dirigido por casos de uso, y establece un modelo iterativo e incremental basado en la arquitectura a lo largo de distintas etapas (Proceso Unificado de Desarrollo de Software, 2022). Sus principales características son:

- **Iterativo e incremental:** se divide en cuatro fases explicadas a continuación. Cada fase se divide, asimismo en una serie de iteraciones que van produciendo un incremento en el resultado final, produciendo mejoras o añadiendo funcionalidades. Fases del Proceso Unificado:
  - **Inicio:** durante la fase inicial se establece el modelo de negocio, se definen los requisitos y objetivos del sistema y se realiza la planificación temporal y estimación de esfuerzo del proyecto.
  - **Elaboración:** en esta fase se pulen los conceptos establecidos anteriormente para el proyecto, por lo que se terminan de incluir algunos requisitos y se reajustan las estimaciones.
  - **Construcción:** esta etapa abarca los últimos pasos del diseño y la mayor parte de la implementación del sistema, Cumpliendo así con los requisitos mínimos establecidos.
  - **Transición:** en esta última etapa el sistema debe estar terminado, listo para entregarse al usuario final y proceder a su uso. A demás se comenzará el desarrollo de nuevas funcionalidades requeridas tras su uso y al mantenimiento de las actuales.

En la Figura 7 se observan las múltiples fases del desarrollo y la distribución heterogénea de las distintas tareas durante las mismas:

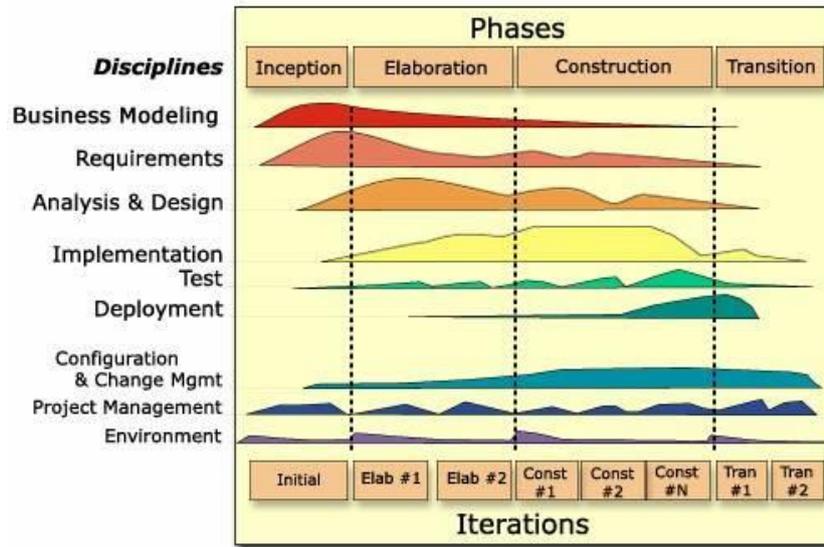


Figura 7. Fases del Proceso Unificado

(SIMULATION MODEL FOR RATIONAL UNIFIED PROCESS (RUP) SOFTWARE DEVELOPMENT LIFE CYCLE  
- Scientific Figure, 2022)

- **Dirigido por casos de uso:** el punto central del Proceso Unificado son los casos de uso, que establecen las acciones que realizará el sistema (requisitos funcionales). Estos definen, por tanto, los contenidos que se llevarán a cabo en cada iteración.
- **Centrado en la arquitectura:** existen multitud de modelos, esquemas y vistas que, en conjunto, definen la arquitectura de un sistema, ya que es imposible abarcar todos los aspectos de este mediante un único modelo.

## 6.2. Estimación del esfuerzo

Es una de las tareas iniciales en la planificación del proyecto, que tuvo como objetivo obtener una estimación del tiempo de desarrollo que tomaría este, y así asignarles a las distintas tareas un tiempo más realista en la planificación temporal.

Utilizando la herramienta EZEstimate se obtuvieron los resultados que muestra la Figura 8, unos 4 meses de desarrollo para una sola persona, empleando 6 horas diarias.

Los detalles sobre la estimación del esfuerzo y su cálculo se encuentran en el *Anexo I – Plan de proyecto*.

Estimation Summary	
UAW	9
UUCW	210
UUPC = UAW + UUCW	219
TFactor	23
EFactor	17
TCF = 0.6 + (.01*TFactor)	0,83
EF = 1.4 + (-0.03*EFactor)	0,89
UCP = UUCP*TCT*EF	161,7753
Total Effort@ 5 Hrs/UCP	808,8765

Figura 8. Estimación del esfuerzo. EZEstimate

### 6.3. Planificación temporal

La planificación temporal se llevó a cabo tras conocer la estimación de tiempo de desarrollo que requeriría el proyecto. Utilizando el marco de trabajo del Proceso Unificado explicado anteriormente, se realizó el calendario del proyecto, dividiendo las tareas en sucesivas iteraciones y asignándoles los tiempos previstos con anterioridad. Cada iteración se ha dividido en las siguientes fases de desarrollo:

- **Modelo de negocio:** se llevarán a cabo tareas relacionadas con la exploración del entorno y del mercado para buscar el enfoque idóneo del proyecto.
- **Requisitos:** se establecerán y documentarán los objetivos del sistema.
- **Análisis:** se estudiarán, describirán y documentarán los requisitos anteriores de forma precisa.
- **Diseño:** se describirá el funcionamiento de los diversos componentes del sistema para su posterior implementación.
- **Implementación:** se programarán los múltiples elementos del sistema.
- **Pruebas:** se realizarán las pruebas correspondientes a los elementos implementados.

Utilizando la herramienta Microsoft Project, se creó el calendario del proyecto (la Figura 9 muestra parte de este) y el diagrama de Gantt del desarrollo (la Figura 10 muestra parte de este).

## Memoria del proyecto

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
<b>Inicio</b>	<b>14,5 días</b>	<b>mar 01/03/22</b>	<b>mié 16/03/22</b>	
<b>Iteración 1</b>	<b>9 días</b>	<b>mar 01/03/22</b>	<b>jue 10/03/22</b>	
<b>Modelo de negocio</b>	<b>1 día</b>	<b>mar 01/03/22</b>	<b>mié 02/03/22</b>	
Definición de objetivos del proyecto	1 día	mar 01/03/22	mié 02/03/22	
<b>Requisitos</b>	<b>3 días</b>	<b>mié 02/03/22</b>	<b>vie 04/03/22</b>	<b>3</b>
Identificar actores y casos de uso	2 días	mié 02/03/22	jue 03/03/22	
Detallar casos de uso críticos	1 día	jue 03/03/22	vie 04/03/22	6
<b>Análisis</b>	<b>2 días</b>	<b>vie 04/03/22</b>	<b>lun 07/03/22</b>	<b>5</b>
Completar casos de uso	1 día	vie 04/03/22	lun 07/03/22	
Determinar clases del análisis	1 día	lun 07/03/22	lun 07/03/22	9
<b>Diseño</b>	<b>2 días</b>	<b>mar 08/03/22</b>	<b>mié 09/03/22</b>	<b>8</b>
Diseñar la interfaz de usuario	2 días	mar 08/03/22	mié 09/03/22	
<b>Implementación</b>	<b>1 día</b>	<b>mié 09/03/22</b>	<b>jue 10/03/22</b>	<b>11</b>
Definición de los lenguajes de programación	1 día	mié 09/03/22	jue 10/03/22	
Hito fin Iteración 1	0 días	jue 10/03/22	jue 10/03/22	14
<b>Iteración 2</b>	<b>5,5 días</b>	<b>jue 10/03/22</b>	<b>mié 16/03/22</b>	<b>2</b>
<b>Modelo de negocio</b>	<b>1 día</b>	<b>jue 10/03/22</b>	<b>jue 10/03/22</b>	<b>13</b>
Estudio de aplicaciones con funcionalidades similares	1 día	jue 10/03/22	jue 10/03/22	
<b>Requisitos</b>	<b>1,5 días</b>	<b>vie 11/03/22</b>	<b>lun 14/03/22</b>	<b>17</b>
Definir requisitos no funcionales	1 día	vie 11/03/22	vie 11/03/22	
Definir requisitos de información	0,5 días	vie 11/03/22	lun 14/03/22	20
<b>Análisis</b>	<b>1 día</b>	<b>lun 14/03/22</b>	<b>lun 14/03/22</b>	<b>19</b>
Paquetes de análisis	1 día	lun 14/03/22	lun 14/03/22	
<b>Diseño</b>	<b>1 día</b>	<b>lun 14/03/22</b>	<b>mar 15/03/22</b>	<b>22</b>

Figura 9. Calendario del proyecto. Microsoft Project

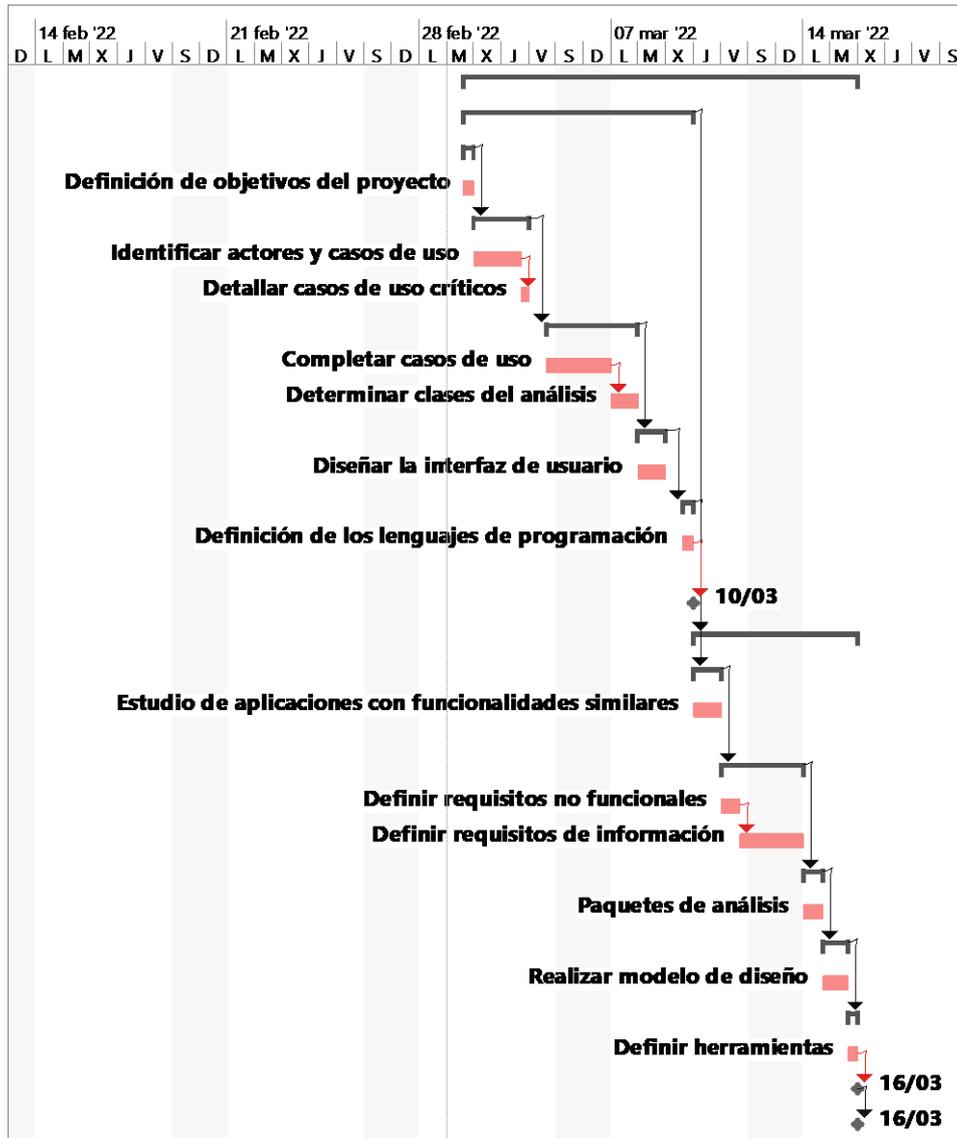


Figura 10. Diagrama de Gantt del desarrollo. Microsoft Project

Los detalles sobre la planificación temporal, y el calendario y diagrama completos se encuentran en el *Anexo I – Plan de proyecto*.

## 6.4. Especificación de requisitos

Esta es una de las fases iniciales del desarrollo del sistema, en ella se recogen los objetivos, requisitos, actores, y casos de uso que conformarán el mismo. La especificación de requisitos se realiza siguiendo la metodología de Durán y Bernárdez (Durán Toro & Bernárdez Jiménez, Metodología para la Elicitación de Requisitos de Sistemas Software, 2002) y utilizando las herramientas REM y Visual Paradigm para su documentación. A continuación, se va a hacer un repaso superficial de estos requisitos, que pueden ser consultados de manera completa en el *Anexo II – Especificación de requisitos del sistema*.

### 6.4.1. Participantes

Las personas involucradas en el desarrollo del proyecto son las siguientes:

- Javier Sánchez Bernardo
- Héctor Sánchez San Blas
- Juan Francisco de Paz Santana
- Gabriel Villarrubia González

La Tabla 1 muestra un ejemplo de las tablas de los participantes.

Participante	Javier Sánchez Bernardo
Organización	Universidad de Salamanca
Rol	Autor
Es desarrollador	Sí
Comentarios	Ninguno

Tabla 1. Participante Javier Sánchez Bernardo

### 6.4.2. Objetivos del sistema

Los objetivos que debe cumplir el sistema son los siguientes:

- Gestión de usuarios.
- Gestión de viajes.
- Gestión de gastos.
- Gestión de notas.
- Gestión de búsqueda de viajes.
- Gestión de mensajes.

Estos objetivos han sido detallados en el apartado *3.1 – Objetivos del sistema*.

La Tabla 2 muestra un ejemplo de las tablas de objetivos.

<b>OBJ-0002</b>	<b>Gestión de viajes</b>
<b>Versión</b>	1.0 (12/03/2022)
<b>Autores</b>	Javier Sánchez Bernardo
<b>Descripción</b>	El sistema deberá gestionar la creación de viajes por parte de los usuarios y la modificación de sus datos, así como su unión a viajes ya existentes.
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	Ninguno

Tabla 2. Objetivo Gestión de viajes

### 6.4.3. Requisitos de almacenamiento de información

Estos requisitos definen la información que deberá almacenar el sistema, en este caso son los siguientes:

- Información de usuario registrado.
- Información de viajes.
- Información de sesión.
- Información de notas.
- Información de gastos.
- Información de mensajes.

La Tabla 3 muestra un ejemplo de las tablas de los requisitos de información.

<b>IRQ-0001</b>	<b>Información de usuario registrado</b>
<b>Versión</b>	1.0 (13/03/2022)
<b>Autores</b>	Javier Sánchez Bernardo
<b>Objetivos y requisitos asociados</b>	[OBJ-0001] Gestión de usuarios
<b>Descripción</b>	El sistema deberá almacenar la información relativa a los usuarios que se hayan registrado en el sistema.
<b>Datos específicos</b>	ID Nombre de usuario Email Contraseña Roles Fecha de creación
<b>Estado</b>	Validado
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	Ninguno

Tabla 3. Requisito de información de usuario registrado

#### 6.4.4. Actores

Los actores que participan en el sistema son los siguientes:

- **Usuario no autenticado:** es aquel que no tiene una cuenta en el sistema, o que no ha iniciado sesión en este.
- **Usuario autenticado:** es aquel que tiene una cuenta en el sistema y ha iniciado sesión, obteniendo así acceso.
- **Usuario administrado:** es un usuario con cuenta en el sistema, sesión iniciada, y el rol de administrador asignado, lo cual le dota de más funcionalidades.

La Tabla 4 muestra un ejemplo de las tablas de actores, y la Figura 11 muestra la jerarquía que siguen estos actores.

<b>ACT-0001</b>	<b>Usuario no autenticado</b>
<b>Versión</b>	1.0 (12/03/2022)
<b>Autores</b>	Javier Sánchez Bernardo
<b>Descripción</b>	Este actor representa un usuario que intenta acceder a la aplicación, pero no está registrado en el sistema o no ha iniciado una sesión en el mismo.
<b>Comentarios</b>	Ninguno

Tabla 4. Actor usuario no autenticado

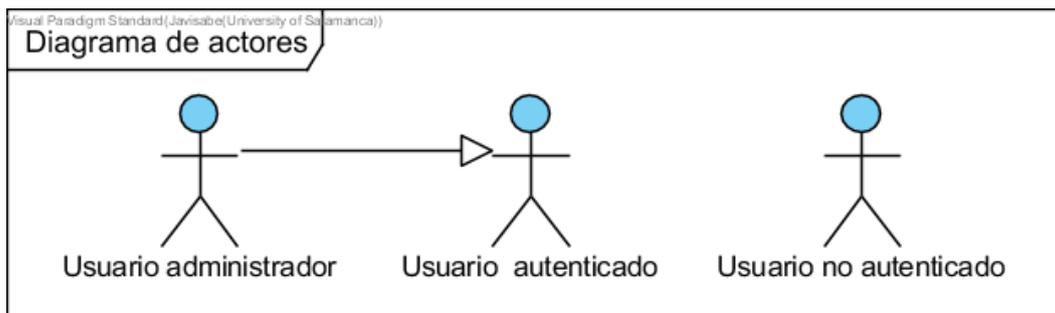


Figura 11. Diagrama de actores

#### 6.4.5. Casos de uso

Para una mejor organización del desarrollo, el sistema está dividido en paquetes de funcionalidad, que coinciden con los objetivos de este.

La Figura 12 representa el diagrama de paquetes, que muestra una visión global del sistema.

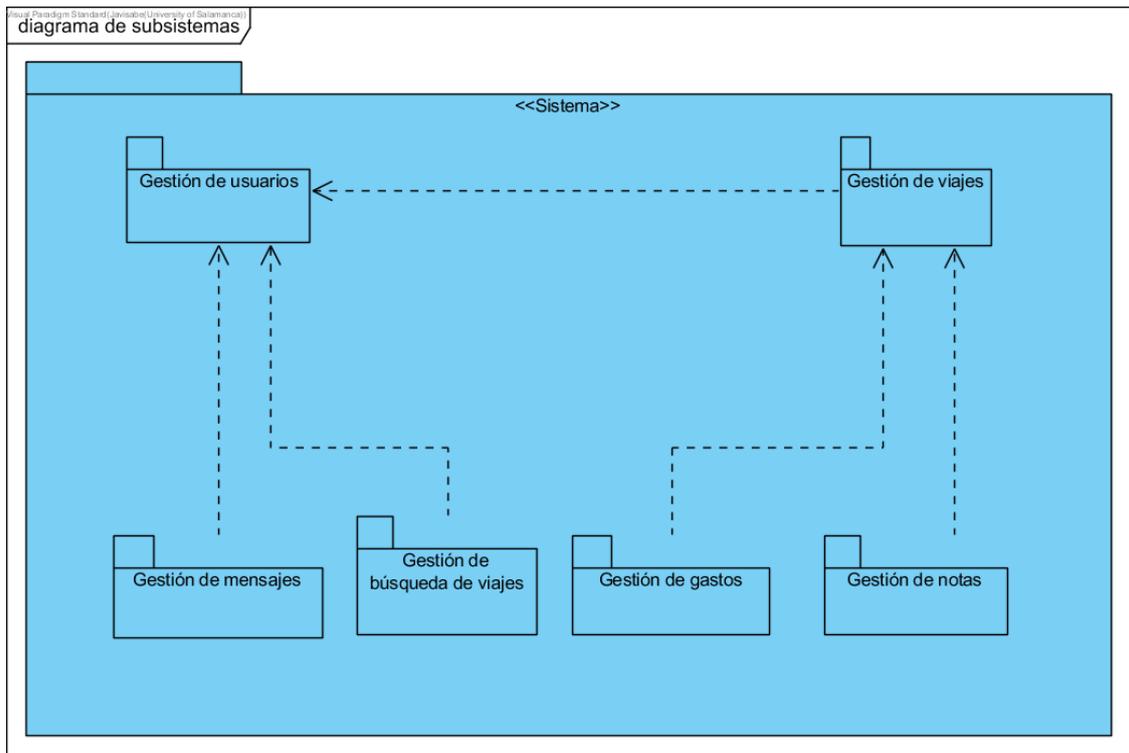


Figura 12. Diagrama de paquetes

Para la especificación de los casos de uso se han realizado diagramas de cada uno de los paquetes, como el que se muestra en la Figura 13, y después se han especificado todos y cada uno de los casos de uso mediante tablas descriptivas, como la que se muestra en la Tabla 5.

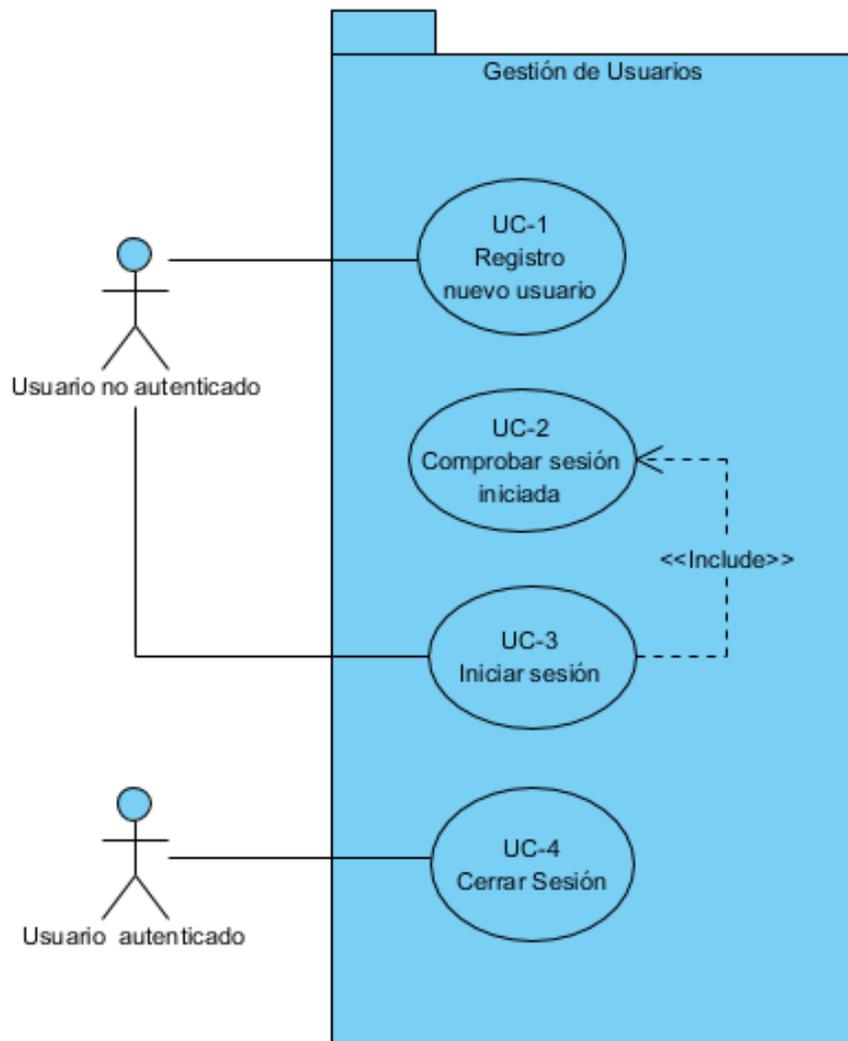


Figura 13. Diagrama de casos de uso del paquete Gestión de usuarios

UC-0001		Registro nuevo usuario	
<b>Versión</b>	1.0 (14/03/2022)		
<b>Autores</b>	Javier Sánchez Bernardo		
<b>Dependencias</b>	<a href="#">[OBJ-0001] Gestión de usuarios</a> <a href="#">[IRQ-0001] Información de usuario registrado</a> <a href="#">[NFR-0003] Escalabilidad</a>		
<b>Descripción</b>	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario solicite su registro en el sistema		
<b>Precondición</b>	El usuario no debe estar registrado		
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>	
	1	El actor <a href="#">Usuario no autenticado (ACT-0001)</a> solicita al sistema registrar un nuevo usuario.	
	2	El sistema solicita al usuario los datos pertinentes.	
	3	El actor <a href="#">Usuario no autenticado (ACT-0001)</a> aporta los datos requeridos por el sistema.	
	4	El sistema comprueba que los datos sean correctos.	
	5	El sistema almacena los datos en el sistema y crea el nuevo usuario.	
	6	El sistema informa al usuario de que la operación se ha completado satisfactoriamente.	
<b>Postcondición</b>	El usuario pasa a estar registrado en el sistema		
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>	
	4	Si el sistema encuentra algún error en los datos introducidos, informa al usuario y el caso de uso queda sin efecto.	
<b>Importancia</b>	Vital		
<b>Estado</b>	Validado		
<b>Estabilidad</b>	Alta		
<b>Comentarios</b>	Ninguno		

Tabla 5. Caso de uso Registro nuevo usuario

#### 6.4.6. Requisitos no funcionales

Los requisitos no funcionales establecen restricciones y normas a la hora del desarrollo del sistema, en el caso de este proyecto son los siguientes:

- Entorno Android.
- Servidor REST.
- Escalabilidad.
- Experiencia de usuario.
- Almacenamiento.
- Tolerancia a fallos.

La Tabla 6 muestra un ejemplo de especificación de un requisito no funcional.

<b>NFR-0003</b>	<b>Escalabilidad</b>
<b>Versión</b>	1.0 (14/03/2022)
<b>Autores</b>	Javier Sánchez Bernardo
<b>Dependencias</b>	Ninguno
<b>Descripción</b>	El sistema debe facilitar su escalabilidad, en cuanto a capacidad del servidor en caso de aumento del número de usuarios, y en cuanto a funcionalidad, en caso de requerir actualizaciones.
<b>Importancia</b>	Importante
<b>Urgencia</b>	Alta
<b>Estado</b>	Validado
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	Ninguno

Tabla 6. Requisito no funcional Escalabilidad

#### 6.5. Análisis de requisitos

En esta fase se procede a analizar de manera profunda los requisitos especificados anteriormente mediante los modelos y diagramas que se expondrán a continuación.

De la misma forma que antes, se expondrá superficialmente el análisis, ya que la información y diagramas al completo se especifican en el *Anexo III – Análisis de requisitos*.

##### 6.5.1. Modelo del dominio

El modelo del dominio recoge las necesidades impuestas por los requisitos de almacenamiento de información y las transforma en entidades. Relaciona dichas

entidades entre sí y especifica sus atributos. Se utiliza el diagrama de clases de la Figura 14 para representar este modelo.

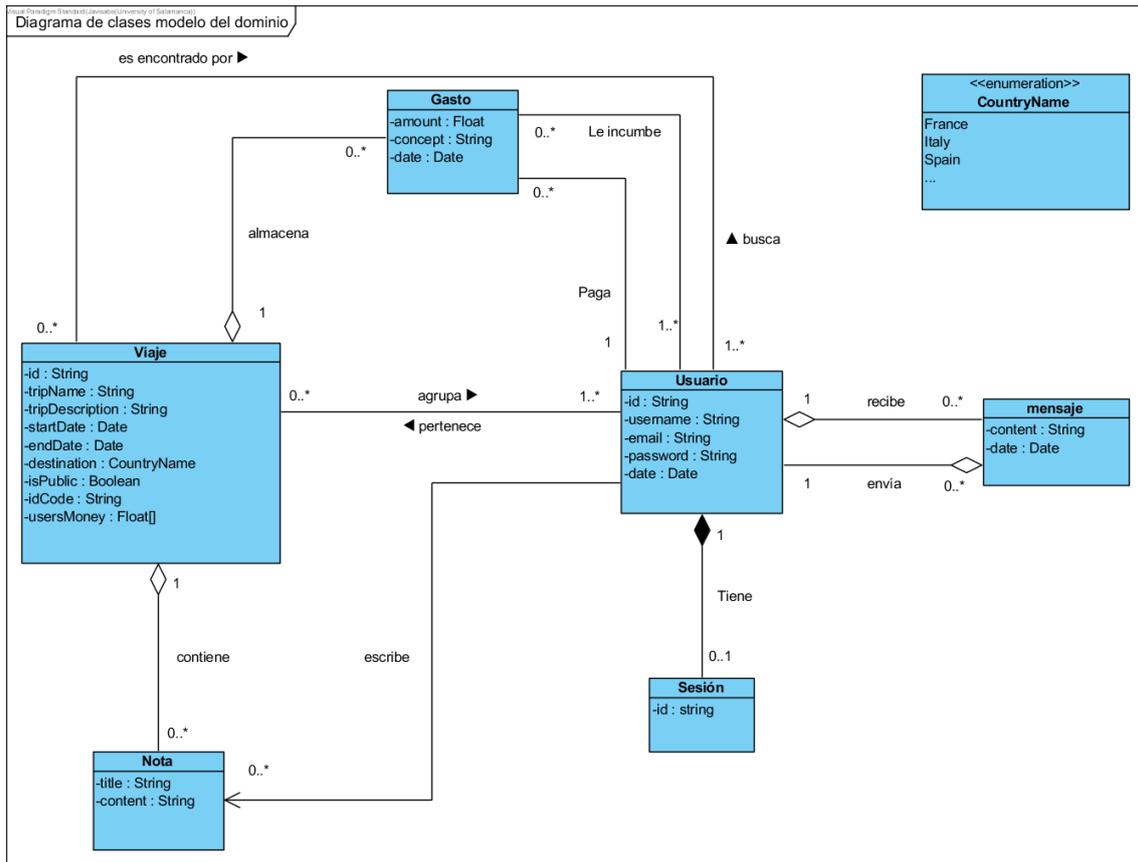


Figura 14. Diagrama de clases del modelo del dominio

### 6.5.2. Realización de casos de uso del análisis

En esta fase de análisis se procede a especificar los casos de uso obtenidos en la fase de captura de requisitos mediante diagramas de secuencia, definiendo así las iteraciones y mensajes entre objetos del sistema.

La Figura 15 muestra uno de estos diagramas como ejemplo.

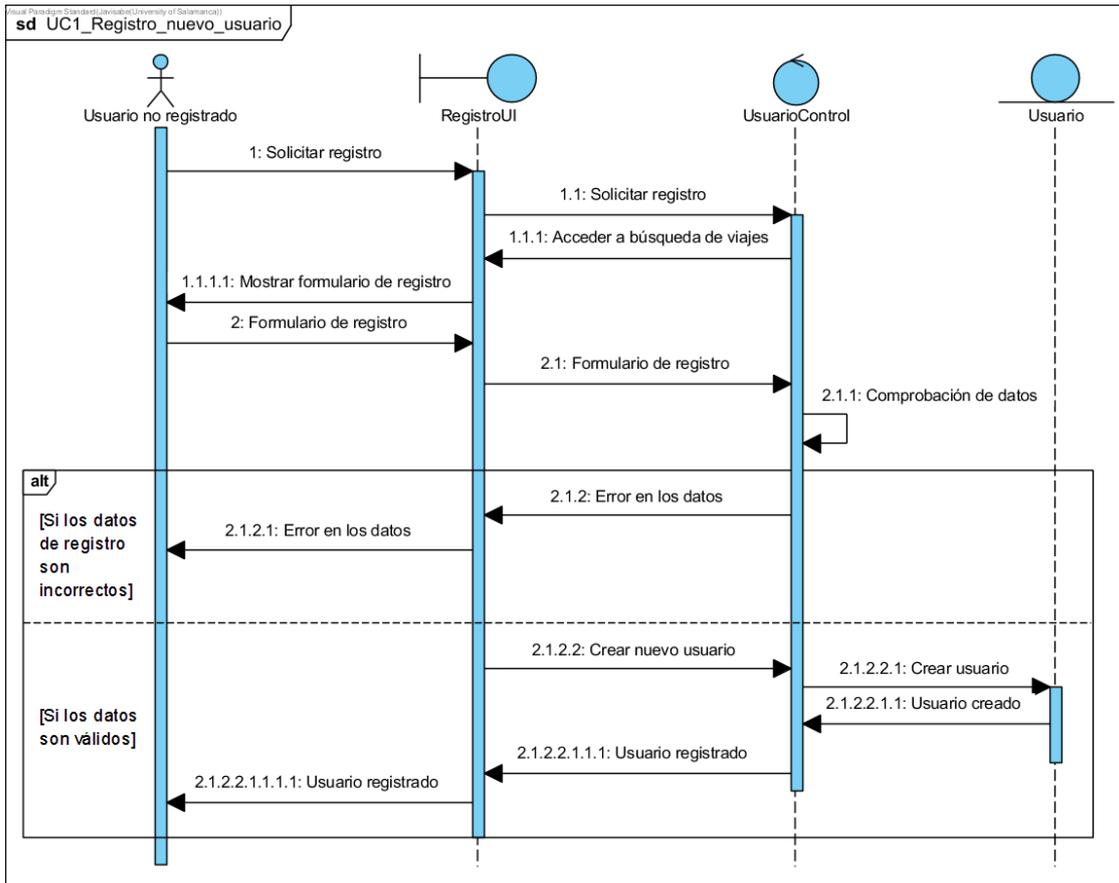


Figura 15. Diagrama de secuencia del análisis. Registro nuevo usuario

### 6.5.3. Clases del análisis

Mediante diagramas de comunicación, se muestran las relaciones de colaboración entre los objetos del sistema representados anteriormente, divididos dentro de los diferentes paquetes que lo componen.

La Figura 16 muestra el diagrama del paquete de gestión de usuarios como ejemplo.

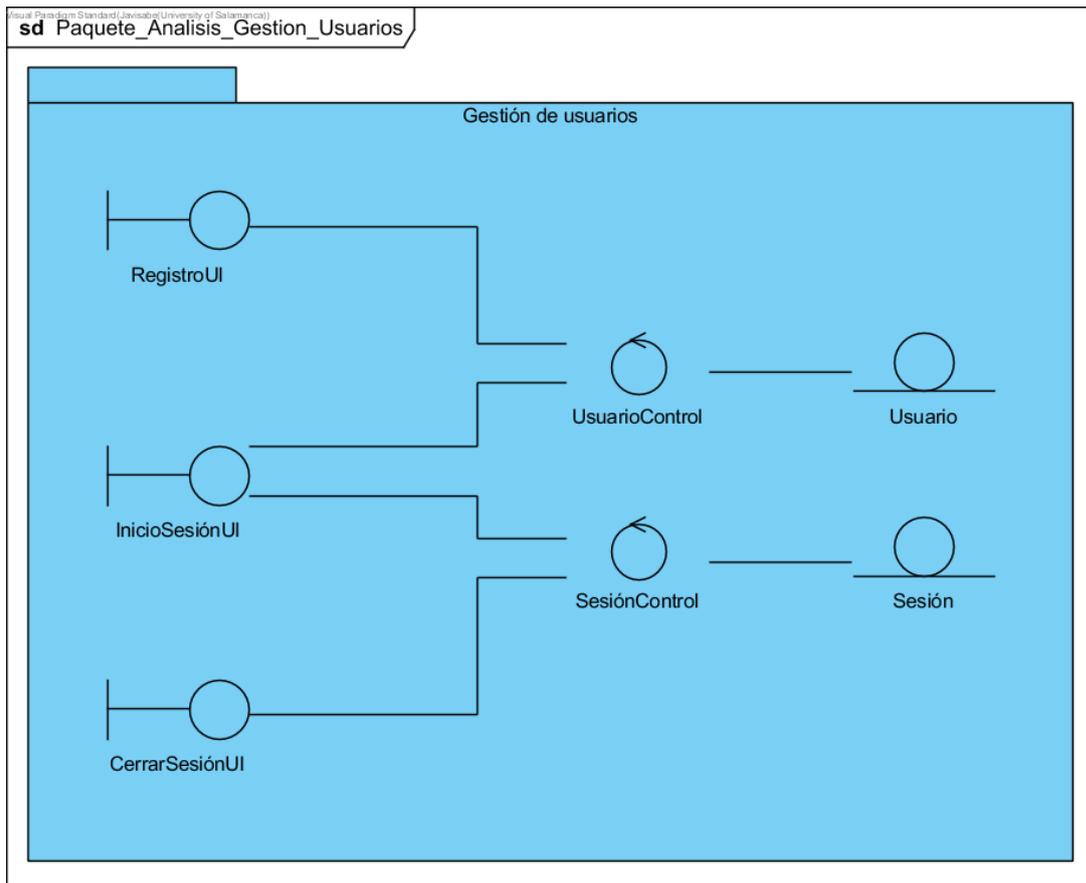


Figura 16. Diagrama de comunicación del paquete Gestión de usuarios

#### 6.5.4. Vista de arquitectura

El objetivo de este apartado es representar la arquitectura del sistema mediante los objetos que lo componen y sus relaciones, dentro del patrón arquitectónico Modelo-Vista-Controlador. La Figura 17 muestra esta vista de arquitectura del modelo de análisis.

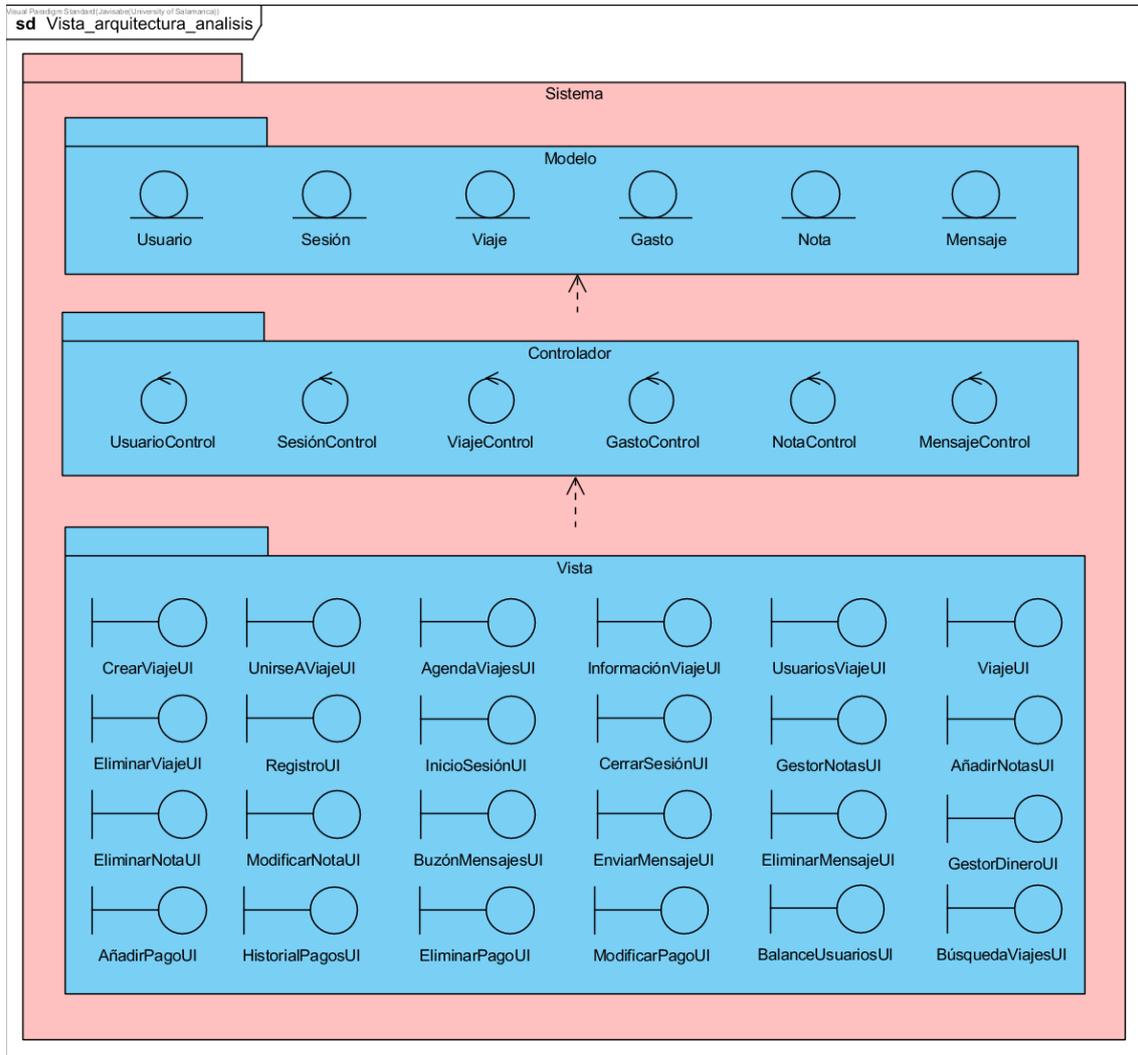


Figura 17. Vista de arquitectura del modelo de análisis

## 6.6. Diseño del sistema

Esta etapa del desarrollo se centra en describir el dominio de la solución en base al análisis del sistema previo. Es la fase previa a la implementación, y, por tanto, busca obtener un diseño lo más aproximado posible a la solución final, incluidos nombres de archivos, métodos y atributos.

La documentación completa y diagramas correspondientes a todos los siguientes subapartados del diseño pueden consultarse en el *Anexo IV – Diseño del sistema*.

### 6.6.1. Patrón arquitectónico MVC

Para la construcción de la estructura del sistema software se ha seguido el patrón arquitectónico Modelo Vista Controlador. El principio de este patrón es dividir en tres componentes fundamentales el programa, para así desacoplar, en gran medida, la interfaz

de usuario, de la lógica del programa y de los datos. Esto facilita el mantenimiento, la comprensibilidad y la escalabilidad del sistema, debido a que la dependencia entre las partes es muy baja (Aguilar, 2022).

Las tres partes en las que se divide son las siguientes:

- **Modelo:** es el encargado de gestionar la información del sistema y los accesos a la misma, en este caso, a la base de datos.
- **Vista:** se encarga de mostrar la información al usuario y permite a este interactuar con el sistema.
- **Controlador:** es el encargado de gestionar los eventos y acciones iniciadas por el usuario mediante la vista, solicitando al modelo la información correspondiente y devolviéndola a la vista, es decir, es el nexo entre el modelo y la vista, y dota de funcionalidad al sistema.

La Figura 18 representa las relaciones entre las distintas partes que componen el patrón.

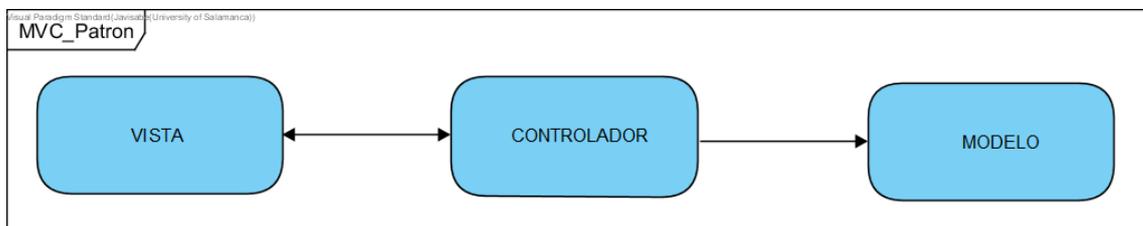


Figura 18. Patrón MVC

### 6.6.2. Subsistemas de diseño

El sistema ha sido dividido en dos paquetes principales, que representan la aplicación móvil y el servidor. El patrón MVC queda, por tanto, dividido entre ambos subsistemas; la vista pertenece a la aplicación, el modelo al servidor, que es quien gestiona la base de datos, y el controlador está dividido en dos partes que actúan de manera conjunta, una en la aplicación y otra en el servidor. La Figura 19 muestra el diagrama de paquetes de los subsistemas.

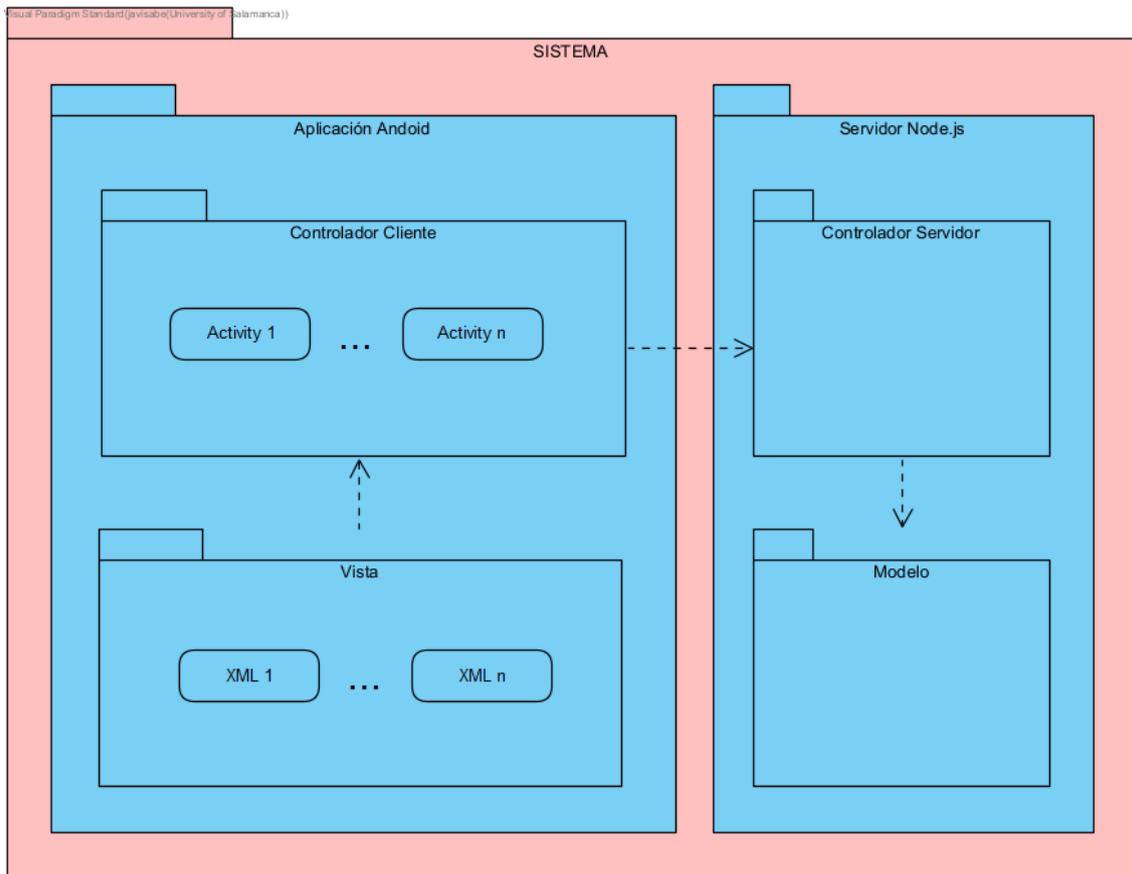


Figura 19. Subsistemas de diseño

### 6.6.3. Clases de diseño

En el siguiente apartado se describen los componentes de cada uno de los paquetes mostrados anteriormente, con sus clases y métodos específicos.

La Figura 20 muestra la vista de la aplicación, que se divide en ‘Views’ o pantallas principales y ‘Dialogs’ o menús emergentes. La Figura 21 muestra el controlador del lado de la aplicación, encargado de gestionar la vista mediante ‘Activities’ tanto para las pantallas principales como para los menús emergentes, además de contener algunas clases con métodos de utilidad.

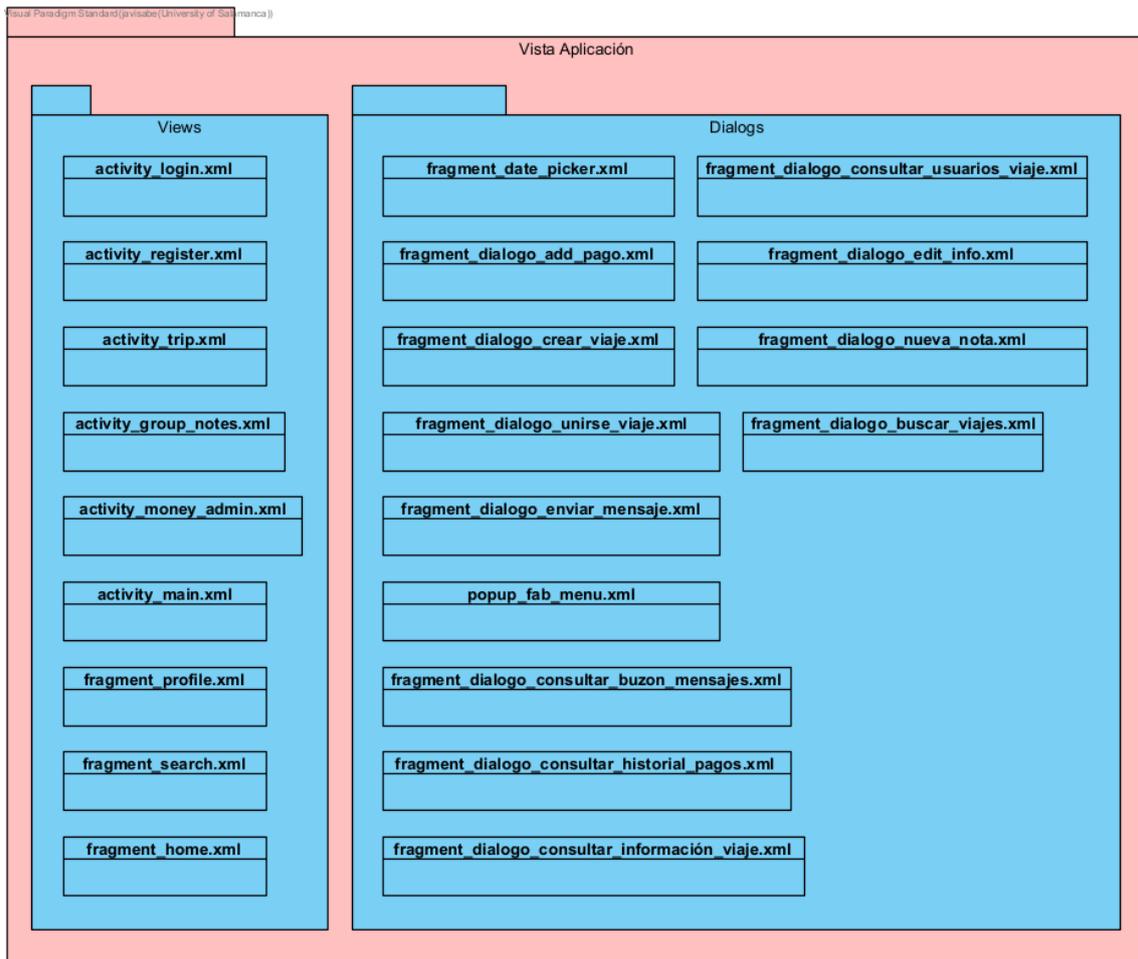


Figura 20. Clases de diseño 1. Vista aplicación

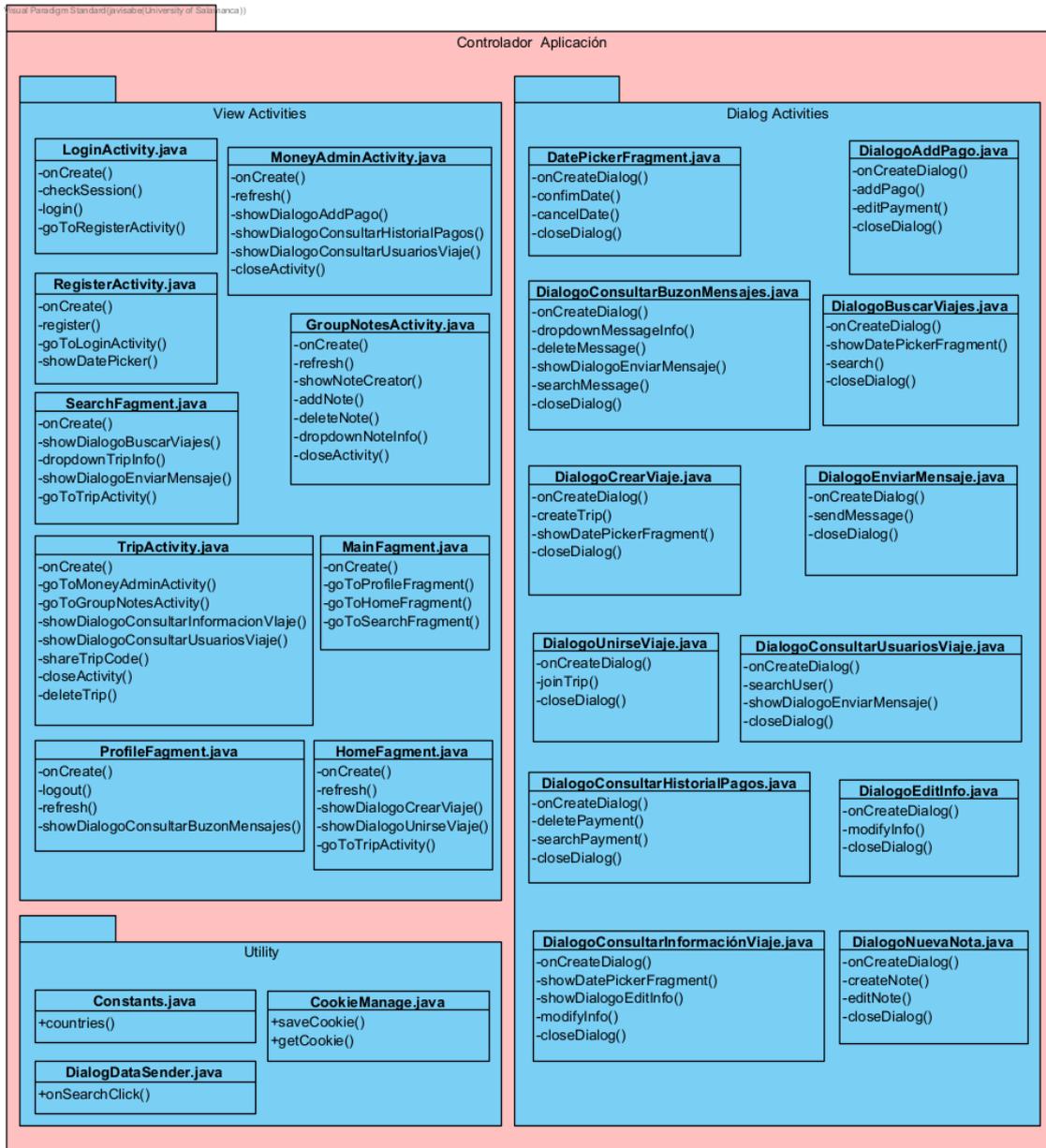


Figura 21. Clases de diseño 2. Controlador aplicación

La Figura 22 muestra el controlador del lado del servidor y sus métodos, ofrece las funcionalidades de acceso a la información, autenticación de usuarios y roles, y servicios REST para la aplicación.

La Figura 23 muestra el modelo del lado del servidor, con los modelos de datos y algunos métodos específicos necesarios.

# Memoria del proyecto

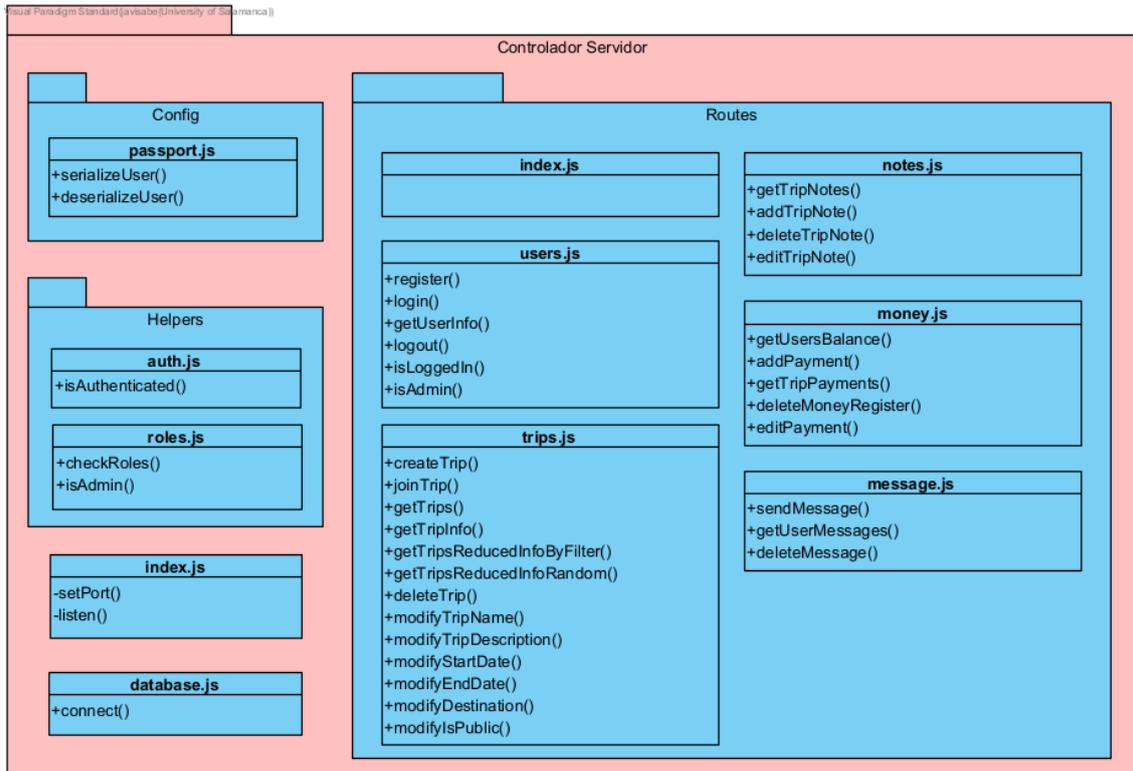


Figura 22. Clases de diseño 3. Controlador servidor

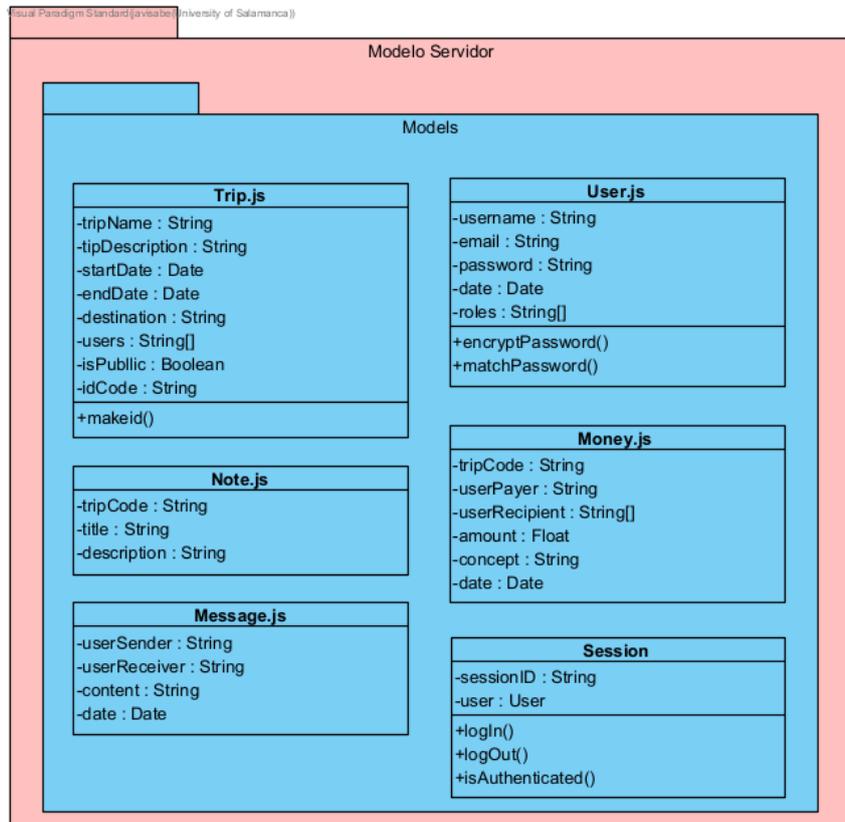


Figura 23. Clases de diseño 4. Modelo servidor

#### 6.6.4. Vista arquitectónica de diseño

En este apartado se detalla la arquitectura del sistema en base al patrón MVC, pero con la separación en paquetes del sistema y sus componentes. La Figura 24 muestra el diagrama representativo de esta vista de arquitectura del diseño.

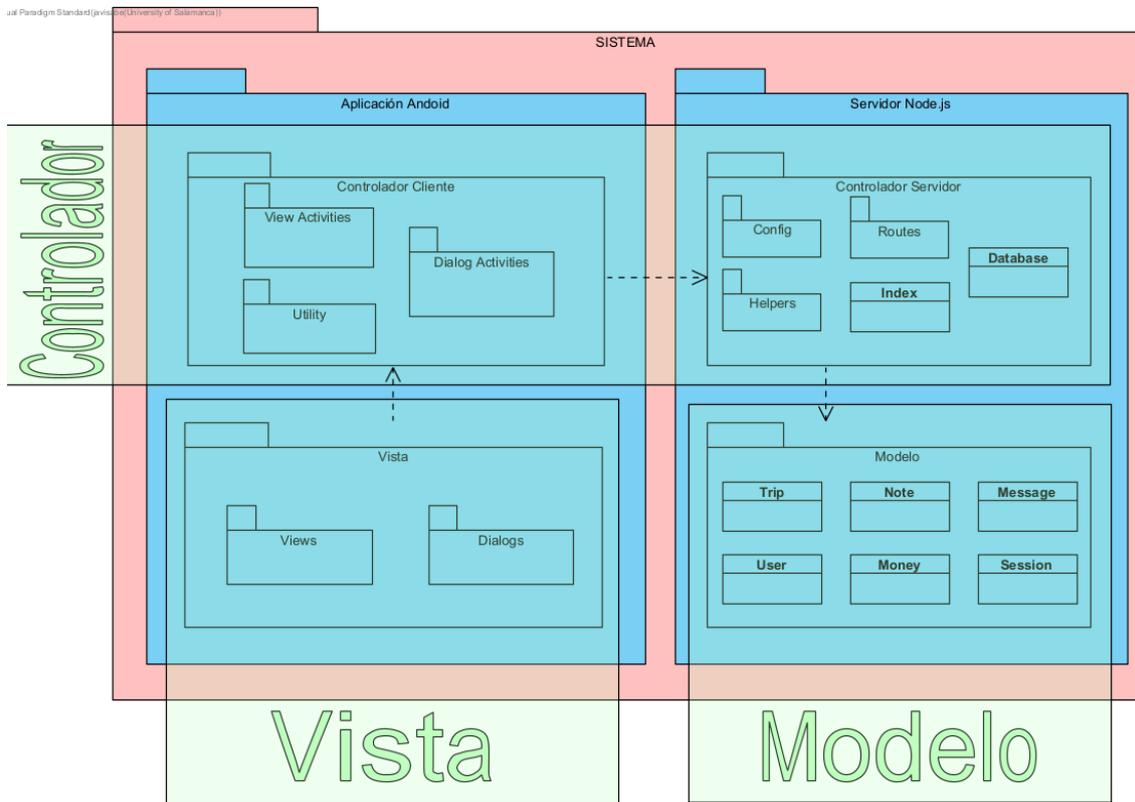


Figura 24. Vista arquitectónica de diseño

#### 6.6.5. Realización de casos de uso del diseño

Este apartado busca detallar los diagramas de secuencia obtenidos durante el análisis, utilizando las clases y componentes del diseño, sus interacciones y mensajes, buscando una aproximación lo más fiel posible a la implementación final. La Figura 25 muestra un ejemplo de estos diagramas de secuencia.

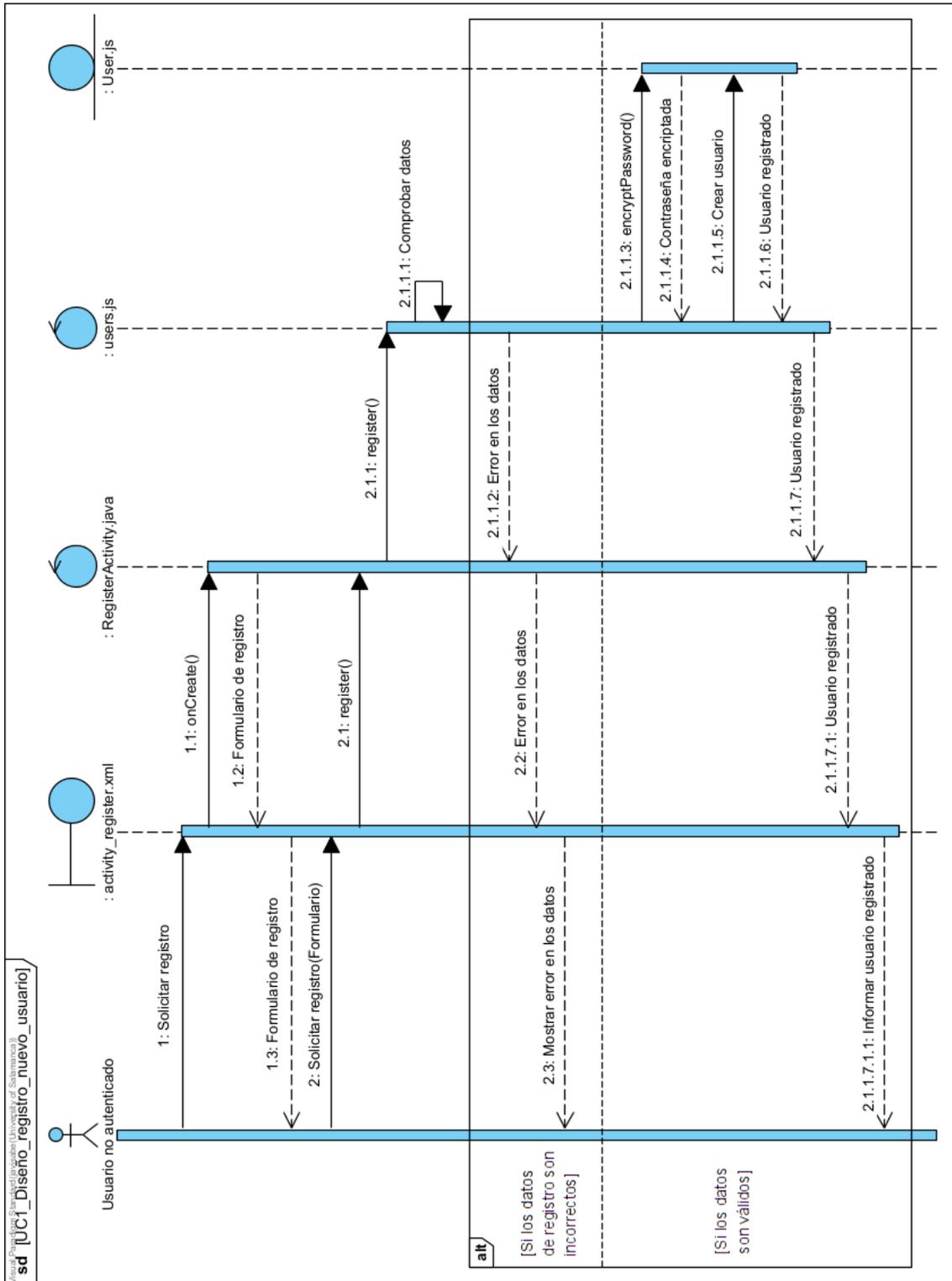


Figura 25. Diagrama de secuencia del diseño. Registro nuevo usuario

### 6.6.6. Diseño de la base de datos

En este apartado se muestra el diagrama del diseño de la base de datos, que representa la estructura final de esta. La implementación se realizará en MongoDB, que no utiliza el formato SQL estándar para el almacenamiento sino documentos en formato JSON. La Figura 26 muestra el diagrama representativo desarrollado para la base de datos.

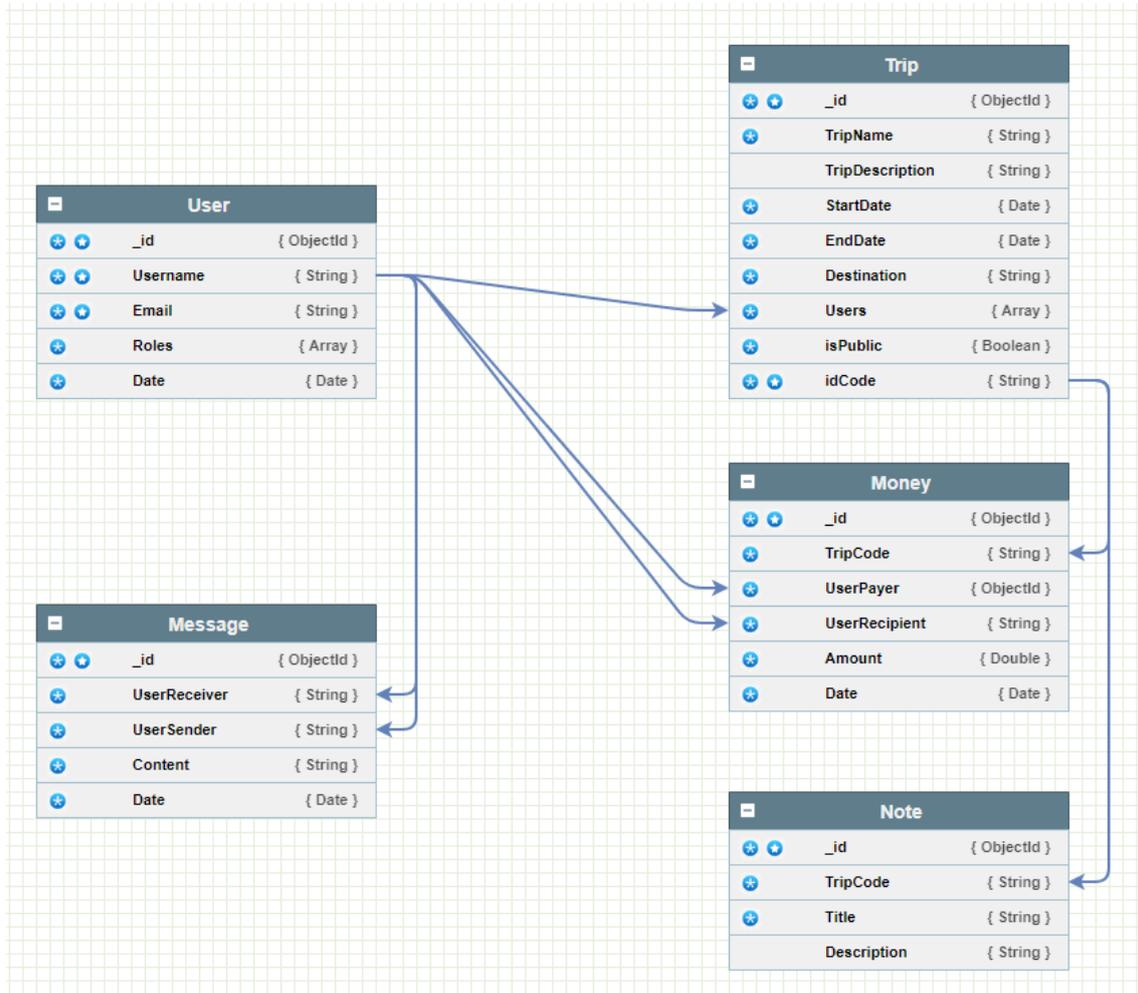


Figura 26. Diagrama de diseño de la base de datos

### 6.6.7. Modelo de despliegue

Este último apartado de la fase de diseño determina un diagrama de despliegue, que define la estructura de los nodos del sistema y sus relaciones, como muestra la Figura 27.

Los nodos que conforman el sistema son los siguientes:

- **Dispositivo móvil:** dispositivo que ejecuta la aplicación en un entorno de ejecución Android. Esta aplicación ofrece a los usuarios el método de interacción con el sistema y les permite utilizar todas sus funcionalidades.
- **Servidor REST:** servidor desplegado en un entorno de ejecución NodeJS es el

encargado de gestionar la información del sistema y proporcionar los métodos necesarios para que la aplicación pueda consultar y modificar la información pertinente en base a las acciones de los usuarios.

- **Base de datos:** servidor MongoDB donde se encontrará la base de datos del sistema y que almacenará toda la información de viajes y usuarios.

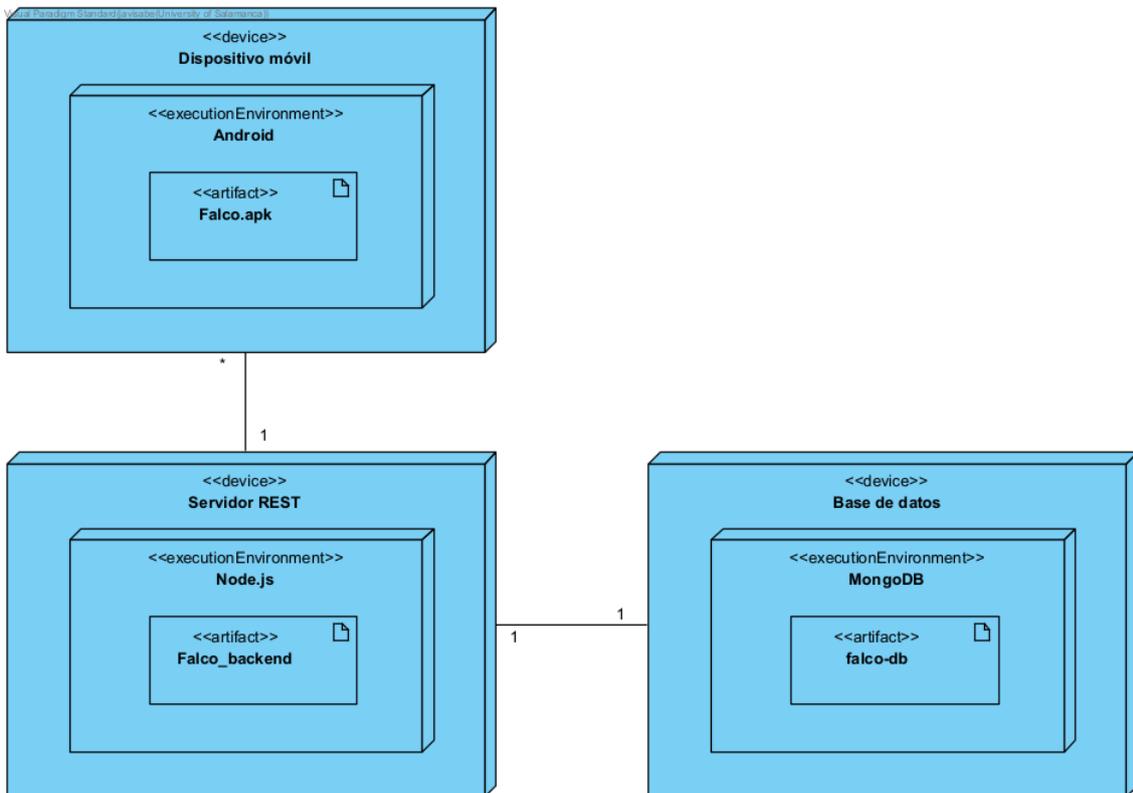


Figura 27. Diagrama de despliegue del sistema

## 6.7. Implementación

Durante esta fase del desarrollo se ha procedido a la programación del sistema, siguiendo los modelos obtenidos en la fase de diseño. Para la implementación se han utilizado las diferentes herramientas y técnicas definidas en el apartado 5 de esta memoria.

Como se ha expuesto anteriormente, el sistema consta de dos partes principales, las cuales han sido implementadas de forma simultánea.

### 6.7.1. Aplicación Android

La implementación de la aplicación comenzó por el diseño de las páginas de inicio de sesión y registro. Una vez implementado y funcional el apartado de la gestión de usuarios se procedió al diseño de los menús principales de la aplicación y su distribución dentro

## Memoria del proyecto

de la misma. Con el diseño claro, se implementó la creación de viajes y la unión a los mismos. Después se implementó toda la funcionalidad relativa a los viajes, es decir, las herramientas del gestor de dinero y notas grupales. La siguiente parte de la aplicación en ser implementada fue el buscador de viajes públicos, y, por último, el perfil del usuario y el buzón de mensajes.

Cabe destacar que prácticamente todas las pantallas y menús emergentes de la aplicación hacen llamadas a la API del servidor, ya sea para solicitar información, añadirla, eliminarla o modificarla, por lo que se han diseñado y desarrollado de forma conjunta para que funcionen de la mejor manera posible.

También se ha tenido en cuenta a la hora de desarrollar la aplicación las posibles futuras ampliaciones de funcionalidad que pudiese sufrir esta, facilitando la reutilización de código y la inclusión de nuevas opciones y herramientas en los menús.

La documentación técnica completa de la aplicación se encuentra en el *Anexo 5 – Documentación técnica*.

### 6.7.2. Servidor NodeJS

La implementación del servidor comenzó por la creación de la base de datos, para el almacenamiento de la información de los usuarios, y luego todo lo relativo a su registro y autenticación. A partir de ahí se fueron implementando los servicios de la API REST en el mismo orden que se implementaban las funcionalidades de la aplicación.

En la implementación del servidor también se ha seguido una política de futura escalabilidad, facilitando a nivel de código posibles ampliaciones de la API de forma sencilla. El servidor se ha desplegado en un host Windows 11 durante todo el desarrollo. La documentación técnica completa del servidor se encuentra en el *Anexo 5 – Documentación técnica*.

## 6.8. Pruebas

La realización de pruebas durante todas las fases del desarrollo del proyecto es un aspecto fundamental para evitar fallos en el sistema final y para asegurarse de que se cumplen todos los requisitos y objetivos establecidos.

Para las pruebas de la API del servidor se utilizó la herramienta Postman, sobre todo en los compases iniciales de la implementación, después se utilizó de forma mayoritaria la propia aplicación.

## Memoria del proyecto

Las pruebas de funcionamiento de la aplicación se hicieron de dos maneras simultáneas, mediante el emulador de Android incorporado por Android Studio, que utiliza un ‘Google Pixel 4’, y con un dispositivo físico ‘Realme x50 Pro’ con una configuración distinta, para encontrar y depurar la mayor cantidad de fallos posible.

Una vez implementadas la mayoría de las funcionalidades, se procedieron a hacer pruebas del sistema completo con usuarios reales, en concreto 4 adultos y 4 jóvenes en simultáneo, que probaron todas y cada una de las herramientas que ofrecía la aplicación y ayudaron a solucionar varios fallos que no se habían detectado anteriormente.

### 6.9. Funcionalidad del sistema

En este apartado se describen de manera resumida todas las funcionalidades que se pueden llevar a cabo desde la aplicación, y se va a dividir en distintas secciones en base a estas funcionalidades.

La guía completa de la aplicación se puede consultar en el *Anexo VI – Manual de usuario*.

#### 6.9.1. Registro, inicio y cierre de sesión

La aplicación cuenta con la posibilidad de registrar usuarios, que deberán tener una cuenta en el sistema para poder iniciar sesión y acceder a él. La Figura 28 muestra la pantalla de registro a la izquierda y la de inicio de sesión o *login* a la derecha.

Los usuarios también tienen la opción de consultar su información en el perfil y de cerrar sesión, como se muestra en la Figura 29.

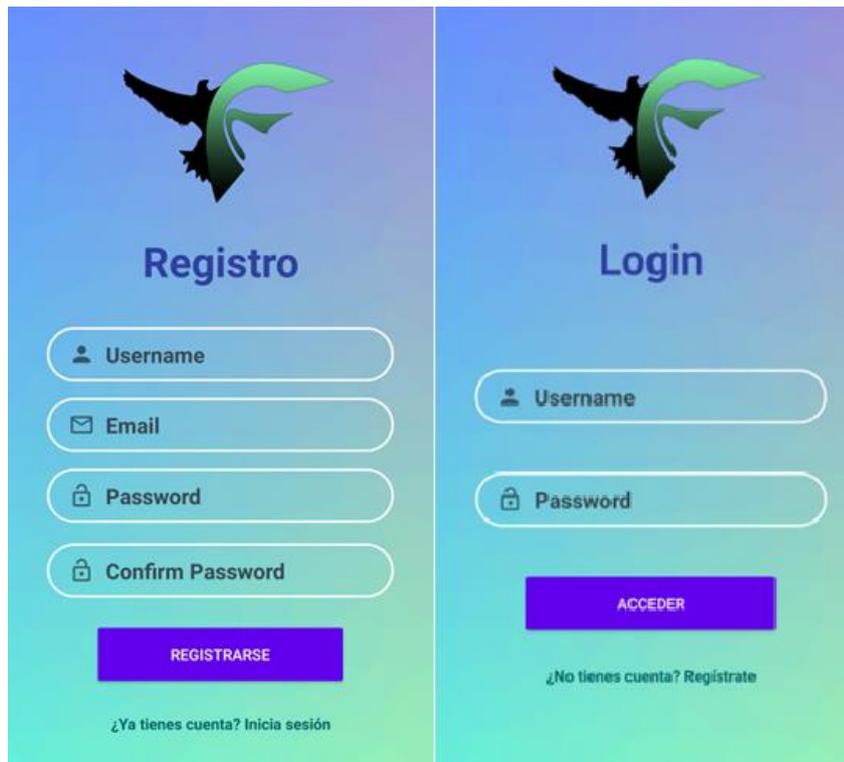


Figura 28. Pantallas de registro e inicio de sesión

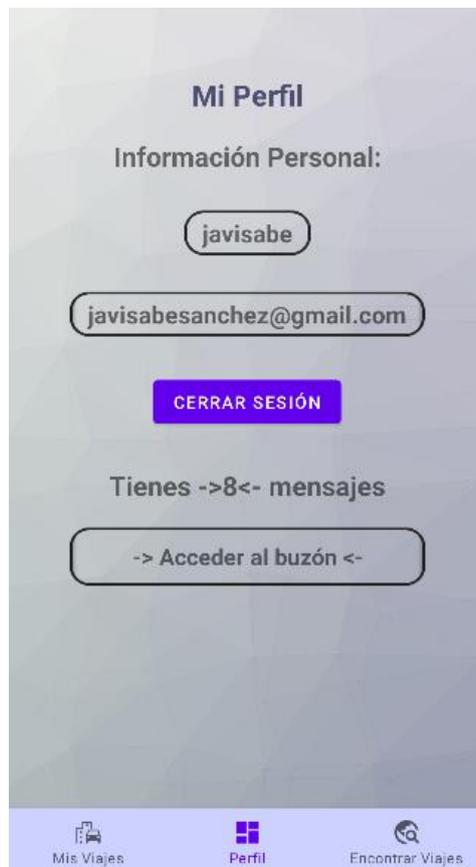


Figura 29. Información del usuario y cierre de sesión

### 6.9.2. Agenda de viajes

Los usuarios disponen de una agenda de viajes, donde aparecen todos los que han creado o a los que pertenecen, como se muestra en la Figura 30. La Figura 31 muestra a la izquierda el menú de creación de viajes, y a la derecha el menú de unirse a un viaje ya existente mediante su código. Estos dos menús son accedidos mediante el botón flotante ‘+’ de la parte inferior derecha de la agenda.

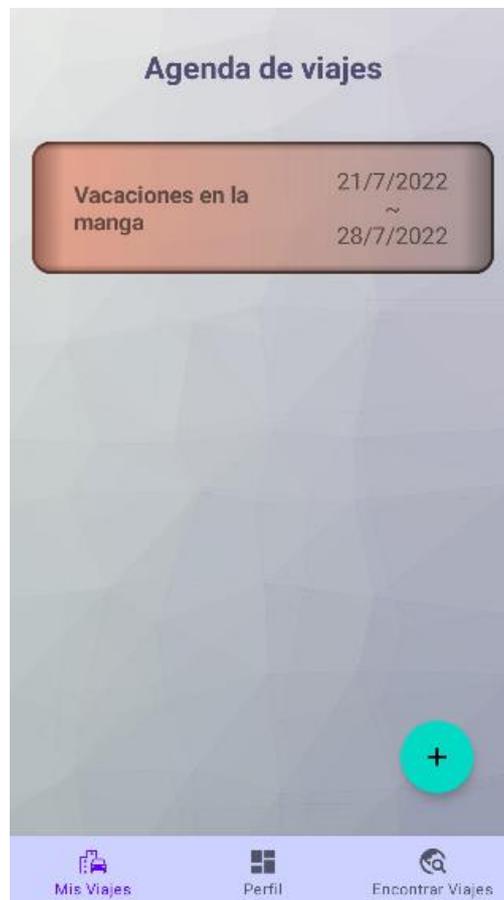


Figura 30. Agenda de viajes



Figura 31. Menús de creación y unión de viajes

### 6.9.3. Menú del viaje

Una vez accedido a uno de los viajes de la agenda, el usuario accede a su menú, con todos los accesos a sus funcionalidades como muestra la Figura 32.

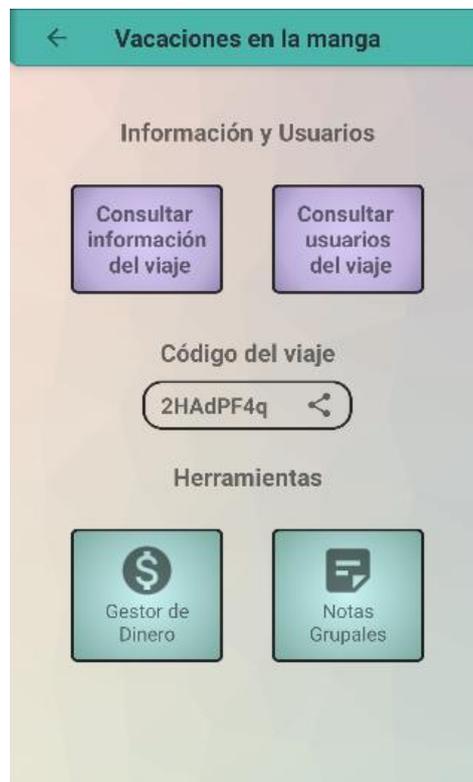


Figura 32. Menú del viaje

Los usuarios tienen la posibilidad de consultar y modificar toda la información relativa al viaje mediante el botón ‘Consultar información del viaje’ que muestra el cuadro de

diálogo de la Figura 33 y que aporta dicha funcionalidad.



Figura 33. Consultar información de viaje

A demás tienen la opción de ver todos los usuarios del viaje, y enviarles mensajes, mediante el botón 'Consultar usuarios del viaje', que despliega el cuadro de diálogo de la Figura 34. Este menú incorpora un buscador de usuarios en tiempo real.



Figura 34. Consultar usuarios del viaje

Desde el menú del viaje también podrás acceder a las funcionalidades de este; el Gestor de dinero (6.9.4) y las notas grupales (6.9.5).

#### 6.9.4. Gestor de dinero

El sistema permite a los usuarios del viaje llevar una cuenta de los gastos que se produzcan durante este, repartiendo así las deudas de manera equitativa. En el menú principal del gestor, que muestra en la Figura 35, se puede observar tú balance y los últimos pagos.

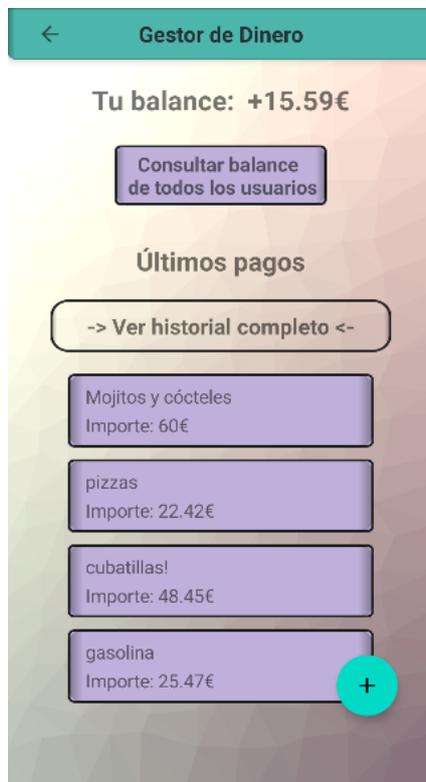


Figura 35. Gestor de gastos

El botón ‘Consultar balance de todos los usuarios’ permite ver las deudas de todos los miembros del viaje como muestra la Figura 36, e incorpora un buscador de usuarios en tiempo real.

El botón ‘Ver historial completo’ muestra un cuadro de diálogo con todos los pagos del viaje y su información completa como muestra la Figura 37. La Figura 38 muestra el menú de modificación de pagos, que es accedido pulsando cualquier pago del historial mencionado anteriormente, y manteniendo pulsado sobre estos pagos se permite su eliminación, como muestra la Figura 39. El historial cuenta con un buscador de pagos en tiempo real para agilizar su uso.

El botón ‘+’ de la parte inferior derecha permite añadir un nuevo pago como muestra el cuadro de diálogo de la Figura 40.

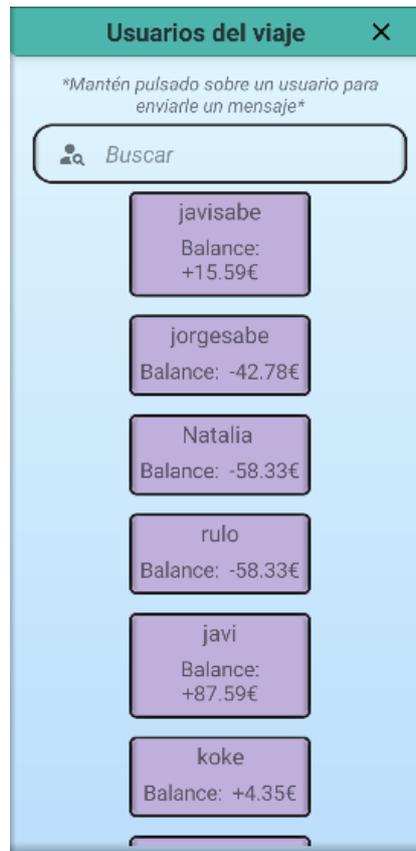


Figura 36. Balance de usuarios del viaje

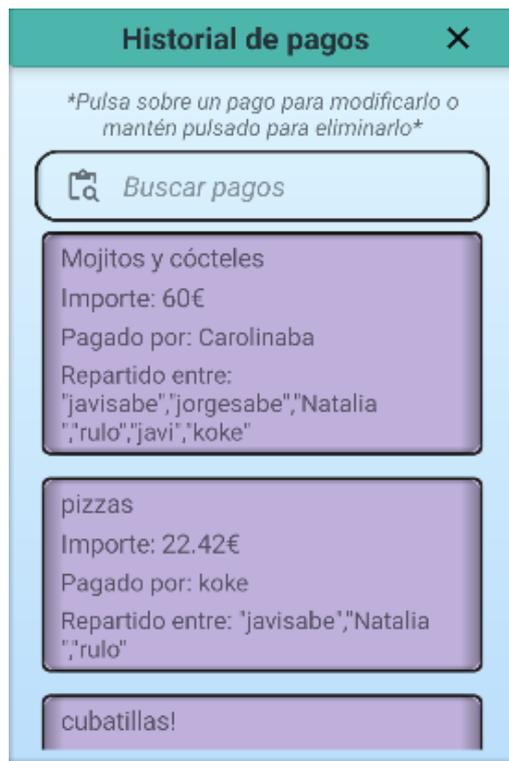


Figura 37. Historial de pagos

**Modificar pago** ✕

**Añade el concepto**

Mojitos y cócteles

**Importe**

60

**¿A repartir entre?**

javisabe

jorgesabe

Natalia

Figura 38. Modificación de pago

### Eliminar Pago

¿Está seguro de que desea eliminar el pago permanentemente?

**CANCELAR** **ELIMINAR**

Figura 39. Eliminación de pago

**Nuevo Pago** ✕

**Añade el concepto**

Concepto

**Importe**

€

**¿A repartir entre?**

javisabe

jorgesabe

Natalia

Figura 40. Adición de pago

### 6.9.5. Notas grupales

Los usuarios de un viaje tienen la capacidad de crear, modificar y eliminar notas de forma conjunta, en el menú que muestra la Figura 41.

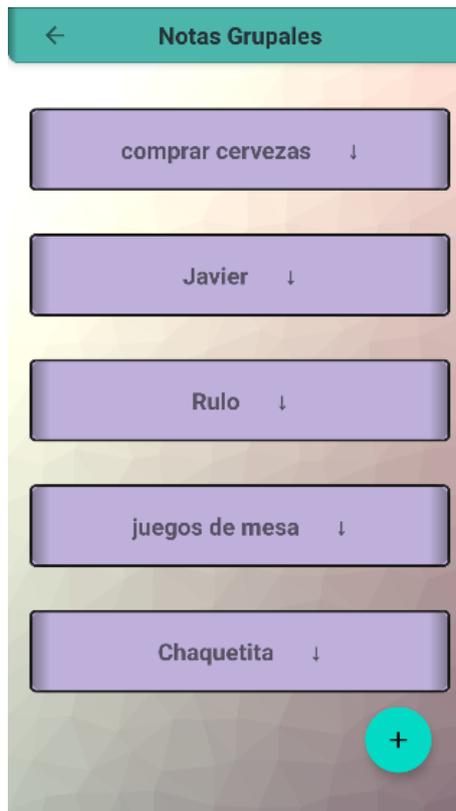


Figura 41. Notas grupales

El botón '+' de la parte inferior derecha, permite añadir una nueva nota, mostrando el menú emergente de la Figura 42.

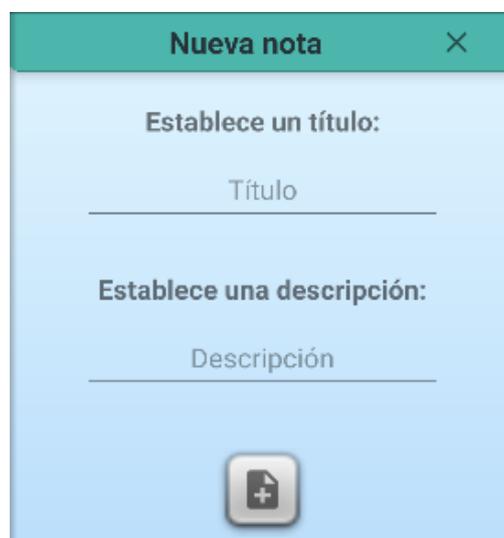


Figura 42. Añadir nota

## Memoria del proyecto

Las notas aparecen contraídas, desplegando su información (Figura 43) se muestran los botones para eliminar (similar a la Figura 39) y modificar (Figura 44) las notas.

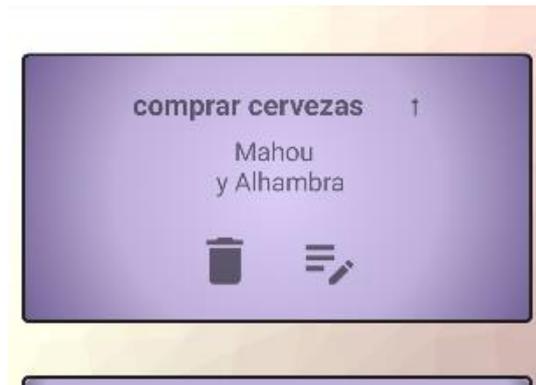


Figura 43. Desplegar nota

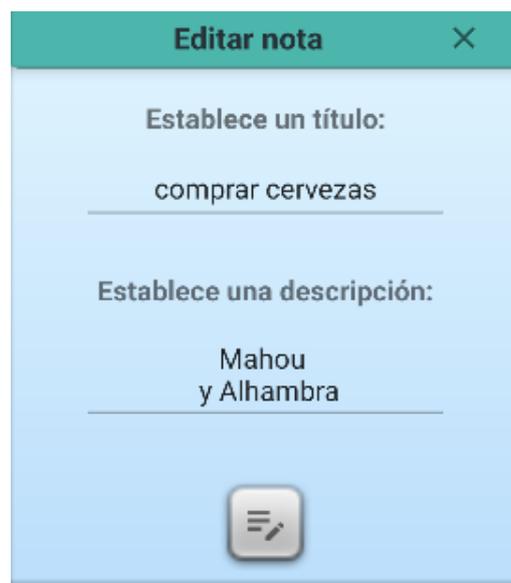


Figura 44. Editar nota

### 6.9.6. Mensajes entre usuarios

Los usuarios tienen la opción de enviar mensajes a otros, ya sea a los miembros de su viaje, a los creadores de viajes públicos encontrados en el buscador, o en forma de respuesta a mensajes recibidos. El menú de envío de mensajes es el mostrado en la Figura 45.

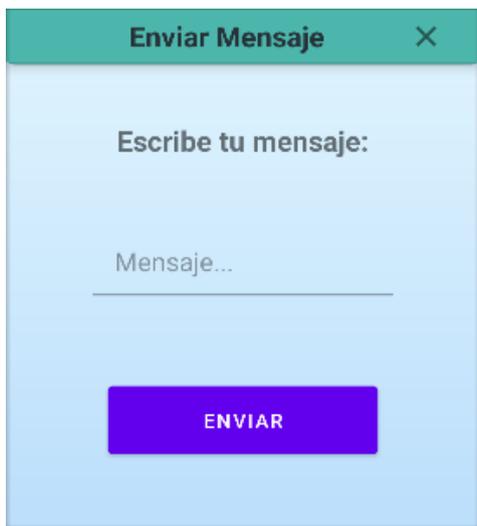


Figura 45. Enviar mensaje

Los usuarios pueden consultar su buzón de mensajes recibidos mediante el botón ‘Acceder al buzón’ (situado en su perfil, Figura 29). Dentro de este menú aparecerán todos los mensajes recibidos, como muestra la Figura 46, que podrán ser filtrados mediante el buscador en tiempo real que incorpora.

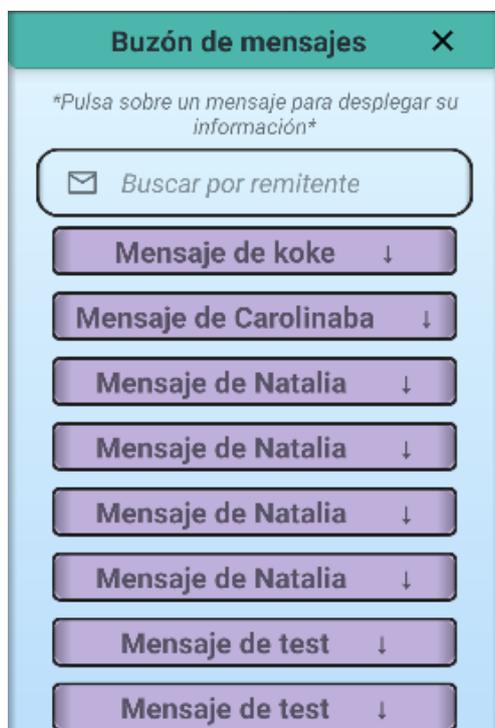


Figura 46. Buzón de mensajes

Tras desplegar la información del mensaje (Figura 47), se mostrarán las opciones de eliminar (similar a la Figura 39) y responder mensaje.



Figura 47. Mensaje desplegado

### 6.9.7. Buscador de viajes

Existe la opción de buscar viajes creados y publicados por otros usuarios mediante el buscador de la Figura 48. Este buscador mostrará unos viajes aleatorios de forma predeterminada.



Figura 48. Buscador de viajes

Desplegando estos viajes, se mostrará su información completa y la opción de enviar un mensaje al creador, como se ve en la Figura 49.

Para realizar una búsqueda por filtros hay que pulsar el botón 'Buscar' de la parte superior, que desplegará el cuadro de diálogo de la Figura 50 donde se especifican estos parámetros de búsqueda.

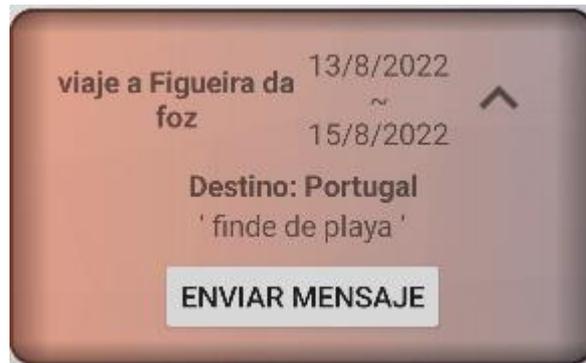


Figura 49. Información de viaje público



Figura 50. Búsqueda de viajes por filtros

#### 6.9.8. Opciones del administrador

Los administradores del sistema disponen de algunas opciones únicas específicas dentro de la aplicación.

En la búsqueda de viajes, tienen la opción de acceder a la información completa de estos como muestra la Figura 51, pudiendo así acceder al menú del viaje y a toda la información de usuarios, gastos, notas, etc.

También tienen la opción de eliminar viajes de otros usuarios desde el menú del viaje, como muestra la Figura 52.

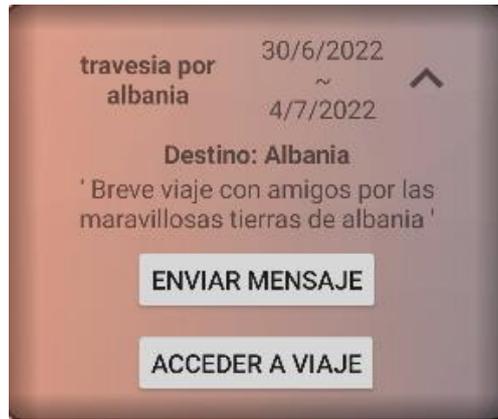


Figura 51. Acceso a viaje – Administrador

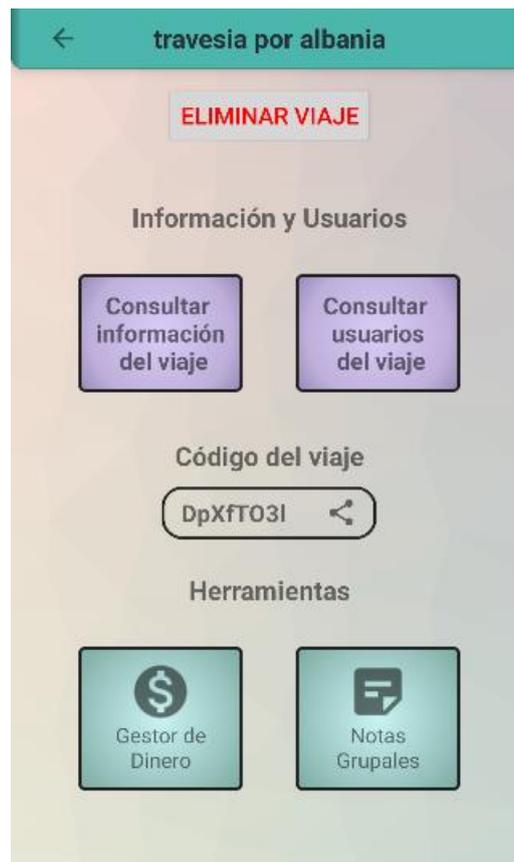


Figura 52. Eliminar viaje - Administrador

## 7. Limitaciones de la aplicación

Aunque la aplicación es completamente funcional, tiene ciertas limitaciones, sobre todo enfocada a un amplio público internacional.

- La aplicación ha sido desarrollada de forma exclusiva para dispositivos Android, por lo cual, las personas que dispongan de dispositivos con IOS no tienen la posibilidad de utilizarla. Esto puede dejar fuera a algunos usuarios que formen parte de un viaje, y la aplicación no cumpliría con su funcionalidad satisfactoriamente.
- La aplicación está enfocada a un público internacional, por lo que necesitaría estar traducida a muchos otros idiomas para su correcta integración en el mercado.
- El sistema no puede asegurar que los viajes o personas que se encuentren en el buscador sean reales y lícitos, aunque existen los usuarios con rol de administrador en el sistema que pueden eliminar estos viajes, muchos son indetectables.

## 8. Conclusiones y líneas de trabajo futuras

En este apartado se recogen las conclusiones personales alcanzadas tras finalizar el desarrollo del proyecto y las posibles implementaciones o mejoras para la aplicación en las que se podría trabajar en un futuro.

### 8.1. Conclusiones

Una vez concluido el desarrollo del proyecto y habiendo cumplido todos los objetivos propuestos al comienzo, se ha llegado a las siguientes conclusiones:

- Se ha desarrollado de forma satisfactoria el trabajo *Desarrollo de aplicación móvil de gestión de viajes en grupo*, consiguiendo que sea completamente funcional y cumpliendo todos los objetivos propuestos para el sistema. El sistema es capaz de gestionar la creación y búsqueda de viajes, así como la interacción simultánea de todos los usuarios con las herramientas internas de dichos viajes. Permite añadir, modificar y eliminar gastos y notas, repartir deudas de forma equitativa, y enviar mensajes entre usuarios.
- Se ha desarrollado un servidor como gestor de la información del sistema, y que implementa una API REST propia para el funcionamiento global de la aplicación, aunando conocimientos de programación, redes y sistemas.
- Para el desarrollo de la aplicación Android se han empleado conocimientos de programación y de diseño de interfaces, concluyendo en una aplicación que ofrece una experiencia de usuario satisfactoria.
- Ha sido la primera vez que se ha llevado a cabo un proyecto de tal envergadura y de forma solitaria, por lo que se ha experimentado el esfuerzo que conlleva, y, sobre todo, la importancia que tienen todas y cada una de las fases del desarrollo, en especial la planificación temporal y el diseño del sistema.
- Se ha visto la importancia de tener conocimientos en cuanto a los lenguajes de programación, los patrones de diseño, y otros ámbitos de la programación y de la ingeniería del software de forma previa a comenzar la implementación, ya que ahorra tiempo y evita una gran cantidad de errores, teniendo como consecuencia un producto final de mayor calidad.
- Como aprendizaje y conclusión negativa, debido a la inexperiencia en este tipo de proyectos y dificultades con algunas herramientas en las que se era novato, se han experimentado dificultades a la hora de seguir el calendario establecido para el

desarrollo, lo que ha impedido su entrega en la convocatoria ordinaria.

## 8.2. Líneas de trabajo futuras

Tras el desarrollo de la aplicación y las pruebas con usuarios, se han observado futuras implementaciones que la mejorarían, como son las siguientes:

- Implementar más herramientas para los viajes, hay infinidad de posibilidades, pero, por ejemplo, se podría añadir una herramienta de reparto equitativo de tareas, o una herramienta de calendario conjunto para todos los miembros del viaje, u ofrecer la posibilidad de subir imágenes a los viajes para hacer un álbum conjunto de fotografías de todos los miembros.
- Hacer integraciones con las API's de TripAdvisor o Google Maps para buscar lugares o comercios y añadirlos como marcadores en tus viajes, pudiendo así crear un cuaderno de bitácora.
- Se podrían establecer herramientas o funcionalidades de pago en caso de que la aplicación saliese al mercado de forma oficial, para obtener así un rendimiento económico.
- Traducir la aplicación a múltiples idiomas para que pueda ser utilizada por usuarios de todo el mundo.
- Desarrollar tanto una versión para IOS como una versión web de la aplicación, para que todos los posibles miembros de un viaje puedan tener acceso.
- Implementar un sistema de verificación de usuarios mediante documento oficial de identidad, para evitar perfiles falsos e intenciones maliciosas.

## 9. Referencias

- Adobe. (2018). *Photoshop*. Obtenido de Adobe: <https://www.adobe.com/es/products/photoshop.html>
- Aguilar, J. M. (2022). *Patrón MVC*. Obtenido de CampusMVP: <https://www.campusmvp.es/recursos/post/que-es-el-patron-mvc-en-programacion-y-por-que-es-util.aspx>
- Automatic. (9 de febrero de 2022). *Mongoose js*. Obtenido de <https://mongoosejs.com/>
- Background Generator. (2022). *Bgenerator*. Obtenido de <https://bggenerator.com/>
- Bartolomeo, N., Girish, S., & Wirtz, D. (17 de febrero de 2022). *Bcrypt.js*. Obtenido de Github: <https://github.com/dcodeIO/bcrypt.js>
- BlaBlaCar*. (2022). Obtenido de <https://www.blablacar.es/>
- Booch, G., Rumbaugh, J., & Jacobson, I. (2006). *El lenguaje unificado de modelado. Guía del usuario*. Madrid: Pearson Educación.
- Concur Technologies, Inc. (2022). *TripIt*. Obtenido de <https://www.tripit.com/web>
- Durán Toro, A. (10 de abril de 2022). *REM*. Obtenido de Departamento de lenguajes y sistemas informáticos - Universidad de Sevilla: [http://www.lsi.us.es/descargas/descarga\\_programas.php?id=3](http://www.lsi.us.es/descargas/descarga_programas.php?id=3)
- Durán Toro, A., & Bernárdez Jiménez, B. (2002). *Metodología para la Elicitación de Requisitos de Sistemas Software*. Sevilla: Departamento de Lenguajes y Sistemas Informáticos, Escuela Técnica Superior de Ingeniería Informática.
- ECMA International. (2017). *Standard ECMA-404. The JSON Data Interchange Syntax*. Geneva: ECMA International.
- Google. (2022). *Google Maps API*. Obtenido de <https://developers.google.com/maps>
- Google. (2022). *Keep*. Obtenido de <https://keep.google.com/>
- Google. (8 de febrero de 2022). *Volley*. Obtenido de Github: <https://github.com/google/volley>
- Google Developers. (7 de febrero de 2022). *Android studio*. Obtenido de <https://developer.android.com/studio/intro>
- Hanson, J. (9 de febrero de 2022). *Passport js*. Obtenido de <https://www.passportjs.org/>
- Herramienta CASE*. (2022). Obtenido de GlosarioIT: [https://www.glosarioit.com/Herramienta\\_CASE](https://www.glosarioit.com/Herramienta_CASE)
- JetBrains. (23 de junio de 2022). *IntelliJ IDEA*. Obtenido de JetBrains: <https://www.jetbrains.com/idea/>

## Memoria del proyecto

- Joinmytrip*. (2022). Obtenido de <https://www.joinmytrip.com/en/>
- Maldeadora, N. (2022). *Qué es Frontend y Backend: diferencias y características*. Obtenido de Platzi: <https://platzi.com/blog/que-es-frontend-y-backend/>
- Microsoft. (20 de marzo de 2022). *Microsoft Project*. Obtenido de Microsoft: <https://www.microsoft.com/es-es/microsoft-365/project/project-management-software>
- Microsoft. (8 de febrero de 2022). *Visual studio code*. Obtenido de <https://code.visualstudio.com/>
- MongoDB, Inc. (9 de febrero de 2022). *MongoDB*. Obtenido de <https://www.mongodb.com/es>
- Mozilla Foundation. (2 de febrero de 2022). *JavaScript*. Obtenido de Mozilla developers: <https://developer.mozilla.org/es/docs/Web/JavaScript>
- Mozilla Foundation. (3 de febrero de 2022). *XML: Extensible Markup Lenguaje*. Obtenido de Mozilla developers: [https://developer.mozilla.org/es/docs/Web/XML/XML\\_introduction](https://developer.mozilla.org/es/docs/Web/XML/XML_introduction)
- NoSQL DB Schema Modeling*. (2022). Obtenido de <https://nosqldbm.ru/>
- NPM, Inc. (11 de febrero de 2022). *npmjs*. Obtenido de <https://www.npmjs.com/>
- Open Source project. (17 de junio de 2022). *JSDoc*. Obtenido de <https://jsdoc.app/>
- Oracle. (3 de febrero de 2022). *Java*. Obtenido de <https://www.java.com/es/>
- Oracle. (17 de junio de 2022). *Javadoc*. Obtenido de Oracle docs website: <https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html>
- Postman, Inc. (2022). *Postman*. Obtenido de <https://www.postman.com/>
- Proceso Unificado de Desarrollo de Software*. (2022). Obtenido de Blogspot: <http://pnfiingenieriadesoftwaregrupocuatro.blogspot.com/2012/07/proceso-unificado-proceso-unificado-de.html>
- Purushothaman, B. (25 de febrero de 2009). *EZEstimate*. Obtenido de Google sites: <https://sites.google.com/site/ezeestimate/>
- Sellarès, T. (26 de abril de 2022). *MVC*. Obtenido de Portal de educación de la Universidad de Girona: <https://ima.udg.edu/~sellares/EINF-ES1/MVC-Toni.pdf>
- SIMULATION MODEL FOR RATIONAL UNIFIED PROCESS (RUP) SOFTWARE DEVELOPMENT LIFE CYCLE - Scientific Figure*. (2022). Obtenido de ResearchGate: [https://www.researchgate.net/figure/Development-Phase-of-Rational-Unified-Process-RUP-10\\_fig1\\_331054091](https://www.researchgate.net/figure/Development-Phase-of-Rational-Unified-Process-RUP-10_fig1_331054091)

## Memoria del proyecto

The OpenJS Foundation. (9 de febrero de 2022). *Express Js*. Obtenido de <http://expressjs.com/es/>

The OpenJS Foundation. (9 de febrero de 2022). *Node JS*. Obtenido de <https://nodejs.org/es/>

*Tricount*. (2022). Obtenido de <https://www.tricount.com/es/>

TripAdvisor. (2022). *TripAdvisor API*. Obtenido de <https://www.tripadvisor.com/developers>

Visual Paradigm. (2022). *Visual Paradigm*. Obtenido de <https://www.visual-paradigm.com/>

vmarchaud. (18 de junio de 2022). *jsdoc-http-plugin*. Obtenido de Github: <https://github.com/vmarchaud/jsdoc-http-plugin>

Wozniewicz, B. (2022). *The Difference Between a Framework and a Library*. Obtenido de FreeCodeCamp: <https://www.freecodecamp.org/news/the-difference-between-a-framework-and-a-library-bd133054023f/>

Zepeda, E. (2022). *Características básicas de una API REST y recomendaciones*. Obtenido de coffeebytes: <https://coffeebytes.dev/caracteristicas-basicas-de-una-api-rest/>