

DESARROLLO DE UN SISTEMA PARA FACILITAR LAS VISITAS AL PARQUE DE EL CAPRICHIO DE MADRID



TRABAJO FIN DE GRADO
CURSO 2022-2023

AUTOR
JORGE SAN FRUTOS IGLESIAS

DIRECTOR
ANTONIO SARASA CABEZUELO

GRADO EN INGENIERÍA DE COMPUTADORES
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

DEVELOPMENT OF A SYSTEM TO FACILITATE VISITS TO THE CAPRICE PARK IN MADRID

FINAL DEGREE PROJECT REPORT IN COMPUTER
ENGINEERING

AUTHOR
JORGE SAN FRUTOS IGLESIAS

DIRECTED BY
ANTONIO SARASA CABEZUELO

FACULTY OF COMPUTER SCIENCE
COMPLUTENSE UNIVERSITY OF MADRID

JUNE 2023

DEDICATORIA

A todos los familiares y amigos que me han apoyado durante estos años de carrera.

AGRADECIMIENTOS

Lo primero de todo agradecer a mi tutor Antonio Sarasa Cabezuelo por orientarme durante la realización del trabajo, ayudarme dándome consejos y estar disponible en todo momento para las dudas que me iban surgiendo.

También agradecer a todos los familiares y amigos que han estado apoyándome y ayudándome durante el trabajo y a lo largo de toda la carrera.

RESUMEN

El objetivo de este trabajo es facilitar las visitas al parque del Capricho de Madrid a través de una aplicación móvil, en la que se pueda consultar información sobre los distintos lugares e itinerarios del parque.

El usuario tendrá acceso a la lista completa de lugares e itinerarios disponibles en el parque, con información detallada, así como la posibilidad de ver su ubicación actual e iniciar una ruta al lugar de interés o al itinerario elegido.

Además, se ofrece al usuario registrarse en la aplicación para obtener una mejor experiencia, pudiendo agregar comentarios a los lugares y guardarse los itinerarios que más le gusten.

Se ha desarrollado una aplicación móvil para dispositivos Android, en el entorno de Android Studio, que implementa la información que se va guardando en la base de datos a través de un hosting.

Palabras clave

Parque del Capricho, aplicación móvil, lugar, itinerario, mapa.

ABSTRACT

The objective of this work is to facilitate visits to the Caprice Park in Madrid through a mobile application, where you can consult information about the different places and itineraries of the park.

The user will have access to the entire list of places and itineraries that are available in the park with detailed information, as well as the ability to view your current location and start a route to the place of interest or itinerary you choose.

In addition, it offers the possibility of registering in the application to obtain a better experience, being able to add comments to the places and keep the itineraries that you like the most.

A mobile application for Android devices has been developed in Android Studio that implements the information that is saved in the database through a hosting.

Keywords

Caprice Park, Mobile application, place, itinerary, map.

ÍNDICE DE CONTENIDOS

Dedicatoria	III
Agradecimientos.....	V
Resumen	VII
Abstract	IX
Índice de contenidos.....	X
Índice de figuras.....	XV
Índice de tablas.....	XIX
Capítulo 1 - Introducción	21
1.1 Motivación	21
1.2 Objetivos.....	21
1.3 Plan de trabajo.....	22
1.3.1 Fases del proyecto	22
1.3.2 Organización del trabajo.....	23
1.4 Estructura de la memoria	24
Introduction	25
Motivation.....	25
Objectives	25
Work plan.....	26
Project phases.....	26
Work organization	27
Structure of the report.....	28
Capítulo 2 - Estado del arte	29

2.1 Zoo Aquarium Madrid.....	29
2.2 Parques Nacionales.....	29
2.3 Parques y jardines de Madrid.....	29
Capítulo 3 - Tecnología empleada	30
3.1 Android Studio	30
3.2 Visual Studio Code.....	30
3.3 Xampp	30
3.4 Picasso	30
3.5 X10hosting	31
3.6 PhpMyAdmin	31
3.7 Google Drive.....	31
3.8 Mapbox	31
3.9 Volley.....	31
Capítulo 4 - Especificación de requisitos.....	32
4.1 Actores.....	32
4.2 Diagramas de casos de uso	32
4.3 Especificación de requisitos	34
4.3.1 Casos de uso de Gestión de usuarios	36
4.3.2 Casos de uso de información del parque.....	41
4.3.3 Casos de uso de gestión de itinerarios	47
4.3.4 Casos de uso del administrador.....	50
Capítulo 5 - Arquitectura de la aplicación y modelo de datos	57
5.1 Arquitectura de la aplicación.....	57
5.2 Modelo de la base de datos.....	58
5.2.1 Usuarios	59

5.2.2 Lugares.....	59
5.2.3 Comentarios.....	59
5.2.4 Itinerario	60
5.2.5 Itinerarios favoritos.....	60
Capítulo 6 - Implementación y diseño de la aplicación.....	61
6.1 Módulo usuario	62
6.1.1 Registro	62
6.1.2 Login	63
6.1.3 Logout	65
6.1.4 Modificar datos	65
6.1.5 Eliminar cuenta	67
6.2 Módulo de información del parque.....	68
6.2.1 Mostrar plano del parque	68
6.2.2 Mostrar ubicación en plano del parque	69
6.2.3 Mostrar lugares de interés	70
6.2.4 Buscar lugar de interés	71
6.2.5 Agregar comentario a lugar de interés.....	72
6.2.6 Modificar comentario a lugar de interés.....	73
6.2.7 Eliminar comentario de lugar de interés.....	74
6.2.8 Iniciar ruta a lugar de interés.....	75
6.3 Módulo de itinerarios	76
6.3.1 Ver itinerarios.....	76
6.3.2 Iniciar itinerario.....	78
6.3.3 Añadir itinerario favorito	79
6.3.4 Eliminar itinerario favorito	81

6.3.5 Ver itinerarios favoritos.....	82
6.4 Administrador.....	84
6.4.1 Añadir itinerario	84
6.4.2 Eliminar itinerario.....	85
6.4.3 Eliminar usuario	87
6.4.4 Eliminar comentario de usuario.....	88
6.4.5 Añadir lugar de interés	89
6.4.6 Modificar lugar de interés	91
6.4.7 Eliminar lugar de interés.....	92
Capítulo 7 - Conclusiones y trabajo futuro	93
7.1 Conclusiones.....	93
7.2 Trabajo futuro.....	93
Conclusions and future work	95
Conclusions	95
Future work	95
Bibliografía.....	97
Anexo A: Guía de Uso	99
A.1 Eliminar Usuario	99
A.2 Eliminar comentario de un usuario	100
A.3 Gestión de lugares	101
A.4 Gestión de itinerarios.....	102
A.5 Registro y Login	103
A.6 Logout, modificación datos y eliminar usuario	104
A.7 Gestión de comentarios.....	105
A.8 Mapa.....	106

A.9 Iniciar ruta a lugar de interés	107
A.10 Lugares de interés	108
A.11 Ver itinerarios y añadir a favoritos.....	109
A.12 Iniciar itinerario	109
A.13 Itinerarios favoritos.....	110

ÍNDICE DE FIGURAS

Figura 1:Diagrama de casos de uso módulo usuario	32
Figura 2: Diagrama de casos de uso módulo información del parque.....	33
Figura 3: Diagrama de casos de uso módulo itinerarios	33
Figura 4: Diagrama de casos de uso módulo administrador	34
Figura 5: Arquitectura de la aplicación	57
Figura 6: Modelo E.R base de datos.....	58
Figura 7: Menú lateral	61
Figura 8: Pantalla de inicio.....	61
Figura 9: Pantalla de registro de un usuario	62
Figura 10: Pantalla de registro con campos no válidos	62
Figura 11: Fragmento de código del registro de un usuario.....	63
Figura 12: Pantalla de login	64
Figura 13: Pantalla de login con mensaje de error	64
Figura 14: Fragmento de código del login	64
Figura 15: Fragmento de código del logout	65
Figura 16: Pantalla de modificar datos del usuario.....	66
Figura 17: Fragmento de código para modificar datos del usuario	66
Figura 18: Fragmento de código para eliminar un usuario.....	67
Figura 19: Pantalla del mapa	68
Figura 20: Pantalla del mapa mostrando lugar de interés.....	68
Figura 21: Fragmento de código para mostrar los lugares en el mapa	69
Figura 22: Pantalla del mapa con la ubicación actual del usuario	69
Figura 23: Fragmento de código para mostrar la ubicación del usuario.....	70
Figura 24: Pantalla para mostrar los lugares de interés	70
Figura 25: Fragmento de código para mostrar los lugares	71
Figura 26: Buscar lugar.....	71
Figura 27: Fragmento de código para buscar los lugares	72
Figura 28: Fragmento de código para agregar comentarios.....	72
Figura 29: Pantalla para añadir, modificar y eliminar comentarios.....	73
Figura 30: Fragmento de código para modificar comentarios	74

Figura 31: Fragmento de código para eliminar comentarios	74
Figura 32: Pantalla iniciar ruta	75
Figura 33: Pantalla mostrar ruta paso a paso	75
Figura 34: Fragmento de código para iniciar una ruta a un lugar	76
Figura 35: Pantalla para visualizar itinerarios	77
Figura 36: Pantalla para ver los lugares del itinerario.....	77
Figura 37: Fragmento de código para mostrar los itinerarios	77
Figura 38: Iniciar itinerario	78
Figura 39: Ruta paso a paso del itinerario	78
Figura 40: Fragmento de código para iniciar la ruta de un itinerario	79
Figura 41: Pantalla para añadir un itinerario a favoritos.....	80
Figura 42: Fragmento de código para añadir un itinerario a favoritos.....	80
Figura 43: Pantalla para iniciar o eliminar un itinerario de favoritos	81
Figura 44: Fragmento de código para eliminar un itinerario	82
Figura 45: Pantalla itinerario favoritos sin hacer login	83
Figura 46: Pantalla ver itinerario favoritos	83
Figura 47: Fragmento de código para mostrar los itinerarios favoritos	83
Figura 48: Pantalla itinerarios administrador	84
Figura 49: Pantalla para añadir lugares al itinerario	84
Figura 50: Fragmento de código para añadir un itinerario	85
Figura 51: Pantalla para acceder a añadir o eliminar un itinerario.....	86
Figura 52: Fragmento de código para eliminar un itinerario	86
Figura 53: Pantalla para que el administrador elimine a un usuario	87
Figura 54: Fragmento de código para que el administrador elimine un usuario	88
Figura 55: Pantalla para que el administrador elimine comentarios.....	89
Figura 56: Pantalla para añadir, modificar o eliminar un lugar	90
Figura 57: Fragmento de código para añadir un lugar	90
Figura 58: Pantalla para acceder a añadir, modificar o eliminar un lugar.....	91
Figura 59: Fragmento de código para modificar un lugar	92
Figura 60: Fragmento de código para eliminar un lugar	92
Figura 61: Administrador eliminar usuario	99
Figura 62: Administrador eliminar comentario	100

Figura 63: Mostrar lugares.....	101
Figura 64: Gestionar lugares.....	101
Figura 65: Mostrar itinerarios.....	102
Figura 66: Gestionar itinerarios.....	103
Figura 67: Añadir lugar a itinerario	103
Figura 68: Login	104
Figura 69: Registro.....	104
Figura 70: Gestionar usuario.....	105
Figura 71: Gestionar comentarios	106
Figura 72: Actualizar comentarios.....	106
Figura 73: Ver mapa.....	107
Figura 74: Mostrar lugar en el mapa.....	107
Figura 75: Iniciar ruta	107
Figura 76: Mostrar lugares.....	108
Figura 77: Buscar lugar.....	108
Figura 78: Mostrar itinerarios.....	109
Figura 79: Añadir itinerario a favoritos	109
Figura 80: Iniciar itinerario	110
Figura 81: Ver ruta paso a paso del itinerario	110
Figura 82: Mostrar itinerarios favoritos.....	111
Figura 83: Eliminar itinerario favorito	111

ÍNDICE DE TABLAS

Tabla 1: Planificación del trabajo	22
Tabla 2: Work planning	26
Tabla 3: Registro de usuario	36
Tabla 4: Login	37
Tabla 5: Logout	38
Tabla 6: Modificar datos del usuario	39
Tabla 7: Eliminar usuario.....	40
Tabla 8: Mostrar plano del parque	41
Tabla 9: Ubicación actual del usuario.....	42
Tabla 10: Mostrar lugares de interés	42
Tabla 11: Buscar lugar de interés	43
Tabla 12: Agregar comentario	44
Tabla 13: Modificar comentario.....	45
Tabla 14: Eliminar comentario	46
Tabla 15: Iniciar ruta a lugar de interés.....	46
Tabla 16: Ver itinerarios.....	47
Tabla 17: Iniciar itinerario.....	48
Tabla 18: Añadir itinerario favorito	48
Tabla 19: Eliminar itinerario favorito	49
Tabla 20: Ver itinerarios favoritos.....	49
Tabla 21: Añadir itinerario.....	50
Tabla 22: Eliminar itinerario	51
Tabla 23: Eliminar usuario administrador	52
Tabla 24: Eliminar comentario administrador.....	53
Tabla 25: Añadir lugar de interés	54
Tabla 26: Modificar lugar de interés	55
Tabla 27: Eliminar lugar de interés.....	56

Capítulo 1 - Introducción

En este apartado se mostrará la motivación que ha llevado a la elección de este proyecto, los objetivos que pretende cubrir y la planificación que se ha llevado a cabo para su realización.

1.1 Motivación

El Parque del Capricho de Madrid es un jardín histórico de 14 hectáreas situado en el barrio de Alameda de Osuna, el cual alberga una gran riqueza botánica, escultórica y artística, siendo el único jardín del Romanticismo de la capital.

Sin embargo, existen varios inconvenientes que el usuario puede encontrarse, como realizar un recorrido por el parque si no tiene un itinerario que seguir o dirigirse a un lugar de interés en concreto que desea ver.

Además, debido al gran número de lugares de interés del parque y para mejorar la experiencia del usuario, se pretende que pueda conocer todos los lugares que están disponibles y acceder a la información detallada de cada lugar en todo momento, sin necesidad de recurrir a los carteles informativos.

Por otra parte, para ubicarse en el parque actualmente se tiene que recurrir a un mapa físico para ver la ubicación actual y la de los lugares de interés que se quieren visitar.

Por todo esto, se ha desarrollado una aplicación móvil que pretende facilitar la visita de los usuarios proporcionando descripciones detalladas de los lugares e itinerarios disponibles en el parque, así como la posibilidad de mostrar su ubicación en un mapa e iniciar una ruta. Además, también se podrán guardar los itinerarios que más le gusten y poder añadir comentarios en los lugares para contar su experiencia.

1.2 Objetivos

El objetivo principal del trabajo es desarrollar una aplicación móvil que mejore y facilite la experiencia del usuario en el parque y le permita conocer más acerca de los lugares que se encuentran en él.

Profundizando más en ello los objetivos serían:

- Desarrollar una funcionalidad que permita añadir, modificar y eliminar lugares de interés e itinerarios.
- Permitir mostrar la ubicación actual y los lugares en el mapa.
- Ofrecer la posibilidad de guardar los itinerarios que más le gusten.
- Implementar una funcionalidad que permita añadir, modificar y eliminar comentarios a lugares de interés.
- Restringir el acceso a ciertas funcionalidades de la aplicación solo a usuarios registrados.
- Facilitar el acceso a información acerca de los lugares e itinerarios.
- Proporcionar rutas guiadas a lugares e itinerarios.

1.3 Plan de trabajo

Para la consecución de los objetivos descritos en el apartado anterior, se ha seguido un plan de trabajo para la realización del proyecto.

1.3.1 Fases del proyecto

Previamente a la realización del trabajo, se realizó una planificación en la que se especificaban los requisitos que debía cumplir y los plazos en los que se desarrollaría cada tarea. A continuación, se muestran las distintas fases con sus respectivas fechas de realización.

Fase	Fechas
Investigación y especificación de requisitos.	5 septiembre – 10 octubre
Desarrollo de la funcionalidad para la gestión de usuarios.	11 octubre – 31 enero
Implementar el módulo de administrador.	1 febrero – 4 marzo
Desarrollo del módulo relativo a la información del parque.	5 marzo – 10 abril
Implementar la funcionalidad para gestionar los itinerarios.	11 abril – 3 mayo
Realización de la memoria.	4 mayo – 16 mayo

Tabla 1: Planificación del trabajo

Fase 1: Investigación y especificación de requisitos:

La primera fase se basó en la investigación acerca de las posibles funcionalidades que podía tener la aplicación para facilitar las visitas al parque. Posteriormente se realizó la especificación de los requisitos que debería tener implementada la aplicación.

Fase 2: Desarrollo de la funcionalidad para la gestión de usuarios:

En esta fase se realizó el modelo entidad relación para implementar la estructura de la base de datos y se incluyó una tabla para los usuarios. Se implementó todo lo relativo al registro y login del usuario.

Fase 3: Implementar el módulo de administrador:

Se desarrollaron las funcionalidades del administrador: añadir, modificar y eliminar lugares de interés e itinerarios. Además, se añadió la tabla administrador a la base de datos.

Fase 4: Desarrollo del módulo relativo a la información del parque:

Se implementó todo lo relativo a los lugares: las rutas, los comentarios y la ubicación actual del usuario. También se incluyeron los lugares y los comentarios a la base de datos.

Fase 5: Implementar la funcionalidad para gestionar los itinerarios:

En esta fase se añadieron los itinerarios y los itinerarios favoritos, incluyéndose las tablas necesarias para almacenar esta información y completando la estructura final de la base de datos.

Fase 6: Realización de la memoria:

En esta última fase se desarrolló la memoria final del proyecto y se corrigieron pequeños errores en la aplicación.

1.3.2 Organización del trabajo

El proyecto se ha llevado a cabo siguiendo una serie de hitos, con varias tareas cada uno y que fueron planificados al inicio de este. Estos hitos se iniciaron en el segundo cuatrimestre y se dividieron de tal forma que cada uno tuviese una duración de un mes. Al acabar se realizaba una reunión para ver las tareas realizadas y planificar la próxima

iteración. Y en caso de no poder cumplir los plazos, se realizaba una replanificación del próximo hito.

Debido a la mayor carga de asignaturas del primer cuatrimestre, se dividió el trabajo de tal forma que gran parte de este se estructuraría para la segunda mitad del curso.

1.4 Estructura de la memoria

La memoria del proyecto se ha organizado en forma de capítulos de la siguiente manera:

- Capítulo 1: se describe la motivación del trabajo, los objetivos que se pretenden cumplir, el plan de trabajo que se ha llevado a cabo para cumplirlos y la estructura que tiene la memoria.
- Capítulo 2: se muestran otras aplicaciones con funcionalidades similares a las realizadas en este trabajo.
- Capítulo 3: se describen todas las tecnologías que se han utilizado para el desarrollo del proyecto.
- Capítulo 4: se detallan los tres actores que tiene la aplicación: usuario registrado, usuario sin registrar y administrador. Se presentan los distintos diagramas de caso de uso y los cuatro módulos que tendrá la aplicación: usuario, información del parque, itinerarios y administrador.
- Capítulo 5: se explica la arquitectura que tiene la aplicación y el modelo de base de datos utilizado con sus tablas y campos correspondientes.
- Capítulo 6: se describen las distintas funcionalidades que tiene la aplicación y el diseño que se ha realizado mostrando con imágenes cada una de las pestañas de la aplicación.
- Capítulo 7: se desarrollan las conclusiones del trabajo y posibles mejoras para implementar a la aplicación en el futuro.
- Anexo A – Guía de uso: guía para la utilización de cada una de las funcionalidades de la aplicación.

Introduction

This section will show the motivation that has led to the choice of this project, the objectives it intends to cover and the planning that has been carried out for the realization.

Motivation

The Caprice Park in Madrid is a historic garden of 14 acres located in the neighbourhood of Alameda de Osuna, shelter of a great botanical, sculptural and artistic wealth, being the only Romanticism Garden in the capital.

However, there are inconvenient that the user may found, such as taking a tour of the park if he does not have an itinerary to follow or going to a specific place of interest you want to see.

In addition, due to the large number of places of interest in the park and to improve the user experience, it is intended that you can see all the places that are present and always access to the detailed information of each place, without having information in advertisements.

On the other hand, to orientate yourself you must have a physical map to see where you are and places of interest you want to see and visit.

For all this, a mobile application has been developed that aims to facilitate the visit, providing detailed descriptions of the places and itineraries in the park, as well as the possibility of showing their location on a map to start the route. In addition you could save the itineraries that you like the most and being able to add comments to show your experience.

Objectives

The main purpose of this work is to develop a mobile application that improves and facilitates the user experience in the park and allows to learn more about the places in the park.

Going into more detail, the objectives are:

- Implement and develop functionality to allow adding, modifying, and deleting places of interest and itineraries.
- Allow to show current location and places on the map.
- Offer the possibility of saving the itineraries that you like the most.
- Implement and develop functionality to allow adding, modifying, and deleting comments to places of interest.
- Access to certain functionalities for only registered users.
- Easy access to information about places and itineraries.
- Provide guided routes to places and itineraries.

Work plan

To achieve the objectives described a work plan has been followed for the realisation of the project.

Project phases

Before starting this app, a specific plan was settled with deadlines within each task was to be done, as it's follows.

Phase	Dates
Research and requirements specification.	5 September -10 October
Development of user management functionality.	11 October -31 January
Implement administrator module.	1 February -4 March
Development of the park information module.	5 March -10 April
Implement the functionality to manage itineraries.	11 April -3 May
Final report.	4 May – 16 May

Tabla 2: Work planning

Phase 1: Research and requirements specification:

The first phase was based on research the functionalities that the app could have to show visits to the park easily. And the specification of the requirements that the application should have been carried out.

Phase 2: Development of the functionality for user management:

In this phase, a model to create and manage the database and a table for users was included. Register and login user registration was implemented.

Phase 3: implement the administrator module:

The administrator functionalities were developed, such as adding, modifying, and deleting places of interest and itineraries. in addition, the administrator table was included to the database.

Phase 4: Development of the park information module:

Everything related to places was implemented such as routes and comments and displaying the user's current location. Places and comments were also included in the database.

Phase 5: Implement the functionality to manage the itineraries:

In this phase, itineraries and favourite itineraries were added. The final structure of the database including itineraries and favourite itineraries was also made.

Phase 6: Final report:

In this last phase, the final report of the project was developed and small errors in the application were corrected.

Work organization

The project has been made following a series of targets, each with several tasks, which were planned at the beginning of the project. They were started in the second four-month period and were divided in four months. At the end, a meeting was held to review the tasks completed and to plan the next iteration. If the deadlines could not be done, the objective was rescheduled.

Most amount of subjects were in the first term, that's why work was divided in such a way that the large part of it would be structured in the second half of the academic year.

Structure of the report

The project report is organized in chapters as it follows:

- Chapter 1: work motivation, the objectives to achieve, the work plan that has been carried out, and structure of the report.
- Chapter 2: it shows applications with similar functionalities of this work.
- Chapter 3: describes all the technologies that have been used for the development of the project.
- Chapter 4: it talks about the three actors that the application includes: registered user, unregistered user, and administrator. You can see the case diagrams and the four modules that the application will have: user, park information, itineraries, and administrator.
- Chapter 5: application structure and the database model used with its corresponding tables and fields.
- Chapter 6: describes the different functionalities of the application and its design, showing images of each of the application's tabs.
- Chapter 7: work conclusions and improvements to be implemented in the future.
- Annex A - User's guide: guide to use each of the application's functionalities.

Capítulo 2 - Estado del arte

En este apartado se muestran brevemente otras aplicaciones que tienen algunas características parecidas a las realizadas en este trabajo.

2.1 Zoo Aquarium Madrid

[10] Aplicación móvil para ver información de todos los animales que están presentes en el parque. Tiene un mapa en el que se puede ver todos los lugares de interés y los distintos animales ordenados por continentes. También dispone de una serie de rutas que sugiere realizar y muestra un calendario con las distintas actividades y sus respectivos horarios.

2.2 Parques Nacionales

[11] Es una aplicación que permite obtener información sobre la historia, fauna, flora y excursiones que se pueden realizar en los Parques Nacionales de España.

Además, incluye multitud de puntos de interés y rutas que se pueden visualizar sobre un mapa. Se pueden añadir notas y fotos de las distintas rutas y descargar la información de un parque en nuestro dispositivo.

2.3 Parques y jardines de Madrid

[12] Esta aplicación muestra información de esculturas, flora y fauna de los principales parques de Madrid. Permite filtrar por el parque que se quiera ver y obtener una breve descripción de cada lugar con varias imágenes. También se puede visualizar desde un mapa la ubicación de los distintos lugares de interés.

Capítulo 3 - Tecnología empleada

En este apartado se muestran las aplicaciones y tecnologías utilizadas para la realización del trabajo.

3.1 Android Studio

[1] Android Studio es el entorno de desarrollo integrado oficial para aplicaciones Android. Está basado en el software de IntelliJ IDEA y es gratuito. Está disponible en Windows, Linux, macOS, Microsoft Windows y Chrome OS. Actualmente el lenguaje más utilizado es Kotlin, pero también permite otros lenguajes como Java.

3.2 Visual Studio Code

[4] Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux, macOS y Web. Tiene buena integración con Git, cuenta con soporte para depuración de código y dispone de multitud de extensiones, con lo que permite utilizar multitud de lenguajes de programación en él. Además, es gratuito y de código abierto

3.3 Xampp

[3] Xampp es un servidor web local multiplataforma, trabaja tanto en Windows, Linux y Mac que permite la creación y prueba de páginas web. Incluye un sistema de bases de datos MySQL, un servidor web Apache e intérpretes para los lenguajes de script PHP. Es una herramienta de software libre y completamente gratuita.

3.4 Picasso

[5] Picasso es una librería para gestionar imágenes en Android. Permite cargar imágenes de forma sencilla y con uso de memoria mínimo en las aplicaciones. La librería se encarga de almacenar las imágenes de la memoria interna del dispositivo o descargándola de internet.

3.5 X10hosting

[6] X10hosting es un servicio de alojamiento web gratuito que te proporciona los servicios necesarios para crear tu aplicación o página web. Tiene recursos ilimitados en cuanto a ancho de banda y almacenamiento. Además, incluye base de datos PHP y MySQL ilimitado y ofrece un servicio confiable y rápido de manera sencilla.

3.6 PhpMyAdmin

[7] PhpMyAdmin es una herramienta de software libre escrita en PHP, para la administración de MySQL a través de páginas utilizando un navegador web de forma sencilla. Admite un gran número de operaciones en MySQL Y MariaDB, tales como administración de bases de datos, tablas, columnas, relaciones o permisos.

3.7 Google Drive

[9] Google Drive es un servicio desarrollado por Google donde puedes almacenar y acceder a todos tus archivos y se dispone de 15 GB de espacio gratuito. Sirve para guardar, producir y modificar archivos en línea, compartir archivos y poder acceder a ellos desde cualquier dispositivo.

3.8 Mapbox

[2] Mapbox es un proveedor de mapas en línea realizados por encargo para páginas web.

Ofrece una gran cantidad de productos y servicios a los desarrolladores para diseñar mapas personalizados y poder incorporar las tecnologías de información geográfica a cualquier aplicación móvil o web.

3.9 Volley

[8] Volley es una biblioteca desarrollada por Google para facilitar y agilizar el envío de peticiones de http desde las aplicaciones Android a servidores externos. Ofrece programación automática de solicitudes de red, varias conexiones de red simultáneas y compatibilidad con la priorización de solicitudes.

Capítulo 4 - Especificación de requisitos

En este capítulo se van a describir los actores y la especificación de los requisitos de la aplicación.

4.1 Actores

Los actores de la aplicación son los siguientes:

- **Usuario registrado:** son aquellos usuarios que tienen acceso a todas las funciones que ofrece la aplicación.
- **Usuario no registrado:** son aquellos usuarios que tienen acceso a funciones de la aplicación que no requieran almacenar información, como ver el mapa, iniciar itinerarios o poder ver información de los lugares de interés.
- **Administrador:** es la persona encargada de realizar acciones de mantenimiento sobre la aplicación. Se encargará de eliminar cuentas que hagan un mal uso de la aplicación y de añadir y eliminar los distintos itinerarios.

4.2 Diagramas de casos de uso

En esta sección se muestran los diagramas de casos de uso de los cuatro módulos de la aplicación: módulo usuario, como se puede ver en la figura 1, módulo información del parque, que se puede visualizar en la figura 2, módulo itinerarios, como se puede ver en la figura 3 y módulo administrador, que se puede visualizar en la figura 4.

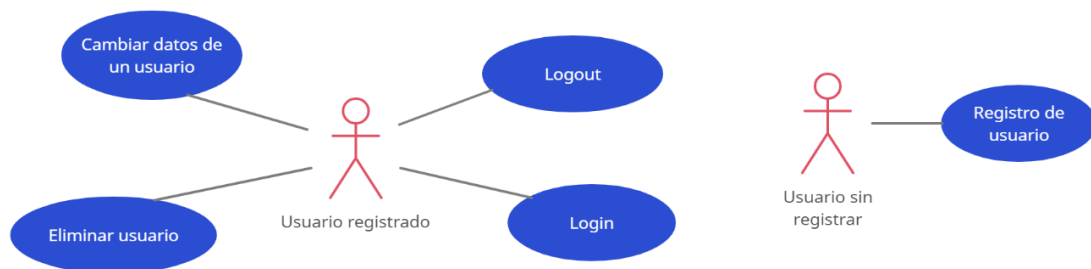


Figura 1: Diagrama de casos de uso módulo usuario

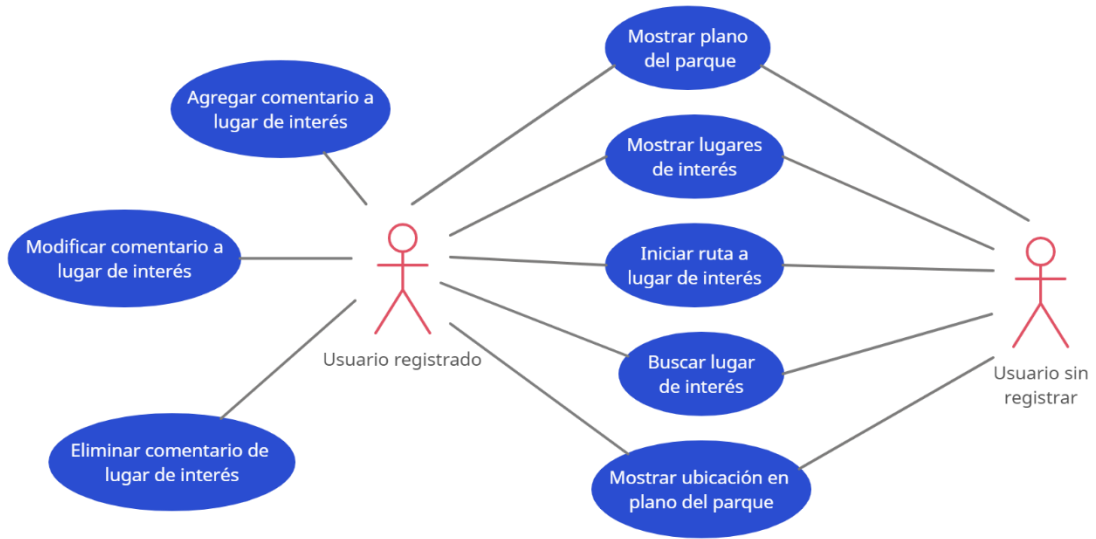


Figura 2: Diagrama de casos de uso módulo información del parque

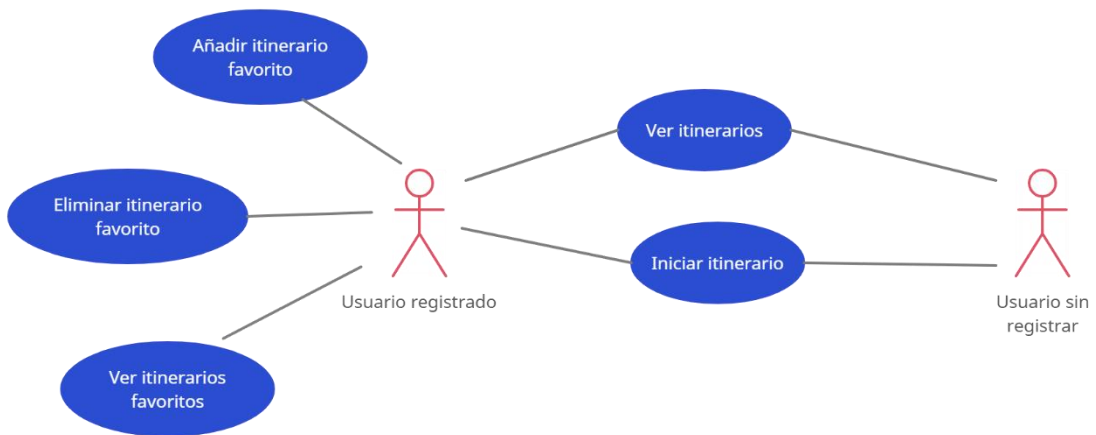


Figura 3: Diagrama de casos de uso módulo itinerarios

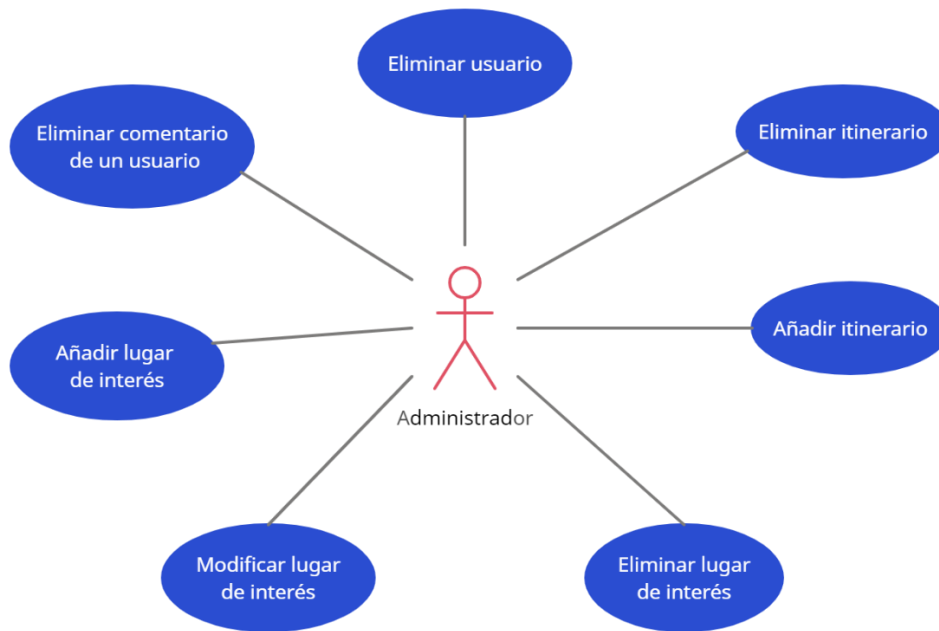


Figura 4: Diagrama de casos de uso módulo administrador

4.3 Especificación de requisitos

A continuación, se van a describir los casos de uso, para lo cual se han agrupado en distintos módulos funcionales.

1. Gestión de usuarios:

- Registro de usuario.
- Login.
- Logout.
- Cambiar datos de un usuario.
- Eliminar usuario.

2. Información del parque:

- Mostrar plano del parque.
- Mostrar ubicación en plano del parque.
- Mostrar lugares de interés.
- Buscar lugar de interés.
- Agregar comentario a lugar de interés.
- Modificar comentario a lugar de interés.
- Eliminar comentario de lugar de interés.

- Iniciar ruta a lugar de interés.

3. Gestión de Itinerarios:

- Ver itinerarios.
- Iniciar itinerario.
- Añadir itinerario favorito.
- Eliminar itinerario favorito.
- Ver itinerarios favoritos.

4. Administrador:

- Añadir itinerario.
- Eliminar itinerario.
- Eliminar usuario.
- Eliminar comentario usuario.
- Añadir lugar de interés.
- Modificar lugar de interés.
- Eliminar lugar de interés.

4.3.1 Casos de uso de Gestión de usuarios

En esta sección se describen los casos de uso del módulo funcional de “Gestión de usuarios”.

Requisito	Registro de usuario	
Identificador	1.1.	
Prioridad	Alta.	
Precondición	NA.	
Descripción	Los usuarios se crean una cuenta para poder acceder a un mayor número de contenidos en la aplicación.	
Entrada	Correo electrónico, usuario y contraseña.	
Salida	NA.	
Secuencia normal	Paso	Acción
	1	El usuario entra en la aplicación y selecciona la opción “Perfil” del menú.
	2	Se abre una nueva pestaña en la que tendrá que pulsar el botón de “Registrar”.
	3	A continuación, en una nueva ventana se pedirán los datos al usuario para poder realizar el registro.
	4	El usuario rellena la información requerida y selecciona la opción de “Registrar”.
	5	Se valida la información introducida por el usuario.
	6	Se vuelve a la pantalla de inicio con el usuario ya registrado correctamente.
Postcondición	El usuario ha creado una cuenta.	
Excepciones	Paso	Acción
	5	Si el correo electrónico, el nombre de usuario o la contraseña no cumplen los requisitos de seguridad se mostrará un mensaje de error solicitando que cambie dichos campos.
	5	Si el correo electrónico ya esta registrado, se mostrará un mensaje de error informando de ello.
Comentarios	NA.	
Actores	Usuario sin registrar, Administrador.	

Tabla 3: Registro de usuario

Requisito	Login	
Identificador	1.2.	
Prioridad	Alta.	
Precondición	El usuario tiene que estar registrado previamente.	
Descripción	Los usuarios que están registrados en el sistema pueden iniciar sesión.	
Entrada	Email y contraseña.	
Salida	NA.	
Secuencia normal	Paso	Acción
	1	El usuario entra en la aplicación y selecciona la opción "Perfil" del menú.
	2	Se abre una nueva pestaña en la que se pedirán los datos al usuario para poder iniciar sesión.
	3	El usuario rellena la información requerida y selecciona la opción de "Login".
	4	Se valida la información introducida por el usuario.
	5	Se vuelve a la pantalla de inicio con el usuario logueado correctamente.
Postcondición	El usuario ha iniciado sesión en la aplicación.	
Excepciones	Paso	Acción
	4	Se muestra un mensaje de error si el correo o la contraseña no son correctas.
Comentarios	NA.	
Actores	Usuario registrado, Administrador.	

Tabla 4: Login

Requisito	Logout	
Identificador	1.3.	
Prioridad	Alta.	
Precondición	El usuario ha tenido que iniciar sesión previamente.	
Descripción	Los usuarios pueden cerrar la sesión en caso de dejar de usar la aplicación o para navegar sin estar registrado.	
Entrada	Nombre de usuario y contraseña.	
Salida	NA.	
Secuencia normal	Paso	Acción
	1	El usuario se dirige a la ventana de "Perfil" del menú.
	2	Presiona sobre el botón de "Logout".
	3	Se cierra la sesión y se vuelve a la pantalla de inicio.
Postcondición	El usuario ha cerrado sesión.	
Excepciones	Paso	Acción
	2	Se muestra un mensaje de error si no se ha podido cerrar la sesión.
Comentarios	NA.	
Actores	Usuario registrado, Administrador.	

Tabla 5: Logout

Requisito	Cambiar datos de un usuario	
Identificador	1.4.	
Prioridad	Media.	
Precondición	El usuario tiene que estar registrado y tener la sesión iniciada.	
Descripción	Los usuarios pueden modificar su información personal.	
Entrada	Nombre de usuario y contraseña.	
Salida	NA.	
Secuencia normal	Paso	Acción
	1	El usuario se dirige a la ventana "Perfil" del menú.
	2	Se abre una nueva pestaña en la que se mostraran los datos para poder modificar los que el usuario desee.
	3	El usuario rellena la información requerida y selecciona la opción de "Actualizar".
	4	Se valida la información introducida.
	5	Se vuelve a la pantalla de inicio con los datos del usuario ya modificados.
Postcondición	El usuario ha cambiado su información de usuario.	
Excepciones	Paso	Acción
	4	Se muestra un mensaje de error si no se ha podido cambiar la información del usuario.
Comentarios	NA.	
Actores	Usuario registrado.	

Tabla 6: Modificar datos del usuario

Requisito	Eliminar usuario	
Identificador	1.5.	
Prioridad	Media.	
Precondición	El usuario debe estar registrado y haber iniciado sesión.	
Descripción	El usuario podrá borrar su cuenta.	
Entrada	Nombre de usuario y contraseña.	
Salida	NA.	
Secuencia normal	Paso	Acción
	1	El usuario entra en la aplicación y se selecciona la opción "Perfil" del menú.
	2	Pulsa sobre el botón "Eliminar" para borrar la cuenta.
	3	Se vuelve a la pantalla de inicio con la cuenta ya eliminada.
Postcondición	El usuario ha eliminado su cuenta.	
Excepciones	Paso	Acción
	2	Se muestra un mensaje de error si no se ha podido eliminar la cuenta.
Comentarios	NA.	
Actores	Usuario registrado.	

Tabla 7: Eliminar usuario

4.3.2 Casos de uso de información del parque

En esta sección se describen los casos de uso del módulo funcional de “Información del parque”.

Requisito	Mostrar plano del parque	
Identificador	2.1.	
Prioridad	Media.	
Precondición	NA.	
Descripción	Muestra un mapa interactivo del parque con su información.	
Entrada	NA.	
Salida	Plano detallado del parque.	
Secuencia normal	Paso	Acción
	1	El usuario entra en la aplicación y selecciona la opción de “Mapa” del menú.
	2	Se abre una nueva pestaña en la que el usuario puede pulsar sobre el plano y se le mostrará información de los distintos elementos.
Postcondición	Se le muestra al usuario la información solicitada.	
Excepciones	Paso	Acción
Comentarios	NA.	
Actores	Usuario sin registrar, usuario registrado.	

Tabla 8: Mostrar plano del parque

Requisito	Mostrar ubicación en plano del parque	
Identificador	2.2.	
Prioridad	Media.	
Precondición	El usuario debe tener activado la geolocalización.	
Descripción	Los usuarios podrán ver su ubicación actual en el parque.	
Entrada	NA.	
Salida	Se muestra un mapa con un punto indicando tu ubicación.	
Secuencia normal	Paso	Acción
	1	El usuario entra en la aplicación y selecciona la opción de "Mapa" del menú.
	2	Se abre una nueva pestaña en la que se muestra el mapa con tu ubicación actual.
Postcondición	Se indica la posición del usuario en el mapa.	
Excepciones	Paso	Acción
	2	Se muestra un mensaje de error si no se ha podido obtener la ubicación del usuario.
Comentarios	NA.	
Actores	Usuario sin registrar, usuario registrado.	

Tabla 9: Ubicación actual del usuario

Requisito	Mostrar lugares de interés	
Identificador	2.3.	
Prioridad	Media.	
Precondición	Los lugares de interés deben estar registrado en la aplicación.	
Descripción	Los usuarios obtienen una lista de los lugares de interés y la información relativa a cada uno.	
Entrada	NA.	
Salida	Información de los lugares de interés.	
Secuencia normal	Paso	Acción
	1	El usuario entra en la aplicación y selecciona la opción "Lugares de Interés" del menú.
	2	Se muestra una lista de todos los lugares de interés del parque, con su información correspondiente.
Postcondición	El usuario ha obtenido la información de los lugares.	
Excepciones	Paso	Acción
Comentarios	NA.	
Actores	Usuario sin registrar, usuario registrado.	

Tabla 10: Mostrar lugares de interés

Requisito	Buscar lugar de interés	
Identificador	2.4.	
Prioridad	Media.	
Precondición	El lugar de interés debe estar registrado en la aplicación.	
Descripción	Los usuarios obtienen la información relativa al lugar de interés solicitado.	
Entrada	NA.	
Salida	Información del lugar de interés.	
Secuencia normal	Paso	Acción
	1	El usuario entra en la aplicación y selecciona la opción de "Lugares de Interés" del menú.
	2	El usuario introduce el nombre del lugar de interés que desea buscar.
	3	Se le muestra al usuario la información del lugar solicitado.
Postcondición	El usuario ha obtenido la información del lugar.	
Excepciones	Paso	Acción
Comentarios	NA.	
Actores	Usuario sin registrar, usuario registrado.	

Tabla 11: Buscar lugar de interés

Requisito	Agregar comentario a lugar de interés	
Identificador	2.5.	
Prioridad	Media.	
Precondición	El usuario debe haber iniciado sesión y el lugar debe estar registrado.	
Descripción	Los usuarios pueden añadir comentarios sobre los lugares del parque.	
Entrada	Comentario introducido por el usuario.	
Salida	NA.	
Secuencia normal	Paso	Acción
	1	El usuario entra en la aplicación y selecciona la opción de "Lugares de Interés" del menú.
	2	Busca el lugar sobre el que quiere añadir una opinión.
	3	Introduce un comentario para el lugar.
	4	Pulsa sobre el botón "Añadir" para agregarlo.
Postcondición	El usuario ha agregado un comentario.	
Excepciones	Paso	Acción
	4	Se muestra un mensaje de error si no se ha podido añadir el comentario.
Comentarios	NA.	
Actores	Usuario registrado.	

Tabla 12: Agregar comentario

Requisito	Modificar comentario de lugar de interés	
Identificador	2.6.	
Prioridad	Baja.	
Precondición	El usuario debe haber iniciado sesión y el lugar debe estar registrado.	
Descripción	Los usuarios pueden modificar sus comentarios sobre los lugares del parque.	
Entrada	Comentario introducido por el usuario.	
Salida	NA.	
Secuencia normal	Paso	Acción
	1	El usuario entra en la aplicación y selecciona la opción de "Lugares de Interés" del menú.
	2	Busca el lugar sobre el que quiere modificar su comentario.
	3	Pulsa sobre el botón "Modificar".
	4	Se abre una nueva ventana para introducir el comentario.
	5	El usuario selecciona el botón "Actualizar" para modificarlo.
	6	Se vuelve a la pantalla de Lugares de Interés con el comentario modificado correctamente.
Postcondición	El usuario ha modificado un comentario.	
Excepciones	Paso	Acción
	5	Se muestra un mensaje de error si no se ha podido actualizar.
Comentarios	NA.	
Actores	Usuario registrado.	

Tabla 13: Modificar comentario

Requisito	Eliminar comentario de lugar de interés	
Identificador	2.7.	
Prioridad	Baja.	
Precondición	El usuario debe haber iniciado sesión y el lugar debe estar registrado.	
Descripción	Los usuarios pueden eliminar sus comentarios sobre los lugares del parque.	
Entrada	Comentario que desea eliminar.	
Salida	NA.	
Secuencia normal	Paso	Acción
	1	El usuario entra en la aplicación y selecciona la opción de "Lugares de Interés" del menú.
	2	Busca el lugar sobre el que quiere eliminar un comentario.
	3	Pulsa sobre el botón "Eliminar" para borrarlo.
Postcondición	El usuario ha eliminado un comentario.	
Excepciones	Paso	Acción
	2	Se muestra un mensaje de error si no se ha podido eliminar.
Comentarios	NA.	
Actores	Usuario registrado.	

Tabla 14: Eliminar comentario

Requisito	Iniciar ruta a lugar de interés	
Identificador	2.8.	
Prioridad	Media.	
Precondición	El usuario debe tener activado la geolocalización.	
Descripción	Los usuarios pueden visitar cualquier lugar de interés iniciando una ruta desde la posición que se encuentran actualmente.	
Entrada	Nombre del lugar.	
Salida	Recorrido para llegar al lugar.	
Secuencia normal	Paso	Acción
	1	El usuario entra en la aplicación y selecciona la opción de "Mapa" del menú.
	2	Pulsa sobre el marker del lugar al que desea ir.
	3	Selecciona la opción de "Iniciar" para empezar la ruta.
Postcondición	El usuario inicia la ruta con el mapa.	
Excepciones	Paso	Acción
Comentarios	NA.	
Actores	Usuario registrado, usuario no registrado.	

Tabla 15: Iniciar ruta a lugar de interés

4.3.3 Casos de uso de gestión de itinerarios

En esta sección se describen los casos de uso del módulo funcional de "Gestión de itinerarios".

Requisito	Ver itinerarios	
Identificador	3.1.	
Prioridad	Media.	
Precondición	NA.	
Descripción	Se muestra una lista con todos los itinerarios disponibles.	
Entrada	NA.	
Salida	Se muestran los itinerarios del parque.	
Secuencia normal	Paso	Acción
	1	El usuario entra en la aplicación y selecciona la opción "Itinerarios" del menú.
	2	Se abre una nueva pestaña en la que se mostrarán los itinerarios disponibles en la aplicación.
Postcondición	Se muestra al usuario los itinerarios de acuerdo con sus criterios de búsqueda.	
Excepciones	Paso	Acción
	2	No hay ningún itinerario en la aplicación.
Comentarios	NA.	
Actores	Usuario sin registrar, usuario registrado.	

Tabla 16: Ver itinerarios

Requisito	Iniciar itinerario	
Identificador	3.2.	
Prioridad	Media.	
Precondición	El itinerario debe existir.	
Descripción	Se inicia el recorrido del itinerario seleccionado.	
Entrada	Nombre del itinerario.	
Salida	Se muestra un mapa con la ruta del itinerario marcada en el mapa.	
Secuencia normal	Paso	Acción
	1	El usuario entra en la aplicación y selecciona la opción "Itinerarios" del menú.
	2	Selecciona la opción de "Empezar" desde la lista de itinerarios.
	3	Se abre una nueva pestaña en la que se muestra el mapa con el recorrido y los puntos por los que pasará.
Postcondición	El usuario ha iniciado un itinerario.	
Excepciones	Paso	Acción
Comentarios	NA.	
Actores	Usuario sin registrar, usuario registrado.	

Tabla 17: Iniciar itinerario

Requisito	Añadir itinerario favorito	
Identificador	3.3.	
Prioridad	Media.	
Precondición	El usuario debe estar registrado y el itinerario debe existir.	
Descripción	Los usuarios pueden agregar los itinerarios que más le gustan a una lista con sus favoritos.	
Entrada	Nombre itinerario.	
Salida	NA.	
Secuencia normal	Paso	Acción
	1	El usuario entra en la aplicación y selecciona la opción "Itinerarios" del menú.
	2	Accede al itinerario que desea añadir a su lista.
	3	El usuario selecciona el botón "Añadir a favoritos".
Postcondición	El usuario ha añadido un itinerario a su lista.	
Excepciones	Paso	Acción
	3	Se muestra un mensaje de error si no se ha podido añadir o ya existe un itinerario favorito con ese nombre.
Comentarios	NA.	
Actores	Usuario registrado.	

Tabla 18: Añadir itinerario favorito

Requisito	Eliminar itinerario favorito	
Identificador	3.4.	
Prioridad	Media.	
Precondición	Debes estar registrado y el itinerario debe estar en la lista de favoritos.	
Descripción	Los usuarios pueden eliminar un itinerario de su lista.	
Entrada	Nombre del itinerario.	
Salida	NA.	
Secuencia normal	Paso	Acción
	1	El usuario entra en la aplicación y selecciona la opción de "Itinerarios favoritos" del menú.
	2	El usuario selecciona el botón "Quitar" del itinerario que desea eliminar.
	3	Se vuelve a la pantalla de inicio con el itinerario ya eliminado.
Postcondición	El usuario ha eliminado un itinerario de su lista.	
Excepciones	Paso	Acción
	2	Se muestra un mensaje de error si no se ha podido eliminar.
Comentarios	NA.	
Actores	Usuario registrado.	

Tabla 19: Eliminar itinerario favorito

Requisito	Ver itinerarios favoritos	
Identificador	3.5.	
Prioridad	Media.	
Precondición	El usuario debe estar registrado.	
Descripción	Los usuarios pueden ver una lista de sus itinerarios favoritos.	
Entrada	NA.	
Salida	Lista de itinerarios.	
Secuencia normal	Paso	Acción
	1	El usuario entra en la aplicación y selecciona la opción de "Itinerarios favoritos" del menú.
	2	Se abre una nueva pestaña en la que se ve una lista con todos los itinerarios que ha agregado.
Postcondición	El usuario ha visto sus itinerarios favoritos.	
Excepciones	Paso	Acción
Comentarios	NA.	
Actores	Usuario registrado.	

Tabla 20: Ver itinerarios favoritos

4.3.4 Casos de uso del administrador

En esta sección se describen los casos de uso del módulo funcional de "Administrador".

Requisito	Añadir itinerario	
Identificador	4.1.	
Prioridad	Media.	
Precondición	El administrador debe haber iniciado sesión y el itinerario no debe existir.	
Descripción	El administrador puede añadir un nuevo itinerario a la aplicación.	
Entrada	Nombre e información del itinerario.	
Salida	NA.	
Secuencia normal	Paso	Acción
	1	El administrador entra en la aplicación y selecciona la opción de "Itinerarios" del menú.
	2	Seleccionará el botón "+" de Añadir o eliminar itinerarios.
	3	Se abre una nueva pestaña en la que se pedirá que indique los puntos del nuevo itinerario.
	4	El administrador rellena la información requerida y selecciona la opción de "Añadir".
	5	Se abre una nueva pestaña en la que se pedirá que introduzca el siguiente lugar del itinerario, presione el botón de "Añadir" y a continuación se le pedirá que siga introduciendo lugares o pulse el botón "Finalizar" en caso de haber acabado.
Postcondición	El administrador ha añadido un nuevo itinerario en la aplicación.	
Excepciones	Paso	Acción
	4	Se muestra un mensaje de error si no se ha podido añadir el itinerario.
Comentarios	NA.	
Actores	Administrador.	

Tabla 21: Añadir itinerario

Requisito	Eliminar Itinerario	
Identificador	4.2.	
Prioridad	Media.	
Precondición	El administrador debe haber iniciado sesión y el itinerario debe existir.	
Descripción	El administrador puede eliminar un itinerario de la aplicación.	
Entrada	Nombre del itinerario.	
Salida	NA.	
Secuencia normal	Paso	Acción
	1	El administrador entra en la aplicación y selecciona la opción de "Itinerarios" del menú.
	2	Seleccionará el botón "+" de Añadir o eliminar itinerarios.
	3	Se abre una nueva pestaña en la que se pedirá al usuario que introduzca el nombre del itinerario que quiere eliminar.
	4	El usuario selecciona el itinerario y selecciona la opción de "Eliminar".
Postcondición	El usuario ha eliminado un itinerario de la aplicación.	
Excepciones	Paso	Acción
	4	Se muestra un mensaje de error si no existe el nombre del itinerario introducido.
Comentarios	NA.	
Actores	Administrador.	

Tabla 22: Eliminar itinerario

Requisito	Eliminar usuario	
Identificador	4.3.	
Prioridad	Media.	
Precondición	El administrador debe estar registrado y haber iniciado sesión.	
Descripción	El administrador podrá borrar cuentas que hagan un mal uso de la aplicación.	
Entrada	Nombre de usuario y contraseña del administrador y nombre de usuario del usuario que se va a eliminar.	
Salida	NA.	
Secuencia normal	Paso	Acción
	1	El administrador entra en la aplicación y selecciona la opción de "Perfil" del menú.
	2	Se abre una nueva pestaña en la que se pedirá que introduzca el email del usuario que se desea eliminar.
	3	El administrador introducirá el nombre del usuario y seleccionará el botón "Eliminar".
	4	Se vuelve a la pantalla de inicio con el usuario ya eliminado.
Postcondición	El administrador ha eliminado a un usuario.	
Excepciones	Paso	Acción
	3	Se muestra un mensaje de error si no existe el usuario introducido.
Comentarios	NA.	
Actores	Administrador.	

Tabla 23: Eliminar usuario administrador

Requisito	Eliminar comentario de un usuario	
Identificador	4.4.	
Prioridad	Baja.	
Precondición	El administrador debe estar registrado y haber iniciado sesión.	
Descripción	El administrador podrá borrar comentarios de un usuario de la aplicación.	
Entrada	Nombre de usuario y comentario que se va a eliminar.	
Salida	NA.	
Secuencia normal	Paso	Acción
	1	El administrador entra en la aplicación y selecciona la opción de "Lugares de interés" del menú.
	2	Se abre una nueva pestaña en la que buscara el lugar de interés del que quiera eliminar el comentario.
	3	El administrador elegirá un comentario y seleccionará la opción "Eliminar" para quitar el comentario de la aplicación.
	4	Se vuelve a la pantalla de inicio con el comentario ya eliminado.
Postcondición	El administrador ha eliminado el comentario de un usuario.	
Excepciones	Paso	Acción
	3	Se muestra un mensaje de error si no se ha podido eliminar el comentario.
Comentarios	NA.	
Actores	Administrador.	

Tabla 24: Eliminar comentario administrador

Requisito	Añadir lugar de interés	
Identificador	4.5.	
Prioridad	Media.	
Precondición	El administrador debe haber iniciado sesión y el lugar de interés no existir ya.	
Descripción	El administrador puede añadir un nuevo lugar de interés a la aplicación.	
Entrada	Nombre e información del lugar de interés.	
Salida	NA.	
Secuencia normal	Paso	Acción
	1	El administrador entra en la aplicación y selecciona la opción de "Lugares de interés" del menú.
	2	Seleccionará el botón "+" de Añadir, actualizar o eliminar lugares.
	3	Se abre una nueva pestaña en la que se pedirá que introduzca los datos del lugar de interés.
	4	El administrador rellena la información requerida y selecciona la opción de "Añadir".
Postcondición	El administrador ha añadido un nuevo lugar de interés a la aplicación.	
Excepciones	Paso	Acción
	4	Se muestra un mensaje de error indicando que no se podido añadir el lugar.
Comentarios	NA.	
Actores	Administrador.	

Tabla 25: Añadir lugar de interés

Requisito	Modificar lugar de interés	
Identificador	4.6.	
Prioridad	Baja.	
Precondición	El administrador debe haber iniciado sesión y el lugar exista ya.	
Descripción	El administrador puede modificar un lugar de interés de la aplicación.	
Entrada	Nombre del lugar de interés.	
Salida	NA.	
Secuencia normal	Paso	Acción
	1	El administrador entra en la aplicación y selecciona la opción de "Lugares de interés" del menú.
	2	Seleccionará el botón "+" de Añadir, actualizar o eliminar lugares.
	3	Se abre una nueva pestaña en la que se pedirá que introduzca los datos que desea modificar.
	4	El administrador rellena la información requerida y selecciona la opción de "Modificar".
Postcondición	El administrador modificado un lugar de interés de la aplicación.	
Excepciones	Paso	Acción
	4	Se muestra un mensaje de error indicando que no se podido modificar el lugar.
Comentarios	NA.	
Actores	Administrador.	

Tabla 26: Modificar lugar de interés

Requisito	Eliminar lugar de interés	
Identificador	4.7.	
Prioridad	Media.	
Precondición	El administrador debe haber iniciado sesión y el lugar de interés debe existir.	
Descripción	El administrador puede eliminar un lugar de la aplicación.	
Entrada	Nombre del lugar de interés.	
Salida	NA.	
Secuencia normal	Paso	Acción
	1	El administrador entra en la aplicación y selecciona la opción de "Lugares de interés" del menú.
	2	Seleccionará el botón "+" de Añadir, actualizar o eliminar lugares.
	3	Se abre una nueva pestaña en la que se pedirá al usuario que introduzca el nombre del lugar que quiere eliminar.
	4	El administrador selecciona la opción de "Eliminar".
Postcondición	El usuario ha eliminado un lugar de interés de la aplicación.	
Excepciones	Paso	Acción
	4	Se muestra un mensaje de error si el nombre del lugar introducido no existe.
Comentarios	NA.	
Actores	Administrador.	

Tabla 27: Eliminar lugar de interés

Capítulo 5 - Arquitectura de la aplicación y modelo de datos

En este apartado se mostrará la arquitectura y el modelo de datos que se ha llevado a cabo para la realización del proyecto.

5.1 Arquitectura de la aplicación

Se ha desarrollado una arquitectura modelo cliente-servidor, la cual está formada por dos componentes, el proveedor y el consumidor. El proveedor es un servidor que presta una serie de recursos, que son consumidos por los clientes.

Hay múltiples clientes que realizan peticiones para obtener los recursos que necesitan y un servidor que es el que da la respuesta al servicio solicitado por el cliente.

En la aplicación el cliente realiza peticiones HTTP a través de la API de Volley, la cual procesa la petición y recupera los datos solicitados de la base de datos y finalmente recibe la respuesta por parte del servidor. También se ha utilizado la API de Mapbox para cargar la información correspondiente a los mapas. Todo el esquema de la arquitectura se puede ver en la figura 5.

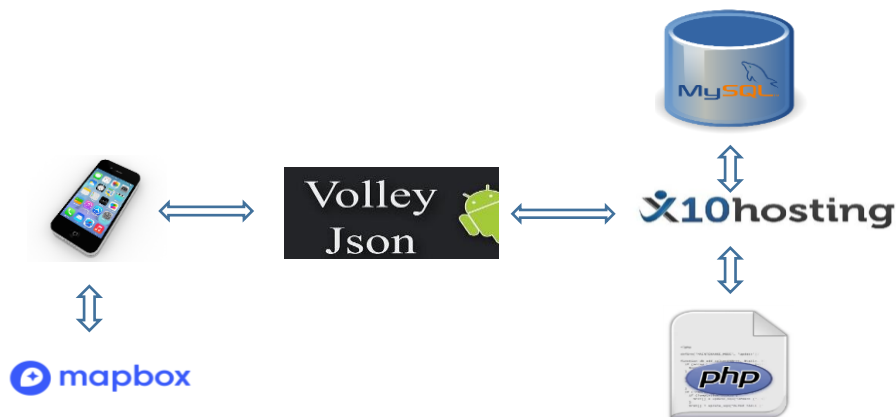


Figura 5: Arquitectura de la aplicación

5.2 Modelo de la base de datos

Para el almacenamiento persistente de información se optó por usar la base de datos MySQL, ya que ofrece un gran rendimiento y flexibilidad. Utilizar una base de datos relacional permite organizar los datos de manera coherente y estructurada. También garantiza la integridad de tablas y relaciones, lo que permite que, por ejemplo, al eliminar un usuario de nuestra aplicación, al estar relacionado con la tabla comentarios se elimine de esta permitiendo tener los datos siempre actualizados de manera consistente.

La información se ha almacenada en cinco tablas: usuarios, para almacenar la información del cliente, lugares, para guardar todo lo relativo a los puntos de interés, comentarios, para almacenar las opiniones de los usuarios en los lugares y por último las tablas de itinerarios e itinerarios favoritos para gestionar las distintas rutas y su información. Estas tablas y sus relaciones se pueden visualizar en la figura 6.

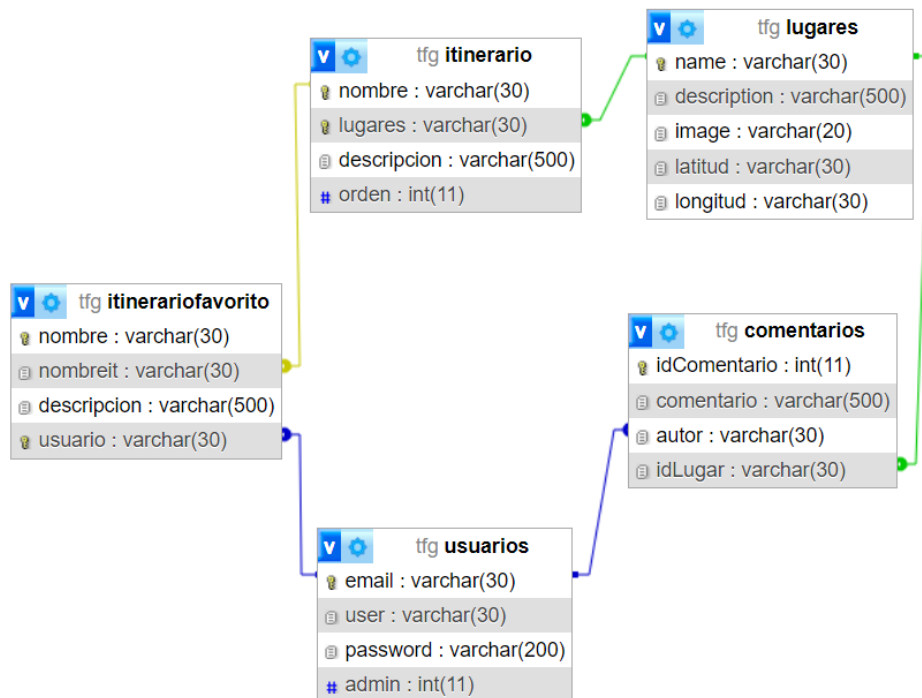


Figura 6: Modelo E.R base de datos

5.2.1 Usuarios

Para gestionar la información de los usuarios, se ha utilizado la tabla usuarios que tiene los siguientes campos:

- email: correo electrónico del usuario que se usa tanto para el registro como para el login.
- user: nombre del usuario en la aplicación.
- password: contraseña que se almacenará cifrada.
- admin: booleano para diferenciar si es administrador o no.

5.2.2 Lugares

Para gestionar los lugares de interés, se ha utilizado la tabla lugares con los siguientes campos:

- name: nombre del lugar de interés.
- description: breve definición del lugar.
- image: imagen del lugar.
- latitud: coordenada para la localización en el mapa.
- longitud: coordenada para la localización en el mapa.

5.2.3 Comentarios

Para los comentarios de los lugares, se ha empleado la tabla comentarios con los siguientes campos:

- idComentario: identificador del comentario.
- comentario: opinión del usuario del lugar.
- autor: creador del comentario.
- idLugar: nombre del lugar de interés.

5.2.4 Itinerario

- nombre: nombre del itinerario.
- lugares: nombre del lugar del itinerario.
- descripción: breve resumen.
- orden: identificador para que al añadir un itinerario siga la disposición en que se añadieron los lugares a este.

5.2.5 Itinerarios favoritos

- nombre: nombre que el usuario ha dado al itinerario.
- nombreit: nombre del itinerario.
- descripción: definición del itinerario.
- usuario: cliente que ha añadido el itinerario.

Capítulo 6 - Implementación y diseño de la aplicación

Para desarrollar la aplicación se han dividido las distintas funcionalidades en cuatro módulos: usuario, información del parque, itinerarios y administrador. Al abrir la aplicación se muestra la pantalla de inicio, como se muestra en la figura 8 y para acceder al menú con las distintas funcionalidades tendrá que pulsar sobre el icono de la parte superior izquierda y se podrá ver una vista como la que se muestra en la figura 7.

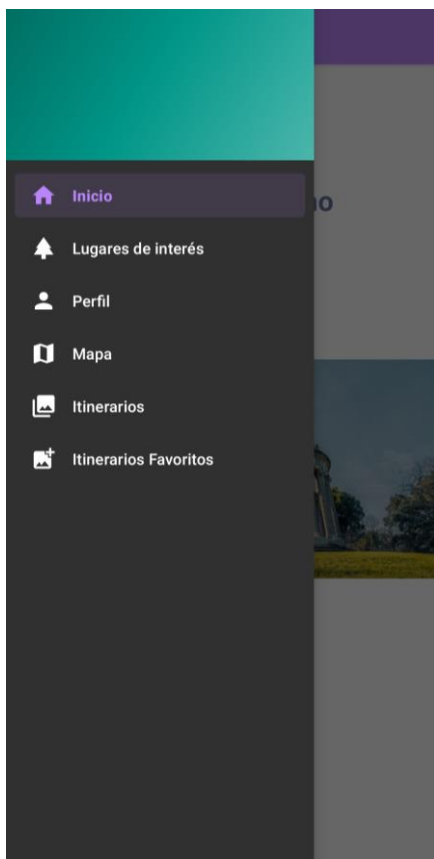


Figura 7: Menú lateral

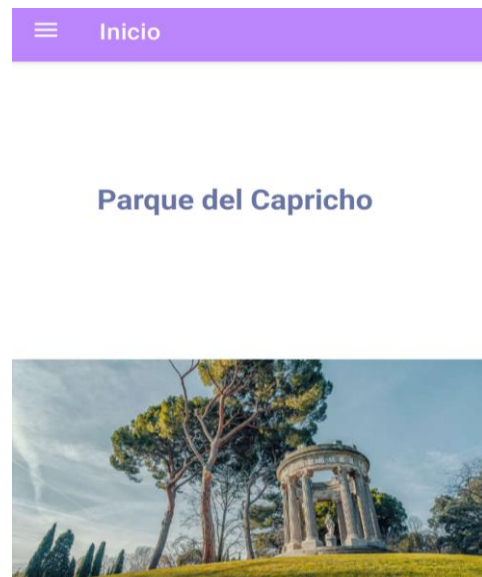


Figura 8: Pantalla de inicio

6.1 Módulo usuario

6.1.1 Registro

Situándose en el menú lateral, seleccionando la opción de perfil y pulsando sobre el botón registrar, el usuario podrá registrarse en la aplicación para poder disponer de un mayor número de funcionalidades, y se mostrará una pantalla como la de la figura 9. Para ello deberá introducir un email, nombre de usuario y contraseña válidos cumpliendo unos requisitos de seguridad, tal como se muestra en el fragmento de código de la figura 11 y si el email introducido ya existe, se mostrará un mensaje de error informándole. En caso de no cumplir los requisitos se mostrará un mensaje de error para que modifique dichos campos, tal como se muestra en la figura 10. Y en el caso de que los valores introducidos sean válidos, se realizará una petición a la API para registrar al usuario en la aplicación, se le redirigirá a la pantalla de inicio y se le mostrará un mensaje de registrado correctamente.

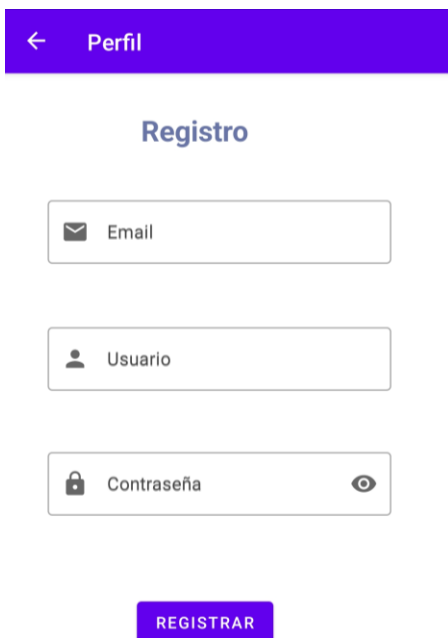


Figura 9: Pantalla de registro de un usuario

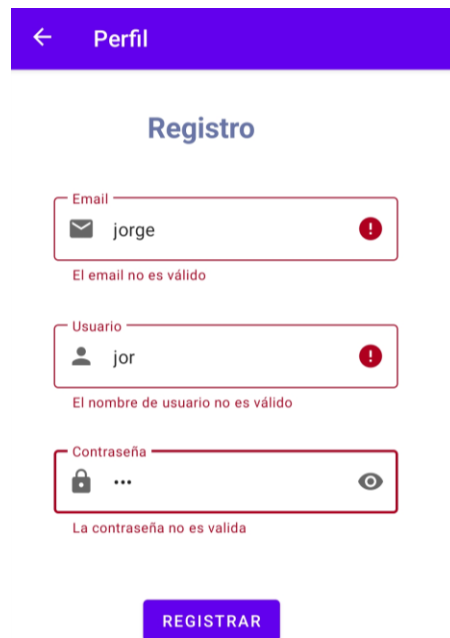


Figura 10: Pantalla de registro con campos no válidos

```

private boolean isValidEmail(String correo) {
    Pattern correoRegex= Pattern.compile(
        regex: "^" +
            "(?=.*[@])" + //1 caracter especial @
            "(?=.*[0-9])" + // 1 digito
            "(?=\s+$)" + // sin espacios
            ".{8,}" + // minimo 8 caracteres
            "$"
    );
    if(correo.isEmpty()){
        emailLayout.setError("El campo email no puede estar vacio");
        emailLayout.setErrorEnabled(true);
        return false;
    }
    else if(!correoRegex.matcher(correo).matches()) {
        emailLayout.setError("El email no es válido");
        emailLayout.setErrorEnabled(true);
        return false;
    }
    else{
        emailLayout.setError(null);
    }
    return true;
}

```

Figura 11: Fragmento de código del registro de un usuario

6.1.2 Login

Si el usuario se ha registrado previamente podrá iniciar sesión en la aplicación, en caso contrario deberá registrarse para poder hacerlo. Para acceder a la pantalla de login deberá situarse en el menú lateral y seleccionar la opción de perfil, se le mostrará una ventana como la de la figura 12. Introducirá su email y contraseña y presionará el botón de login, en caso de no ser correcta se le mostrará un mensaje de error indicándole que el email o la contraseña no son válidas, tal como se puede ver en la figura 13. Se realiza una petición a la API con los datos para comprobar si el usuario esta registrado en la base de datos, tal como se muestra en el fragmento de código de la figura 14, se le redirigirá a la pantalla de inicio y se le mostrará un mensaje de bienvenida.

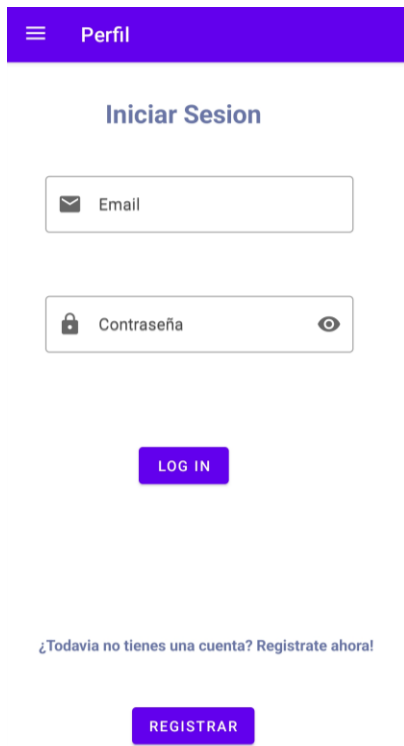


Figura 12: Pantalla de login

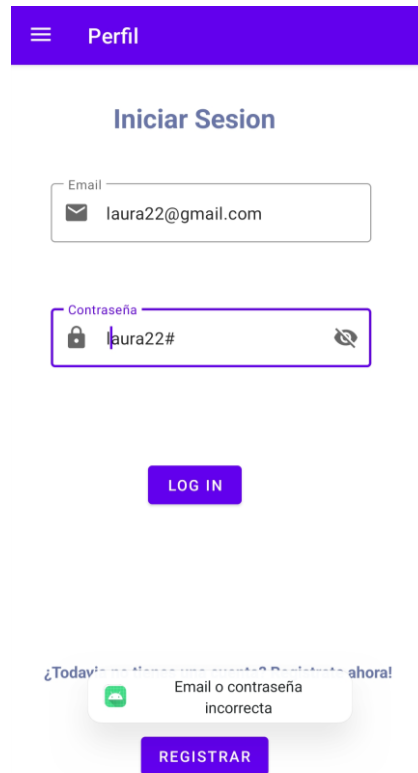


Figura 13: Pantalla de login con mensaje de error

```
private void RunLogin(String URL, View v){
    StringRequest stringRequest= new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            if(!response.isEmpty()){
                preferencesHelper.putIsLogin(true);
                preferencesHelper.putEmail(email);
                preferencesHelper.putPassword(password);
                loginEmail=preferencesHelper.getEmail();
                findUser( URL= obtenerIP().getIP()+"tfq/buscarUsuario.php?email="+loginEmail+"");
                Navigation.findNavController(v).navigate(R.id.action_nav_slideshow_to_nav_home);
                Toast.makeText(getApplicationContext(), text: "Hola "+loginEmail, Toast.LENGTH_SHORT).show();
            }
            else{
                Toast.makeText(getApplicationContext(), text: "Email o contraseña incorrecta", Toast.LENGTH_SHORT).show();
            }
        }
    }, new Response.ErrorListener(){
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(getApplicationContext(), text: "No se ha podido logear", Toast.LENGTH_SHORT).show();
        }
    }){
        @Nullable
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {
            Map<String, String> parametros=new HashMap<>();
            parametros.put("email",email);
            parametros.put("password",password);
            return parametros;
        }
    };
    requestQueue= Volley.newRequestQueue(getApplicationContext());
    requestQueue.add(stringRequest);
}
```

Figura 14: Fragmento de código del login

6.1.3 Logout

En caso de haber iniciado sesión previamente, el usuario podrá cerrarla. El usuario deberá dirigirse al menú lateral, seleccionar la opción de perfil y pulsar sobre el botón logout. Se borrarán los datos de sesión del usuario, tal como se muestra en el fragmento de código de la figura 15. Si se ha podido cerrar la sesión se le redirigirá a la pantalla de inicio y se le mostrará un mensaje de despedida.

```
btnLogout.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Navigation.findNavController(v).navigate(R.id.action_nav_slideshow_to_nav_home);  
        userlogin=preferencesHelper.getUser();  
        adminlogin=preferencesHelper.getIsAdmin();  
        Toast.makeText(getApplicationContext(), text: "Hasta pronto "+userlogin, Toast.LENGTH_SHORT).show();  
        preferencesHelper.logout();  
    }  
});
```

Figura 15: Fragmento de código del logout

6.1.4 Modificar datos

El usuario deberá haber iniciado sesión previamente para poder actualizar su información. Se tendrá que situar en el menú lateral, seleccionar la opción de perfil, se le mostrará una ventana como la de la figura 16, rellenará los campos de nombre de usuario y contraseña cambiados y pulsará sobre el botón de actualizar. Se realizará una petición a la API para modificar los campos en la base de datos, tal como se muestra en el fragmento de código de la figura 17. Si se han modificado los campos correctamente se le redirigirá a la pantalla de inicio y se le mostrará un mensaje informándole, en caso contrario se le mostrará un mensaje de error.



Figura 16: Pantalla de modificar datos del usuario

```
private void modifyuser(String URL){
    StringRequest stringRequest= new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Toast.makeText(getApplicationContext(), text: "Usuario actualizado",Toast.LENGTH_SHORT).show();
            userLogin = userLayout.getEditText().getText().toString();
            preferencesHelper.putUser(userLogin);
        }
    }, new Response.ErrorListener(){
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(getApplicationContext(), text: "No se han podido modificar los datos",Toast.LENGTH_SHORT).show();
        }
    }){
        @Nullable
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {
            Map<String, String> parametros=new HashMap<>();
            parametros.put("email", loginEmail);
            parametros.put("user", userLayout.getEditText().getText().toString());
            parametros.put("password", passwordLayout.getEditText().getText().toString());
            return parametros;
        }
    };
    requestQueue= Volley.newRequestQueue(getApplicationContext());
    requestQueue.add(stringRequest);
}
```

Figura 17: Fragmento de código para modificar datos del usuario

6.1.5 Eliminar cuenta

Para poder eliminar la cuenta el usuario deberá estar registrado previamente en la aplicación. Se tendrá que situar en el menú lateral, seleccionar la opción de perfil y pulsar sobre el botón de eliminar. Se borra toda la información del usuario de la base de datos, tal como se muestra en la figura 18. Si se ha podido eliminar al usuario se le redirigirá a la pantalla de inicio y se le mostrará un mensaje de eliminado correctamente.

```
private void removeuser(String URL){
    StringRequest stringRequest= new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Toast.makeText(getApplicationContext(), text: "Usuario eliminado",Toast.LENGTH_SHORT).show();
        }
    }, new Response.ErrorListener(){
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(getApplicationContext(), text: "No se ha podido eliminar la cuenta",Toast.LENGTH_SHORT).show();
        }
    }){
        @Nullable
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {
            Map<String,String> parametros=new HashMap<>();
            parametros.put("email",loginEmail);
            return parametros;
        }
    };
    requestQueue= Volley.newRequestQueue(getApplicationContext());
    requestQueue.add(stringRequest);
}
```

Figura 18: Fragmento de código para eliminar un usuario

6.2 Módulo de información del parque

6.2.1 Mostrar plano del parque

El usuario podrá ver el plano del parque dirigiéndose al menú lateral y seleccionando la opción de mapa. Es un mapa interactivo que podrá ampliar y reducir e ir moviéndose por los distintos lugares mostrados con un icono de color rojo, tal como se muestra en la figura 19, sobre los iconos podrá pulsar y visualizar el nombre del lugar y una imagen, como se puede ver en la figura 20. Se hace uso de la Api de Mapbox para cargar el mapa, visualizar la ventana de los lugares al pulsar sobre las distintas ubicaciones y obtener la imagen con la librería Picasso, tal como se muestra en el fragmento de código de la figura 21.

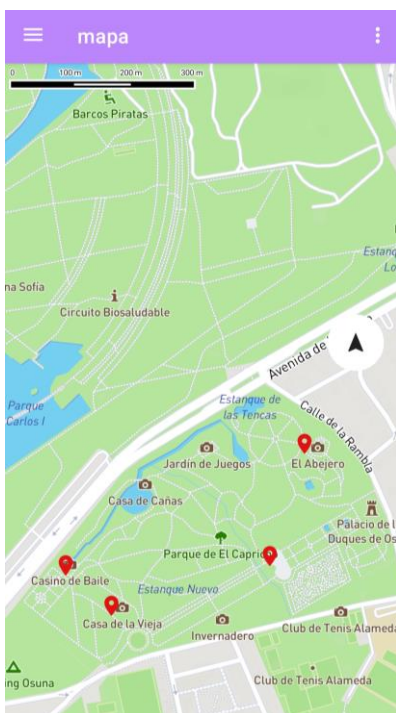


Figura 19: Pantalla del mapa



Figura 20: Pantalla del mapa mostrando lugar de interés

```

private fun createMarkerOnMap(){
    pointAnnotationManager?.apply { this: PointAnnotationManager
        addClickListener(
            OnPointAnnotationClickListener { it ->
                binding.linearlayoutMaps.visibility = View.VISIBLE
                Picasso.get().load(
                    path: obtenerIP!!.ip+"tfq/imagenes/" + lugares[it.id.toInt()].img
                ).into(binding.imagePlac)
                binding.namepla.setText(lugares[it.id.toInt()].name)
                binding.btnCerrarVentana.setOnClickListener(View.OnClickListener { it:View!
                    binding.linearlayoutMaps.visibility = View.INVISIBLE
                })
                var auxlon=lugares[it.id.toInt()].longitud.toDouble()
                var auxlat=lugares[it.id.toInt()].latitud.toDouble()
                binding.btnIni.setOnClickListener(View.OnClickListener { it:View!
                    binding.linearlayoutMaps.visibility = View.INVISIBLE
                    findRoute(Point.fromLngLat(auxlon, auxlat))
                })
                true ^OnPointAnnotationClickListener
            }
        )
    }
}

```

Figura 21: Fragmento de código para mostrar los lugares en el mapa

6.2.2 Mostrar ubicación en plano del parque

Para poder mostrar la ubicación en tiempo real el usuario deberá activar previamente la localización GPS. Se tendrá que dirigir al menú lateral y seleccionar la opción de mapa y ahí se mostrará su ubicación actual, tal como se muestra en la figura 22. Para obtener y mostrar la ubicación del usuario se ha utilizado la Api de Mapbox, como se puede ver en el fragmento de código de la figura 23.

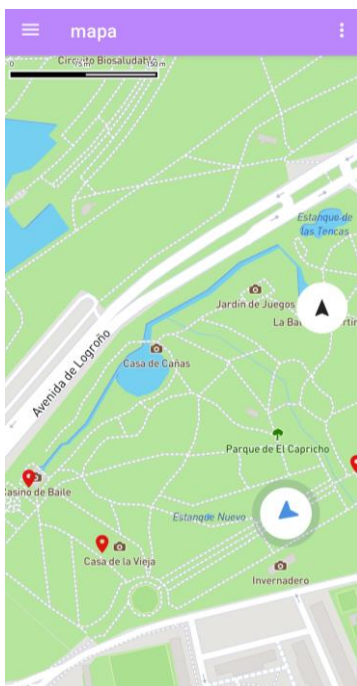


Figura 22: Pantalla del mapa con la ubicación actual del usuario

```

private fun initNavigation() {
    MapboxNavigationApp.setup(
        NavigationOptions.Builder(requireContext())
            .accessToken("pk.eyJ1Ijoiam9yZ2UyMjIiLCJhIjoieY2xmYjQzcDE3MDZzMDNyc6dk...")
            //comment out the location engine setting block to disable simulation
            .build()
    )

    initialize location puck
    binding.mapView.location.apply { this: LocationComponentPlugin
        setLocationProvider(navigationLocationProvider)
        this.locationPuck = LocationPuck2D(
            bearingImage = ContextCompat.getDrawable(
                requireContext(),
                com.mapbox.navigation.R.drawable.mapbox_navigation_puck_icon
            )
        )
        enabled = true
    }
}

```

Figura 23: Fragmento de código para mostrar la ubicación del usuario

6.2.3 Mostrar lugares de interés

El usuario podrá visualizar toda la información relativa a los lugares dirigiéndose al menú lateral y seleccionado la opción de lugares de interés. Se le mostrará una lista con todos los lugares que se encuentran en el parque con su nombre, descripción, imagen y los comentarios, como se muestra en la figura 24. Se hace una llamada a la API para obtener todos los lugares con sus datos respectivos de la base de datos y para visualizarlos se ha utilizado un recyclerView, tal como se muestra en la figura 25.



Figura 24: Pantalla para mostrar los lugares de interés

```

public void findLugares(String URL){
    ArrayList<LugarModelo> lugares=new ArrayList<>();
    JSONArrayRequest jsonArrayRequest=new JSONArrayRequest(URL, new Response.Listener<JSONArray>() {
        @Override
        public void onResponse(JSONArray response) {
            JSONObject jsonObject=null;
            for(int i=0;i<response.length();i++) {
                try {
                    jsonObject=response.getJSONObject(i);
                    lugares.add(new LugarModelo(jsonObject.getString( "name"),
                        jsonObject.getString( "description"),
                        jsonObject.getString( "image"),
                        jsonObject.getString( "longitud"),
                        jsonObject.getString( "latitud")));
                } catch (JSONException e) {
                    Toast.makeText(getApplicationContext(), "Error catch", Toast.LENGTH_SHORT).show();
                }
            }
            recyclerViewLugarAdapter =new RecyclerViewLugarAdapter(lugares,getContext());
            recyclerView.setAdapter(recyclerViewLugarAdapter);
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            //Toast.makeText(getApplicationContext(), "Error find place", Toast.LENGTH_SHORT).show();
        }
    });
    RequestQueue requestQueue= Volley.newRequestQueue(getApplicationContext());
    requestQueue.add(jsonArrayRequest);
}

```

Figura 25: Fragmento de código para mostrar los lugares

6.2.4 Buscar lugar de interés

Se podrá buscar un lugar filtrando por su nombre. Para ello el usuario deberá dirigirse al menú lateral y seleccionar la opción de lugares de interés. En la parte superior, deberá presionar el icono de la lupa y a la derecha deberá introducir el nombre del lugar que desea buscar y entonces se mostrará el lugar seleccionado, tal como se muestra en la figura 26. Para filtrar por nombre del lugar en la recyclerView se ha utilizado el widget SearchView, tal como se muestra en la figura 27.



Figura 26: Buscar lugar

```

public void filtrar(String txtBuscar){
    int longitud=txtBuscar.length();
    if(longitud==0){
        lugarLista.clear();
        lugarLista.addAll(listaOriginal);
    }
    else{
        List<LugarModelo> collection=lugarLista.stream().filter
            (i -> i.getName().toLowerCase().contains(txtBuscar.toLowerCase()))
            .collect(Collectors.toList());
        lugarLista.clear();
        lugarLista.addAll(collection);
    }
    notifyDataSetChanged();
}
}

```

Figura 27: Fragmento de código para buscar los lugares

6.2.5 Agregar comentario a lugar de interés

En caso de estar registrados en la aplicación, los usuarios podrán añadir comentarios sobre los distintos lugares del parque. Tendrá que ir al menú lateral y seleccionar la opción de lugares de interés. Buscará el lugar sobre el que desea añadir el comentario y pulsará sobre saber más. Se dirigirá a la sección de comentarios y sobre el cuadro de texto introducirá la opinión y presionará el botón de añadir. Se realizará una petición a la API para agregar un nuevo comentario de un usuario a la base de datos, tal como se muestra en la figura 28. Se le mostrará un mensaje indicando que el comentario se ha podido añadir al lugar.

```

private void addComment(String URL,ViewHolder holder){
    StringRequest stringRequest= new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Toast.makeText(context, text: "Comentario añadido",Toast.LENGTH_SHORT).show();
        }
    }, new Response.ErrorListener(){
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(context, text: "No se ha podido añadir el comentario",Toast.LENGTH_SHORT).show();
        }
    });
    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> parametros=new HashMap<>();
        parametros.put("comentario",holder.comentariolayout.getEditText().getText().toString());
        parametros.put("autor",preferencesHelper.getEmail());
        parametros.put("idLugar",holder.name.getText().toString());
        return parametros;
    }
};
requestQueue= Volley.newRequestQueue(context);
requestQueue.add(stringRequest);
}

```

Figura 28: Fragmento de código para agregar comentarios

6.2.6 Modificar comentario a lugar de interés

Los usuarios que estén registrados podrán modificar los comentarios que hayan hecho de cualquier lugar. Para ello se tendrá que dirigir al menú lateral y seleccionar la opción de lugares de interés. Buscará el lugar sobre el que desea modificar el comentario y pulsará sobre saber más, verá la pantalla como se muestra en la figura 29. Se dirigirá a la sección de comentarios y pulsará sobre el botón modificar que se encuentra debajo del comentario que desea actualizar. Se le redirigirá a otra ventana donde deberá introducir el comentario modificado y pulsar sobre el botón actualizar para cambiarlo. Se realizará una petición a la API para actualizar el comentario de la base de datos, tal como se muestra en la figura 30. Finalmente, se le redirigirá a la ventana de lugares de interés y se le mostrará un mensaje indicando que el comentario ha sido actualizado.

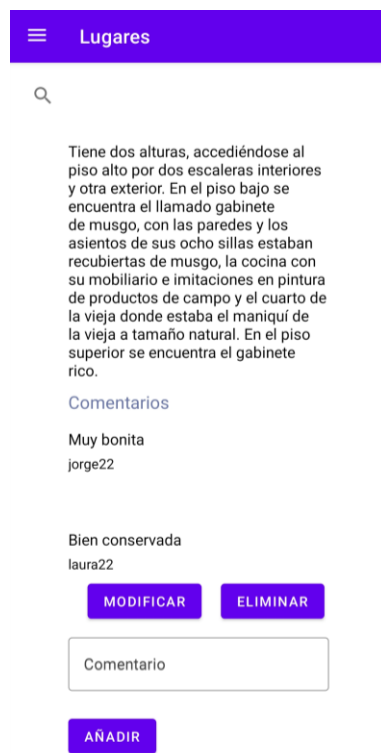


Figura 29: Pantalla para añadir, modificar y eliminar comentarios

```

private void modifyComment(String URL){
    StringRequest stringRequest= new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Toast.makeText(getApplicationContext(), text: "Comentario actualizado", Toast.LENGTH_SHORT).show();
        }
    }, new Response.ErrorListener(){
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(getApplicationContext(), text: "No se ha podido actualizar el comentario", Toast.LENGTH_SHORT).show();
        }
    });
    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> parametros=new HashMap<>();
        parametros.put("idComentario", preferencesHelper.getIdComment());
        parametros.put("comentario", comentarioUpdate.getEditText().getText().toString());
        return parametros;
    }
};
RequestQueue requestQueue= Volley.newRequestQueue(getApplicationContext());
requestQueue.add(stringRequest);
}

```

Figura 30: Fragmento de código para modificar comentarios

6.2.7 Eliminar comentario de lugar de interés

Estando registrado en la aplicación el usuario podrá eliminar cualquier comentario que haya escrito sobre los lugares del parque. Se dirigirá al menú lateral y tendrá que seleccionar la opción de lugares de interés. Buscará el lugar sobre el que desea borrar el comentario y pulsará sobre saber más. Se dirigirá a la sección de comentarios y pulsará sobre el botón eliminar que se encuentra debajo del comentario que desea borrar. Se realizará una petición a la API para eliminar un comentario de la base de datos, tal como se muestra en la figura 31. Se mostrará un mensaje indicando que el comentario se ha eliminado.

```

private void removeComment(String URL){
    StringRequest stringRequest= new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Toast.makeText(context, text: "Comentario eliminado", Toast.LENGTH_SHORT).show();
        }
    }, new Response.ErrorListener(){
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(context, text: "No se ha podido eliminar el comentario", Toast.LENGTH_SHORT).show();
        }
    });
    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> parametros=new HashMap<>();
        parametros.put("idComentario", idComentario);
        return parametros;
    }
};
requestQueue= Volley.newRequestQueue(context);
requestQueue.add(stringRequest);
}

```

Figura 31: Fragmento de código para eliminar comentarios

6.2.8 Iniciar ruta a lugar de interés

Se podrán iniciar rutas a lugares si el usuario ha activado previamente su ubicación en el dispositivo para poder obtener su ubicación actual. Se dirigirá al menú lateral y tendrá que seleccionar la opción de mapa. Elegirá uno de los iconos que representan a los distintos lugares y se mostrará una ventana con el lugar que ha seleccionado, a continuación, presionará sobre el botón de iniciar y se mostrará la ruta paso a paso que el usuario deberá seguir para llegar al lugar, tal como se muestra en las figuras 32 y 33. Para obtener la ruta al lugar se hace uso de la API Mapbox Navigation, tal como se muestra en la figura 34.

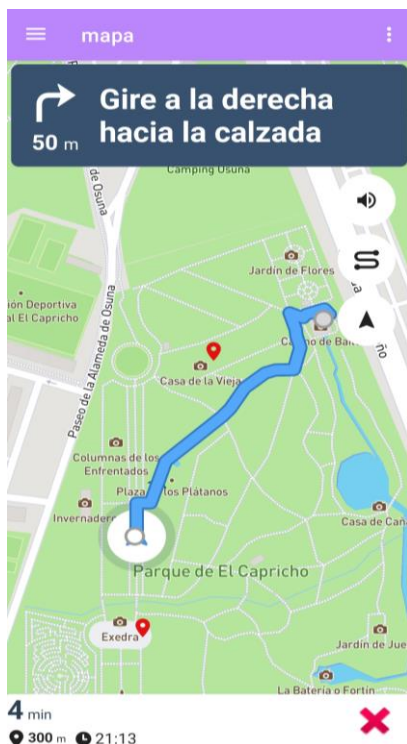


Figura 32: Pantalla iniciar ruta

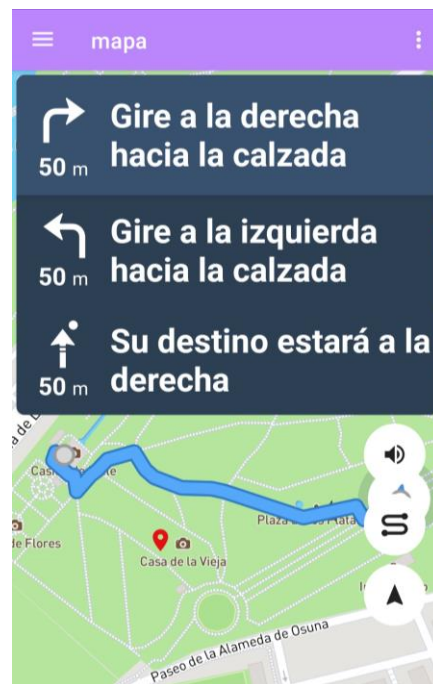


Figura 33: Pantalla mostrar ruta paso a paso

```

private fun findRoute(destination: Point) {
    val originLocation = navigationLocationProvider.lastLocation
    val originPoint = originLocation?.let { it: Location
        Point.fromLngLat(it.longitude, it.latitude)
    } ?: return
    execute a route request
    mapboxNavigation.requestRoutes(
        RouteOptions.builder() RouteOptions.Builder
            .applyDefaultNavigationOptions(DirectionsCriteria.PROFILE_WALKING) RouteOptions.Builder
            .applyLanguageAndVoiceUnitOptions(requireContext())
            .coordinatesList(ListOf(originPoint, destination))
            .build(),
        object : NavigationRouterCallback {
            override fun onCancelled(routeOptions: RouteOptions, routerOrigin: RouterOrigin) {
            }
            override fun onFailure(reasons: List<RouterFailure>, routeOptions: RouteOptions) {
            }
            override fun onRoutesReady(
                routes: List<NavigationRoute>,
                routerOrigin: RouterOrigin
            ) {
                setRouteAndStartNavigation(routes)
            }
        }
    )
}

```

Figura 34: Fragmento de código para iniciar una ruta a un lugar

6.3 Módulo de itinerarios

6.3.1 Ver itinerarios

El usuario podrá visualizar toda la información relativa a los itinerarios dirigiéndose al menú lateral y seleccionado la opción de itinerarios. Se le mostrará una lista con todos los itinerarios que se encuentran en el parque con su nombre y una breve descripción, tal como se muestra en la figura 35 y pulsando el desplegable se pueden ver los lugares que forman el itinerario con una imagen y su descripción, tal como se muestra en la figura 36. Se hace una llamada a la API para obtener todos los itinerarios con su información respectiva de la base de datos y para visualizarlos se ha utilizado un recyclerView, tal como se muestra en la figura 37.

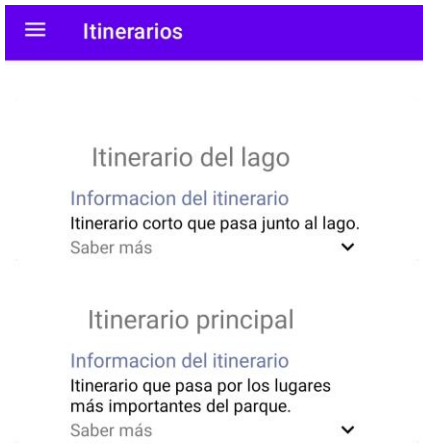
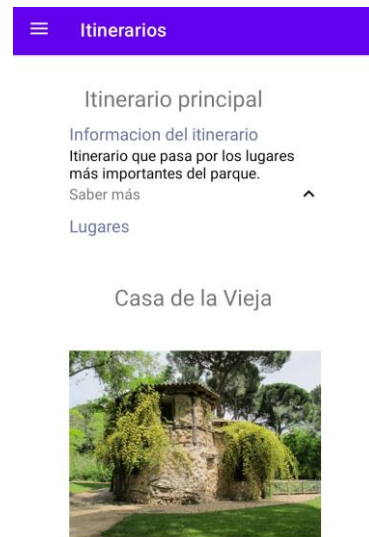


Figura 35: Pantalla para visualizar itinerarios



Tiene dos alturas, accediéndose al piso alto por dos escaleras interiores y otra exterior. En el piso bajo se encuentra el llamado gabinete de musgo, con las paredes y los asientos de sus ocho sillas estaban recubiertas de musgo, la cocina con su mobiliario e imitaciones en pintura de productos de campo y el cuarto de

Figura 36: Pantalla para ver los lugares del itinerario

```

public void findItinerarios(String URL){
    ArrayList<ItinerarioModelo> itinerarios=new ArrayList<>();
    JSONArrayRequest jsonArrayRequest=new JSONArrayRequest(URL, new Response.Listener<JSONArray>() {
        @Override
        public void onResponse(JSONArray response) {
            JSONObject jsonObject=null;
            for(int i=0;i<response.length();i++) {
                try {
                    jsonObject=response.getJSONObject(i);
                    itinerarios.add(new ItinerarioModelo(jsonObject.getString( "name": "nombre"),
                        jsonObject.getString( "name": "descripcion")));
                } catch (JSONException e) {
                    Toast.makeText(getApplicationContext(), "Error catch", Toast.LENGTH_SHORT).show();
                }
            }
            recyclerViewItinerarioAdapter =new RecyclerViewItinerarioAdapter(itinerarios,getContext());
            recyclerView.setAdapter(recyclerViewItinerarioAdapter);
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            //Toast.makeText(getApplicationContext(), "Error find itinerario", Toast.LENGTH_SHORT).show();
        }
    });
    RequestQueue requestQueue= Volley.newRequestQueue(getApplicationContext());
    requestQueue.add(jsonArrayRequest);
}

```

Figura 37: Fragmento de código para mostrar los itinerarios

6.3.2 Iniciar itinerario

Se podrán iniciar rutas de los itinerarios si el usuario ha activado previamente su ubicación en el dispositivo para poder obtener su ubicación actual. Se dirigirá al menú lateral, tendrá que seleccionar la opción de itinerarios y pulsar sobre el botón empezar del itinerario que quiera iniciar la ruta. Se le redirigirá a una nueva ventana con el mapa donde tendrá que seleccionar el botón iniciar para comenzar el itinerario y se le mostrará la ruta paso a paso que debe seguir, tal como se puede ver en las figuras 38 y 39. Para obtener la ruta del itinerario se hace uso de la API Mapbox Navigation añadiendo una lista de los puntos de las distintas paradas que tendrá la ruta, tal como se muestra en la figura 40.

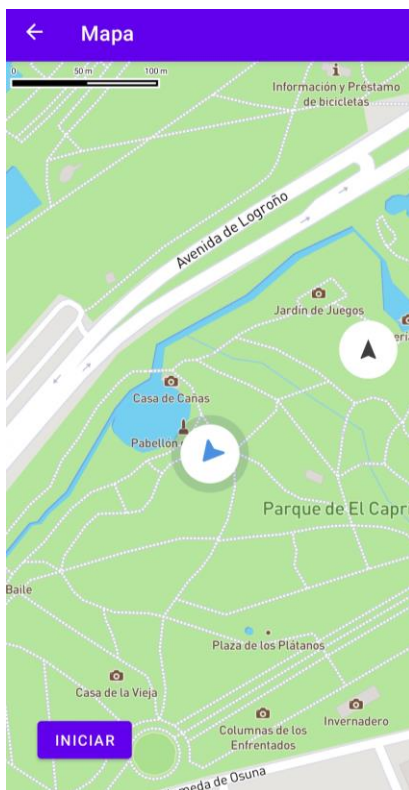


Figura 38: Iniciar itinerario

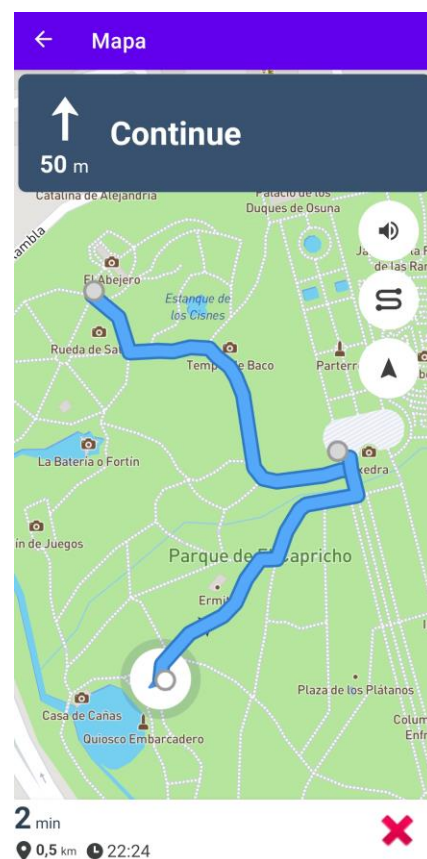


Figura 39: Ruta paso a paso del itinerario

```

protected fun findRouteItinerario(latitudeListIt: ArrayList<Double>, longitudeListIt: ArrayList<Double>) {
    val originLocation = navigationLocationProvider.lastLocation
    val originPoint = originLocation?.let { @ Location
        Point.fromLngLat(it.Longitude, it.Latitude)
    } ?: return
    val points: ArrayList<Point> = ArrayList()
    points.add(originPoint)
    for (i in 0..until < latitudeListIt.size){
        points.add(Point.fromLngLat(longitudeListIt[i], latitudeListIt[i]));
    }
}

execute a route request
mapboxNavigation.requestRoutes(
    RouteOptions.builder() RouteOptions.Builder()
        .applyDefaultNavigationOptions(DirectionCriteria.PROFILE_WALKING) RouteOptions.Builder()
        .coordinatesList(points)
        .build(),
    object : NavigationRouterCallback {
        override fun onCancelled(routeOptions: RouteOptions, routerOrigin: RouterOrigin) {
        }
        override fun onFailure(reasons: List<RouterFailure>, routeOptions: RouteOptions) {
        }
        override fun onRoutesReady(
            routes: List<NavigationRoute>,
            routerOrigin: RouterOrigin
        ) {
            setRouteAndStartNavigation(routes)
        }
    }
)
}

```

Figura 40: Fragmento de código para iniciar la ruta de un itinerario

6.3.3 Añadir itinerario favorito

El usuario debe haberse registrado previamente para poder añadir los itinerarios que más le gusten a su lista de favoritos en la aplicación. Se dirigirá al menú lateral, tendrá que seleccionar la opción de itinerarios y pulsar sobre el botón añadir a favoritos al final del itinerario, tal como se muestra en la figura 41. Para añadir un itinerario a favorito se hace una petición a la Api para guardarlo en la base de datos, tal como se muestra en la figura 42. Finalmente, se le mostrará un mensaje indicándole que el itinerario se ha añadido a favoritos o un mensaje de error informándole de que ya existe un itinerario favorito con ese nombre.



Figura 41: Pantalla para añadir un itinerario a favoritos

```
private void addItinerariofav(String URL){
    StringRequest stringRequest= new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            try {
                JSONObject jsonResponse = new JSONObject(response);
                boolean error = jsonResponse.getBoolean( name: "error");
                String message = jsonResponse.getString( name: "message");
                if (error) {
                    Toast.makeText(context, message, Toast.LENGTH_SHORT).show();
                } else {
                    preferencesHelper.putNombreIt(nombreItfavorito);
                    Toast.makeText(context, "Itinerario favorito añadido", Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    }, new Response.ErrorListener(){
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(context, "No se ha podido añadir el itinerario favorito", Toast.LENGTH_SHORT).show();
        }
    });
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> parametros=new HashMap<String, String>();
        parametros.put("nombre", nombreItfavorito);
        parametros.put("nombreit", nombreItfavorito);
        parametros.put("descripcion", descripcionItfav);
        parametros.put("usuario", preferencesHelper.getEmail());
        return parametros;
    }
};
requestQueue= Volley.newRequestQueue(context);
requestQueue.add(stringRequest);
```

Figura 42: Fragmento de código para añadir un itinerario a favoritos

6.3.4 Eliminar itinerario favorito

Se podrán eliminar itinerarios favoritos de tu lista, si el usuario se ha registrado previamente en la aplicación. Se dirigirá al menú lateral, tendrá que seleccionar la opción de itinerarios favoritos y pulsar sobre el botón quitar situado al final, del itinerario que quiera eliminar, tal como se muestra en la figura 43. Para eliminar un itinerario a favorito se hace una petición a la Api para borrarlo de la base de datos, tal como se muestra en la figura 44. Se le mostrará un mensaje indicándole que el itinerario se ha eliminado de favoritos.



Figura 43: Pantalla para iniciar o eliminar un itinerario de favoritos

```

private void removeItinerarioFav(String URL, String nombre){
    StringRequest stringRequest= new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Toast.makeText(context, R.string.Itinerario_favorito_eliminado, Toast.LENGTH_SHORT).show();
        }
    }, new Response.ErrorListener(){
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(context, R.string.No_se_ha_pudido_eliminar_el_itinerario_favorito, Toast.LENGTH_SHORT).show();
        }
    });
    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> parametros=new HashMap<>();
        parametros.put("nombre", nombre);
        parametros.put("usuario", preferencesHelper.getEmail());
        return parametros;
    }
};
requestQueue= Volley.newRequestQueue(context);
requestQueue.add(stringRequest);
}

```

Figura 44: Fragmento de código para eliminar un itinerario

6.3.5 Ver itinerarios favoritos

El usuario puede visualizar toda la información relativa a los itinerarios favoritos dirigiéndose al menú lateral y seleccionado la opción de itinerarios favoritos. En caso de no haber iniciado se le mostrará un mensaje y un enlace para que se logue, tal como se muestra en la figura 45. Si el usuario ha iniciado sesión se le mostrará una lista con todos los itinerarios que ha añadido a su lista con el nombre que le ha puesto al itinerario favorito, nombre del itinerario, una breve descripción y los lugares que forman el itinerario con una imagen y su descripción, tal como se muestra en la figura 46. Se hace una llamada a la API para obtener todos los itinerarios favoritos con su información respectiva de la base de datos y para visualizarlos se ha utilizado un recyclerView, tal como se muestra en la figura 47.



Inicia sesion para ver tus itinerarios favoritos

LOG IN

Figura 45: Pantalla itinerario favoritos sin hacer login

Casa de la Vieja



Tiene dos alturas, accediéndose al piso alto por dos escaleras interiores y otra exterior. En el piso bajo se encuentra el llamado gabinete de musgo, con las paredes y los asientos de sus ocho sillas estaban recubiertas de musgo, la cocina con

Figura 46: Pantalla ver itinerario favoritos

```
public void findItinerariosfav(String URL){
    ArrayList<ItinerarioModelo> itinerariosfav=new ArrayList<>();
    JSONArrayRequest jsonArrayRequest=new JSONArrayRequest(URL, new Response.Listener<JSONArray>() {
        @Override
        public void onResponse(JSONArray response) {
            JSONObject jsonObject=null;
            for(int i=0;i<response.length();i++){
                try {
                    jsonObject=response.getJSONObject(i);
                    itinerariosfav.add(new ItinerarioModelo(jsonObject.getString("nombre"),
                        jsonObject.getString("nombre"),jsonObject.getString("descripcion")));
                } catch (JSONException e) {
                    Toast.makeText(getApplicationContext(), "Error catch", Toast.LENGTH_SHORT).show();
                }
            }
            RecyclerViewItinerarioFavoritoAdapter =new RecyclerViewItinerarioFavoritoAdapter(getApplicationContext());
            recyclerView.setAdapter(recyclerViewItinerarioFavoritoAdapter);
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            //Toast.makeText(getApplicationContext(), "Error find itinerario fav", Toast.LENGTH_SHORT).show();
        }
    });
}

@Nullable
@Override
protected Map<String, String> getParams() throws AuthFailureError {
    Map<String, String> parametros=new HashMap<>();
    parametros.put("usuario",preferencesHelper.getEmail());
    return parametros;
}

RequestQueue requestQueue= Volley.newRequestQueue(getApplicationContext());
requestQueue.add(jsonArrayRequest);
}
```

Figura 47: Fragmento de código para mostrar los itinerarios favoritos

6.4 Administrador

6.4.1 Añadir itinerario

El usuario administrador podrá añadir nuevos itinerarios a la aplicación, para ello debe haber iniciado sesión previamente. Se dirigirá al menú lateral, tendrá que seleccionar la opción de itinerarios y pulsar sobre el botón "+" en la parte superior y se le redirigirá a otra pantalla donde deberá rellenar los datos del itinerario, tal como se muestra en la figura 48.

A continuación, pulsará sobre el botón añadir y se redirigirá a otra pantalla para que siga introduciendo lugares que componen el itinerario pulsando el botón de añadir y cuando haya terminado de añadir todos los lugares pulsará sobre el botón finalizar, tal como se muestra en la figura 49. Para añadir un itinerario se hace una petición a la Api para guardarlo en la base de datos, tal como se muestra en la figura 50. Finalmente, se le redirigirá a la pantalla de itinerarios y se mostrará un mensaje indicándole que se ha añadido el itinerario.

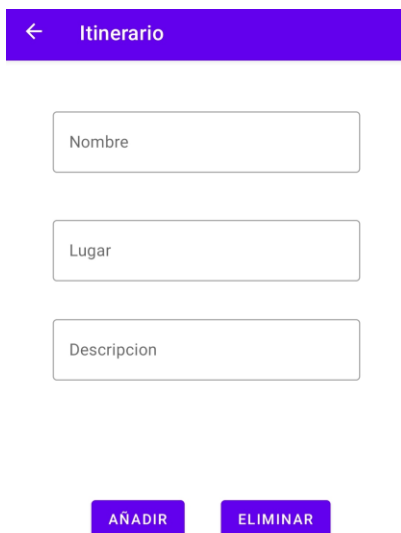


Figura 48: Pantalla itinerarios administrador

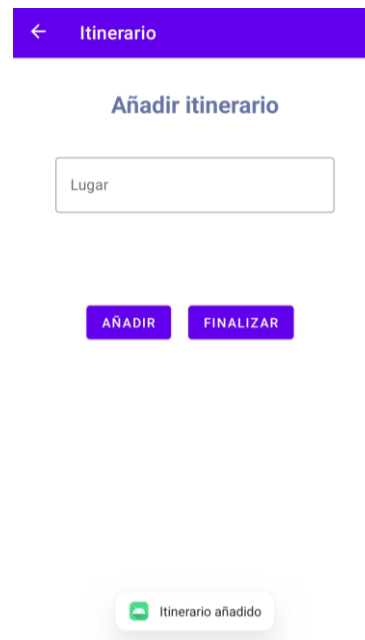


Figura 49: Pantalla para añadir lugares al itinerario

```

private void addItinerario(String URL){
    StringRequest stringRequest= new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            preferencesHelper.putNombreIt(nombre);
            preferencesHelper.putLugarIt(lugares);
            preferencesHelper.putDescripcionIt(descripcion);
            Toast.makeText(getContext(), text: "Itinerario añadido", Toast.LENGTH_SHORT).show();
        }
    }, new Response.ErrorListener(){
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(getContext(), text: "No se ha podido añadir el itinerario", Toast.LENGTH_SHORT).show();
        }
    });
}

@Nullable
@Override
protected Map<String, String> getParams() throws AuthFailureError {
    Map<String, String> parametros=new HashMap<>();
    parametros.put("nombre", nombre);
    parametros.put("lugares", lugares);
    parametros.put("descripcion", descripcion);
    return parametros;
}
};
requestQueue= Volley.newRequestQueue(getContext());
requestQueue.add(stringRequest);
}

```

Figura 50: Fragmento de código para añadir un itinerario

6.4.2 Eliminar itinerario

El usuario debe ser administrador y haberse registrado previamente para disponer de esta funcionalidad. Se dirigirá al menú lateral, tendrá que seleccionar la opción de itinerarios y se le muestra una pantalla como la de la figura 51, pulsará sobre el botón “+” en la parte superior y se le redirigirá a otra pantalla donde deberá rellenar el nombre del itinerario que desea borrar. Finalmente, pulsará sobre el botón eliminar para quitarlo de la lista de itinerarios y si el nombre del itinerario no existe, se le mostrará un mensaje de error informándole de ello, en caso contrario, se mostrará un mensaje indicándole que se ha eliminado el itinerario. Para eliminar un itinerario se hace una petición a la Api para eliminarlo de la base de datos, tal como se muestra en la figura 52.

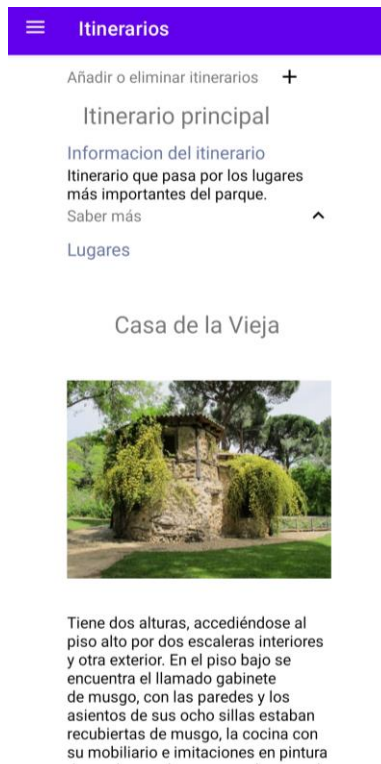


Figura 51: Pantalla para acceder a añadir o eliminar un itinerario

```

private void removeItinerario(String URL){
    StringRequest stringRequest= new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            try {
                JSONObject jsonResponse = new JSONObject(response);
                boolean error = jsonResponse.getBoolean( name: "error");
                String message = jsonResponse.getString( name: "message");
                if (error) {
                    Toast.makeText(getApplicationContext(), text: "El itinerario no existe", Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(getApplicationContext(), text: "Itinerario eliminado", Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    }, new Response.ErrorListener(){
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(getApplicationContext(), text: "No se ha podido eliminar el itinerario", Toast.LENGTH_SHORT).show();
        }
    });
    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> parametros=new HashMap<>();
        parametros.put("nombre", nombreLayout.getEditText().getText().toString());
        return parametros;
    }
};
requestQueue= Volley.newRequestQueue(getApplicationContext());
requestQueue.add(stringRequest);
}

```

Figura 52: Fragmento de código para eliminar un itinerario

6.4.3 Eliminar usuario

El usuario administrador podrá eliminar usuarios de la aplicación, para ello debe haber iniciado sesión previamente. Se dirigirá al menú lateral, tendrá que seleccionar la opción de perfil e introducir el correo del usuario que desea eliminar en la parte inferior de la pantalla, tal como se muestra en la figura 53. Para eliminar un usuario de la aplicación se hace una petición a la Api para borrarlo de la base de datos, tal como se muestra en la figura 54. A continuación, pulsará sobre el botón eliminar y se redirigirá a la pantalla de inicio y se mostrará un mensaje de usuario eliminado o en caso de que el usuario no exista, se le mostrará un mensaje de error.

Perfil

Modificar Datos

Usuario

Contraseña

ELIMINAR ACTUALIZAR LOG OUT

Eliminar usuario

Email

ELIMINAR

Figura 53: Pantalla para que el administrador elimine a un usuario

```

private void removeuserAdmin(String URL){
    StringRequest stringRequest= new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            try {
                JSONObject jsonResponse = new JSONObject(response);
                boolean error = jsonResponse.getBoolean( "name": "error");
                String message = jsonResponse.getString( "name": "message");
                if (error) {
                    Toast.makeText(getContext(), text: "No existe un usuario con ese email", Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(getContext(), text: "Usuario eliminado", Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        Toast.makeText(getContext(), text: "No se ha podido eliminar la cuenta", Toast.LENGTH_SHORT).show();
    }
});
}
@Nullable
@Override
protected Map<String, String> getParams() throws AuthFailureError {
    Map<String, String> parametros=new HashMap<>();
    parametros.put("email",emailLayoutAdmin.getEditText().getText().toString());
    return parametros;
}
};
requestQueue= Volley.newRequestQueue(getContext());
requestQueue.add(stringRequest);
}

```

Figura 54: Fragmento de código para que el administrador elimine un usuario

6.4.4 Eliminar comentario de usuario

El usuario debe ser administrador y haberse registrado previamente para poder borrar comentarios de otros usuarios. Se dirigirá al menú lateral y tendrá que seleccionar la opción de lugares de interés. Buscará el lugar sobre el que desea borrar el comentario y pulsará sobre saber más, tal como se puede ver en la figura 55. Se situará en la sección de comentarios y pulsará sobre el botón eliminar que se encuentra debajo del comentario que desea borrar. Se mostrará un mensaje indicando que el comentario se ha eliminado.

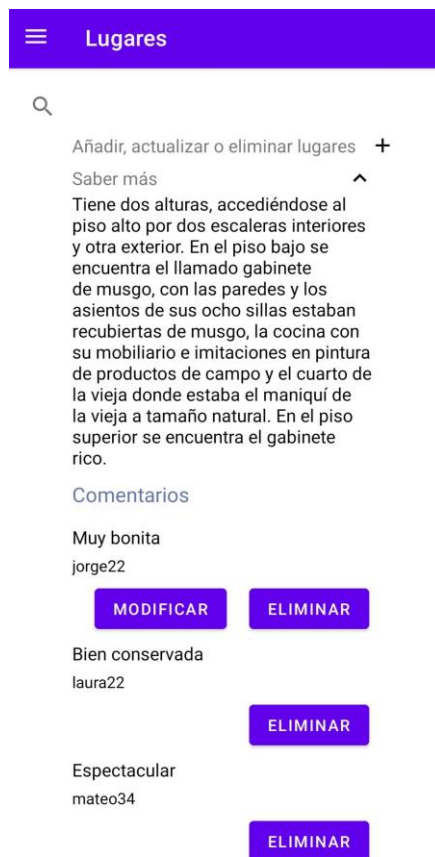


Figura 55: Pantalla para que el administrador elimine comentarios

6.4.5 Añadir lugar de interés

El usuario administrador podrá añadir nuevos lugares a la aplicación, para ello debe haber iniciado sesión previamente. Se dirigirá al menú lateral, tendrá que seleccionar la opción de lugares de interés y pulsar sobre el botón “+” en la parte superior y se le redirigirá a otra pantalla donde deberá rellenar los datos del lugar, tal como se muestra en la figura 56. Para añadir un lugar de interés se hace una petición a la Api para guardarlo en la base de datos, tal como se muestra en la figura 57. A continuación, pulsará sobre el botón añadir y se redirigirá a la pantalla de inicio y se mostrará un mensaje de lugar añadido.

←
Lugares

AÑADIR

MODIFICAR

ELIMINAR

Figura 56: Pantalla para añadir, modificar o eliminar un lugar

```

private void addPlace(String URL){
    StringRequest stringRequest= new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            preferencesHelper.putNamePlace(name);
            preferencesHelper.putDescPlace(description);
            preferencesHelper.putImg(image);
            preferencesHelper.putLongPlace(longitud);
            preferencesHelper.putLatPlace(latitud);
            Toast.makeText(getContext(), text: "Lugar añadido",Toast.LENGTH_SHORT).show();
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(getContext(), text: "No se ha podido añadir el lugar",Toast.LENGTH_SHORT).show();
        }
    });
    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> parametros=new HashMap<>();
        parametros.put("name",name);
        parametros.put("description",description);
        parametros.put("image",image);
        parametros.put("longitud",longitud);
        parametros.put("latitud",latitud);
        return parametros;
    }
};
requestQueue= Volley.newRequestQueue(getContext());
requestQueue.add(stringRequest);
}

```

Figura 57: Fragmento de código para añadir un lugar

6.4.6 Modificar lugar de interés

El usuario debe ser administrador y haberse registrado previamente para poder actualizar los lugares. Se dirigirá al menú lateral, tendrá que seleccionar la opción de lugares de interés, se mostrará una pantalla como la de la figura 58, pulsará sobre el botón "+" en la parte superior y se le redirigirá a otra pantalla donde deberá rellenar el nombre, descripción e imagen del lugar que desea actualizar. Para modificar un lugar se hace una petición a la Api para actualizarlo en la base de datos, tal como se muestra en la figura 59. A continuación, pulsará sobre el botón modificar y se redirigirá a la pantalla de inicio y se mostrará un mensaje de lugar actualizado.

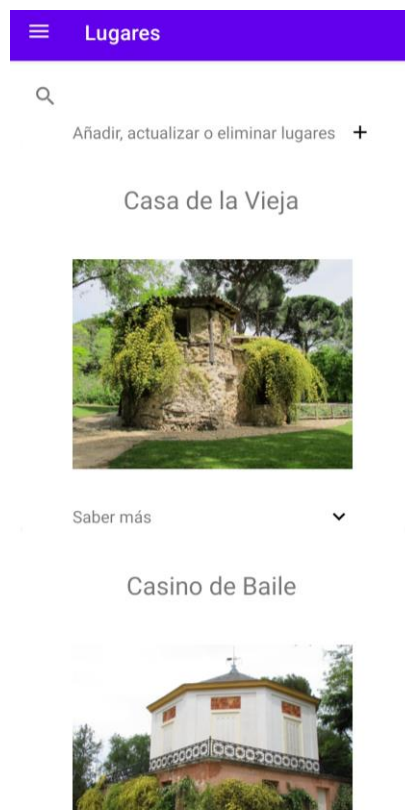


Figura 58: Pantalla para acceder a añadir, modificar o eliminar un lugar

```

private void modifyPlace(String URL){
    StringRequest stringRequest= new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            preferencesHelper.putNamePlace(name);
            preferencesHelper.putDescPlace(description);
            preferencesHelper.putIng(image);
            preferencesHelper.putLongPlace(Longitud);
            preferencesHelper.putLatPlace(latitud);
            Toast.makeText(getApplicationContext(), R.string.lugar_actualizado, Toast.LENGTH_SHORT).show();
        }
    }, new Response.ErrorListener(){
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(getApplicationContext(), R.string.no_se_ha_podido_actualizar_el_lugar, Toast.LENGTH_SHORT).show();
        }
    });
    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> parametros=new HashMap<>();
        parametros.put("name", nameLayout.getEditText().getText().toString());
        parametros.put("description", descriptionLayout.getEditText().getText().toString());
        parametros.put("image", imageLayout.getEditText().getText().toString());
        parametros.put("longitud", longitudLayout.getEditText().getText().toString());
        parametros.put("latitud", latitudLayout.getEditText().getText().toString());
        return parametros;
    }
};
requestQueue= Volley.newRequestQueue(getApplicationContext());
requestQueue.add(stringRequest);
}

```

Figura 59: Fragmento de código para modificar un lugar

6.4.7 Eliminar lugar de interés

El usuario administrador podrá eliminar lugares de la aplicación, para ello debe haber iniciado sesión previamente. Se dirigirá al menú lateral, tendrá que seleccionar la opción de itinerarios y pulsar sobre el botón “+” en la parte superior y se le redirigirá a otra pantalla donde deberá rellenar el nombre del lugar que desea borrar. Para eliminar un lugar de interés se hace una petición a la Api para borrarlo de la base de datos, tal como se muestra en la figura 60. Finalmente, pulsará sobre el botón eliminar para quitarlo de la lista de lugares y se mostrará un mensaje indicándole que se ha eliminado el lugar o un mensaje de error informándole que el lugar introducido no existe.

```

private void removePlace(String URL){
    StringRequest stringRequest= new StringRequest(Request.Method.POST, URL, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            try {
                JSONObject jsonResponse = new JSONObject(response);
                boolean error = jsonResponse.getBoolean("error");
                String message = jsonResponse.getString("message");
                if (error) {
                    Toast.makeText(getApplicationContext(), R.string.el_lugar_no_existe, Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(getApplicationContext(), R.string.lugar_eliminado, Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    }, new Response.ErrorListener(){
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText(getApplicationContext(), R.string.no_se_ha_podido_eliminar_el_lugar, Toast.LENGTH_SHORT).show();
        }
    });
    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> parametros=new HashMap<>();
        parametros.put("name", nameLayout.getEditText().getText().toString());
        return parametros;
    }
};
requestQueue= Volley.newRequestQueue(getApplicationContext());
requestQueue.add(stringRequest);
}

```

Figura 60: Fragmento de código para eliminar un lugar

Capítulo 7 - Conclusiones y trabajo futuro

7.1 Conclusiones

El proyecto se ha realizado con el objetivo de facilitar las visitas al Parque del Capricho de Madrid, pretendiendo que los usuarios que visiten el parque tengan una mejor experiencia disfrutando al máximo de su estancia en él.

Para conseguirlo se ha optado por realizar una aplicación móvil en la que el visitante pueda obtener información de todos los lugares de interés e itinerarios del parque de manera sencilla y desde cualquier lugar. Además, dispone de un mapa a través del cual puede ver donde se encuentran los distintos lugares de interés e iniciar tanto rutas a lugares como itinerarios.

También tendrá la posibilidad de registrarse en la aplicación para disfrutar de todas las funcionalidades y así poder añadir comentarios a los lugares y guardarse los itinerarios que más le gusten.

7.2 Trabajo futuro

La mayor parte de los objetivos propuestos al inicio del trabajo en la especificación de requisitos han sido realizados, pese a esto existen nuevas características que podrían ser implementadas para mejorar la aplicación en el futuro.

- Mostrar todos los puntos de interés del parque: sería importante que en el mapa además de mostrar los lugares de interés se mostrase donde se encuentran los aseos y los accesos al parque.
- Añadir comentarios a los itinerarios: actualmente, solo se permite que los usuarios registrados puedan añadir opiniones sobre los lugares de interés y se podría incorporar una nueva funcionalidad para agregar comentarios sobre los itinerarios.
- Mejorar la información de itinerarios y lugares: la información que se puede visualizar sobre los distintos lugares e itinerarios es una breve descripción del lugar

y una imagen de este. Se podría mostrar información más detallada y varias imágenes para conocer mejor cada lugar.

- Añadir imágenes y valoraciones de los lugares: los usuarios que estén registrados además de añadir comentarios puedan añadir fotografías y valoraciones de los lugares.

Conclusions and future work

Conclusions

The project has been carried out with the aim of facilitating visits to the Caprice Park in Madrid, so that users who visit the park can have a better experience enjoying their walks around.

To achieve this, a mobile application has been created to obtain information about all the places of interest and itineraries in the park easily from anywhere. In addition, it has a map through which you can see the different places of interest are located and start both routes to places and itineraries.

You will also have the possibility of registering in the application to enjoy all the functionalities and add comments to the places and save the itineraries you like the most.

Future work

The objectives and requirements marked at the beginning of this project has been done, however there are new features that could be implemented to improve the application in the future:

- Show every point of interest in the park: it would be important that the map shows not only the places of interest but also where the toilets are and access to the park.
- Add comments to the itineraries: currently, only registered users are allowed to add visit opinions, a new feature could be added with comments on the itineraries.
- Improve information on itineraries and places: the information that can be displayed is brief description of the place and an image of it, could show more information and several images can be settled to know better that location.
- Add images and comments: registered users can add not only comments but also photographs and evaluations of the garden.

Bibliografía

- [1] Android Studio: <https://developer.android.com/studio>
- [2] Mapbox: <https://www.mapbox.com/>
- [3] Xampp: <https://www.apachefriends.org/es/index.html>
- [4] Visual Studio Code: <https://code.visualstudio.com/>
- [5] Picasso: <https://desarrollador-android.com/librerias/square/picasso/>
- [6] X10hosting: <https://x10hosting.com/>
- [7] PhpMyAdmin: <https://www.phpmyadmin.net/>
- [8] Volley: <https://developer.android.com/training/volley?hl=es-419>
- [9] Google Drive: https://www.google.com/intl/es_es/drive/
- [10] Zoo Aquarium Madrid:
<https://play.google.com/store/apps/details?id=com.mobail.zoo&pli=1>
- [11] Parques Nacionales:
<https://play.google.com/store/apps/details?id=es.ign.parquesnacionales&hl=es&gl=US>
- [12] Parques y jardines de Madrid:
https://play.google.com/store/apps/details?id=com.parques_jardines_madrid
- [13] Traductor: <https://www.linguee.es/>
- [14] Instalación Mapbox: <https://docs.mapbox.com/android/maps/guides/install/>
- [15] Markers Mapbox: <https://docs.mapbox.com/android/maps/examples/default-point-annotation/>
- [16] Navegación Mapbox:
<https://docs.mapbox.com/android/navigation/examples/turn-by-turn-experience/>
- [17] Google Maps: <https://developers.google.com/maps>
- [18] JSON: <https://es.wikipedia.org/wiki/JSON>
- [19] PHP: <https://www.php.net/manual/es/index.php>
- [20] SQL: <https://es.wikipedia.org/wiki/SQL>
- [21] HTTP: <https://developer.mozilla.org/es/docs/Web/HTTP>
- [22] Kotlin: <https://kotlinlang.org/>
- [23] Java: <https://www.java.com/es/>

[24] Fragments: <https://developer.android.com/guide/components/fragments?hl=es-419>

[25] Permisos ubicación:

<https://developer.android.com/training/location/permissions?hl=es-419>

[26] RecyclerView:

<https://developer.android.com/guide/topics/ui/layout/recyclerview?hl=es-419>

[27] Shared Preferences: <https://developer.android.com/training/data-storage/shared-preferences?hl=es-419>

Anexo A: Guía de Uso

En este anexo se explicará cómo usar la aplicación, en primer lugar, para usuarios administradores y finalmente para los demás usuarios.

A.1 Eliminar Usuario

La figura 61 muestra la pantalla que verá el administrador tras haber iniciado sesión y haberse dirigido a la opción de perfil del menú lateral. Para eliminar a un usuario de la aplicación tendrá que introducir el email del usuario y presionar sobre el botón eliminar.

The screenshot displays a user profile management interface. At the top, there is a purple header bar with a hamburger menu icon and the text 'Perfil'. Below this, the section 'Modificar Datos' is visible, containing two input fields: 'Usuario' with a person icon and 'Contraseña' with a lock icon and a toggle eye icon. Below these fields are three purple buttons: 'ELIMINAR', 'ACTUALIZAR', and 'LOG OUT'. The section 'Eliminar usuario' follows, featuring an 'Email' input field with an envelope icon and a purple 'ELIMINAR' button.

Figura 61: Administrador eliminar usuario

A.2 Eliminar comentario de un usuario

Para eliminar el comentario de cualquier usuario, el administrador deberá dirigirse a la opción lugares del menú lateral y buscar el lugar sobre el que desea borrar un comentario. A continuación, presionará sobre el desplegable de saber más y ahí seleccionará el botón eliminar situado debajo de cada comentario, tal como se muestra en la figura 62.

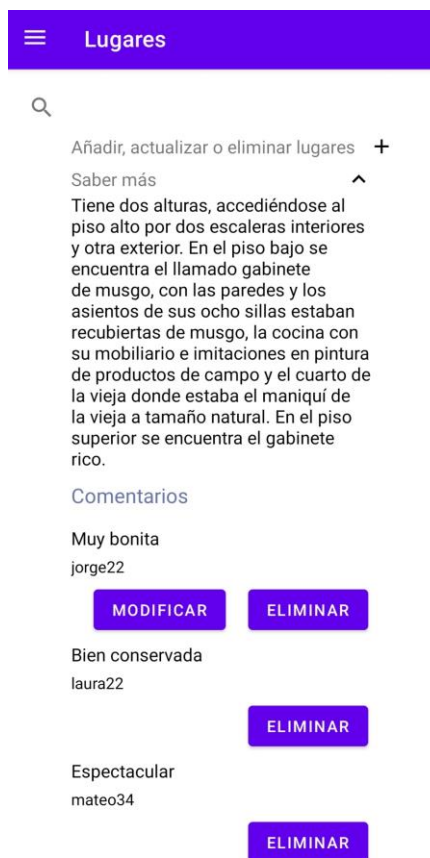


Figura 62: Administrador eliminar comentario

A.3 Gestión de lugares

Para añadir lugares al parque, el administrador deberá seleccionar la opción de lugares del menú lateral y pulsar sobre el botón + a la derecha de añadir, modificar o eliminar lugares, tal como se muestra en la figura 63. Se abrirá una nueva ventana, tal como se muestra en la figura 64, en la que deberá introducir el nombre del lugar, una descripción, el nombre de la imagen, la longitud, la latitud del lugar y pulsar sobre el botón añadir, en el caso de querer agregar un nuevo lugar o rellenar todos los campos y pulsar sobre el botón modificar, en el caso de querer actualizarlo o introducir el nombre del lugar y pulsar sobre el botón eliminar, en el caso de querer borrarlo.

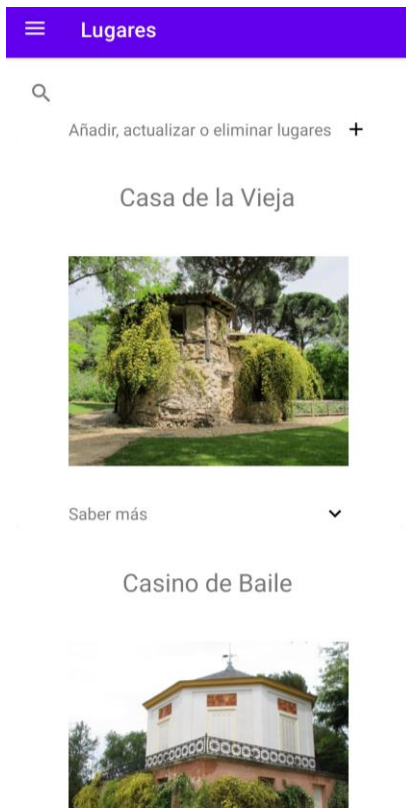


Figura 63: Mostrar lugares

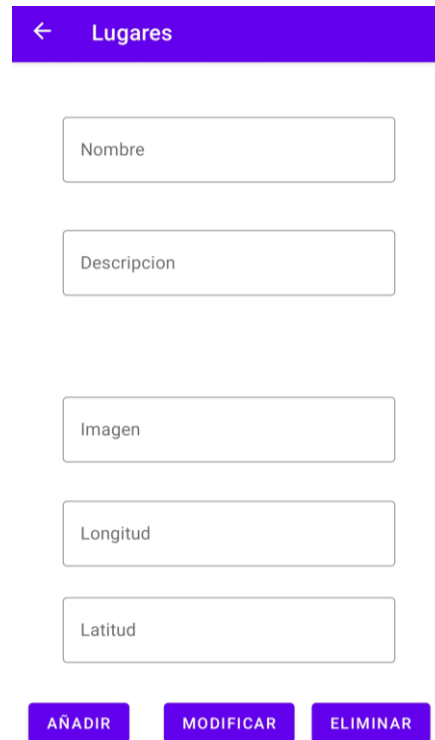


Figura 64: Gestionar lugares

A.4 Gestión de itinerarios

Para añadir itinerarios, el administrador deberá seleccionar la opción de itinerarios del menú lateral y pulsar sobre el botón + a la derecha de añadir o eliminar itinerarios, tal como se muestra en la figura 65. Se abrirá una nueva ventana, tal como se muestra en la figura 66, en la que en el caso de querer eliminar un itinerario deberá introducir el nombre del itinerario y pulsar sobre el botón de eliminar.

Para añadir un nuevo itinerario deberá introducir el nombre del itinerario, el primer lugar del itinerario y una breve descripción del itinerario, a continuación, seleccionar el botón de añadir y se abrirá una nueva ventana como se muestra en la figura 67, para que siga introduciendo lugares que desea agregar a ese itinerario, esa ventana se seguirá mostrando para que siga introduciendo lugares hasta que haya seleccionado todos lugares y pulse el botón de finalizar.

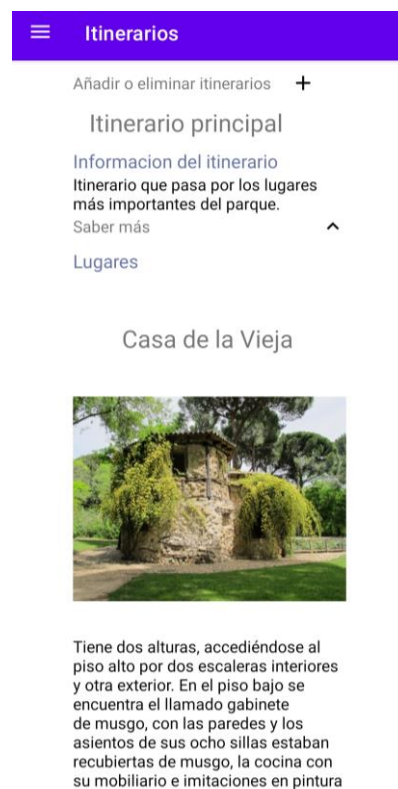


Figura 65: Mostrar itinerarios

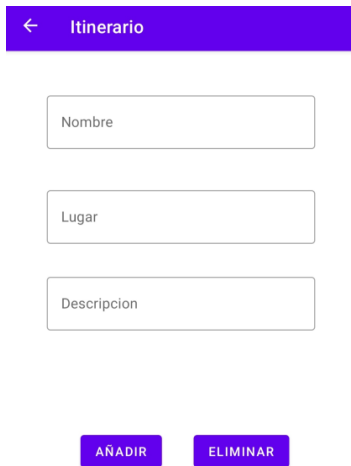


Figura 66: Gestionar itinerarios

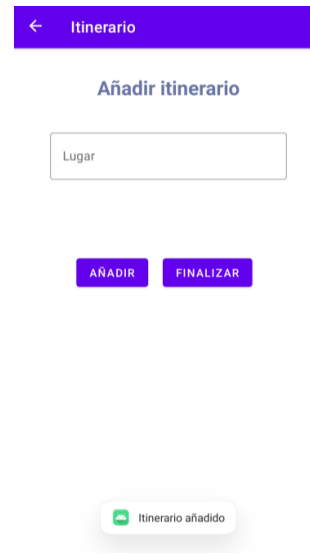


Figura 67: Añadir lugar a itinerario

A.5 Registro y Login

Para registrarse en la aplicación y poder así disfrutar de todas las funcionalidades, deberá seleccionar la opción perfil del menú lateral, se mostrará una pantalla como la de la figura 68. Pulsará sobre el botón registrar situado en la parte inferior y se abrirá una nueva ventana, tal como se muestra en la figura 69, donde deberá introducir su email, un nombre de usuario y una contraseña y pulsar sobre el botón registrar.

Para iniciar sesión en la aplicación deberá seleccionar la opción perfil del menú lateral, se mostrará una ventana como la de la figura 68. Introducirá su email y contraseña y pulsará sobre el botón de login.

☰ Perfil

Iniciar Sesion

Email

Contraseña

LOG IN

¿Todavía no tienes una cuenta? Regístrate ahora!

REGISTRAR

Figura 68: Login

← Perfil

Registro

Email

Usuario

Contraseña

REGISTRAR

Figura 69: Registro

A.6 Logout, modificación datos y eliminar usuario

Para cerrar sesión, actualizar sus datos o eliminar su cuenta, el usuario deberá seleccionar la opción de perfil del menú lateral y se mostrará una pantalla como la de la figura 70. En caso de querer cerrar sesión pulsará sobre el botón de logout, en caso de querer eliminar su cuenta seleccionará el botón de eliminar y, por último, en caso de querer modificar sus datos introducirá su nuevo nombre de usuario y contraseña y pulsará sobre el botón de actualizar.

The image shows a user profile management interface. At the top, there is a purple header with a hamburger menu icon and the text 'Perfil'. Below this, the title 'Modificar Datos' is centered. There are two input fields: the first is labeled 'Usuario' with a person icon, and the second is labeled 'Contraseña' with a lock icon and a toggle eye icon. At the bottom, there are three purple buttons: 'ELIMINAR', 'ACTUALIZAR', and 'LOG OUT'.

Figura 70: Gestionar usuario

A.7 Gestión de comentarios

Para añadir, modificar o eliminar un comentario, tendrá que haberse registrado previamente, deberá seleccionar la opción de lugares del menú lateral y se mostrará una pantalla como la de la figura 71.

En caso de querer agregar un comentario, escribirá el comentario en el recuadro situado encima del botón de añadir y pulsará sobre el botón añadir. En caso de querer borrar su comentario, seleccionará el botón de eliminar situado debajo de su comentario. Por último, en caso de querer actualizar su comentario seleccionará el botón de modificar situado debajo de su comentario y se redirigirá a una nueva pantalla, tal como se muestra en la figura 72, donde deberá introducir su comentario modificado y presionar el botón de actualizar.

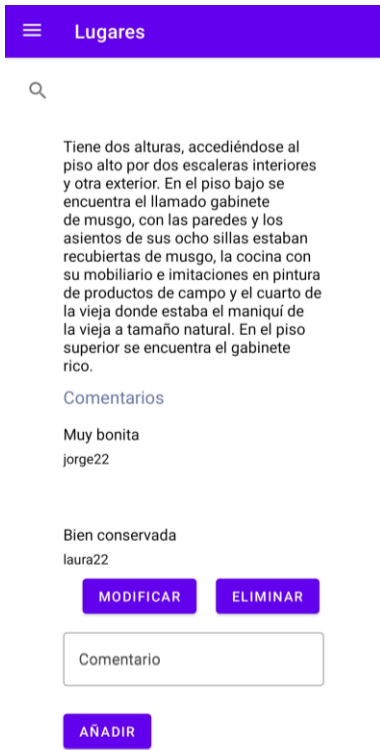


Figura 71: Gestionar comentarios

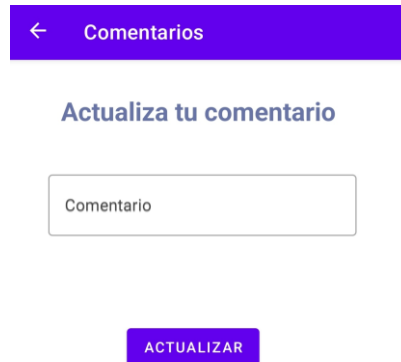


Figura 72: Actualizar comentarios

A.8 Mapa

Para ver el plano del parque deberá seleccionar la opción de mapa del menú lateral y se mostrará una pantalla como la de la figura 73. Podrá ver su ubicación actual en el mapa y los lugares de interés mostrados con un icono de color rojo, pulsando sobre cada uno de ellos obtendrá el nombre del lugar, una imagen y la opción de iniciar la ruta, tal como se muestra en la figura 74.

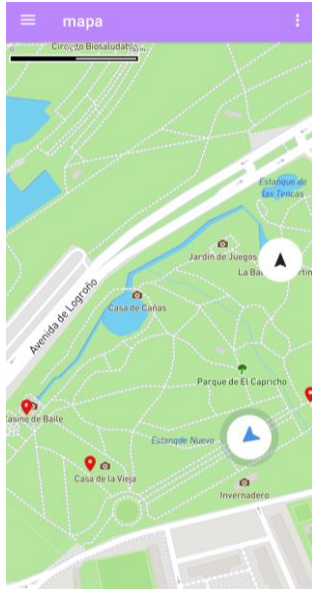


Figura 73: Ver mapa



Figura 74: Mostrar lugar en el mapa

A.9 Iniciar ruta a lugar de interés

Para iniciar una ruta a un lugar de interés deberá seleccionar la opción de mapa del menú lateral y se mostrará una pantalla como la de la figura 73. Seleccionará uno de los lugares mostrados con un icono de color rojo y obtendrá el nombre del lugar, una imagen y la opción de iniciar la ruta, tal como se muestra en la figura 74. Pulsará sobre el botón iniciar y se le mostrará la ruta paso a paso que debe seguir para ir al lugar seleccionado, tal como se muestra en la figura 75.

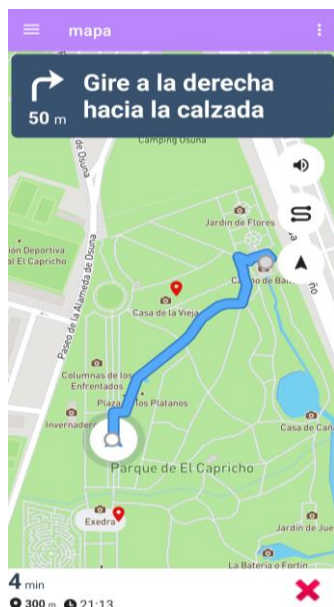


Figura 75: Iniciar ruta

A.10 Lugares de interés

Para ver el listado de lugares del parque deberá seleccionar la opción de lugares del menú lateral y se mostrará una pantalla como la de la figura 76. Podrá ver los lugares presentes en el parque con una imagen y pulsando sobre el desplegable saber más, obtendrá una descripción del lugar y ver los comentarios que tiene. Para buscar un lugar deberá introducir el nombre del lugar pulsando sobre el icono de la lupa en la parte superior, a continuación, se le mostrará el lugar tal como se puede observar en la figura 77.



Figura 76: Mostrar lugares



Figura 77: Buscar lugar

A.11 Ver itinerarios y añadir a favoritos

Para ver el listado de itinerarios del parque deberá seleccionar la opción de itinerarios del menú lateral y se mostrará una pantalla como la de la figura 78. Podrá ver una lista de itinerarios con su nombre, información y todos los lugares presentes en el itinerario.

Para añadir el itinerario a favoritos, deberá introducir el nombre que le quiere dar al itinerario y pulsar sobre el botón añadir a favoritos, tal como se muestra en la figura 79.

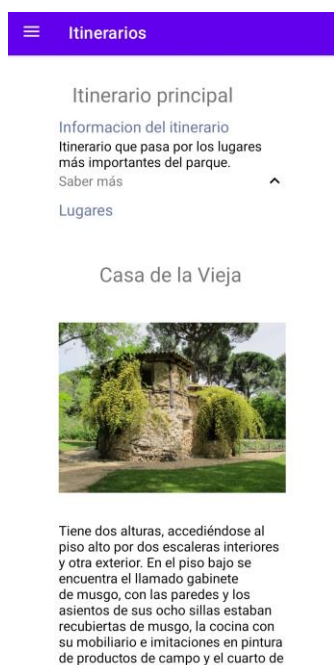


Figura 78: Mostrar itinerarios



Figura 79: Añadir itinerario a favoritos

A.12 Iniciar itinerario

Para iniciar un itinerario deberá seleccionar la opción de itinerarios del menú lateral y se mostrará una pantalla como la de la figura 78. Deberá dirigirse al final del itinerario y pulsar el botón empezar, tal como se muestra en la figura 79. A continuación, se abrirá una nueva pantalla con el mapa donde deberá pulsar el botón iniciar, tal como se muestra en la figura 80. Se mostrará la ruta paso a paso del itinerario pasando por todos los lugares, tal como se muestra en la figura 81.

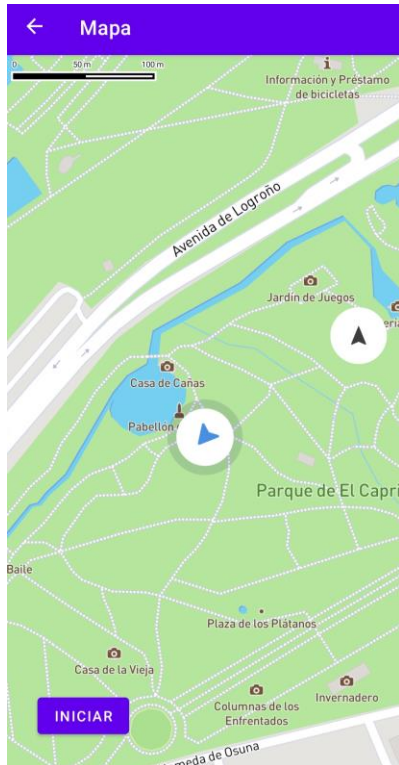


Figura 80: Iniciar itinerario

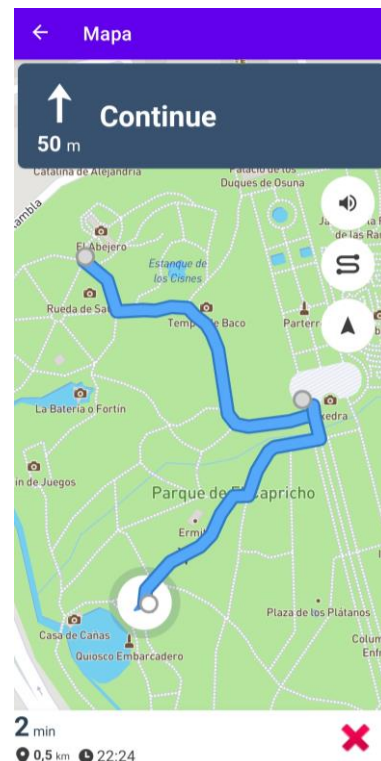


Figura 81: Ver ruta paso a paso del itinerario

A.13 Itinerarios favoritos

Para ver el listado de itinerarios favoritos del parque deberá seleccionar la opción de itinerarios favoritos del menú lateral y se mostrará una pantalla como la de la figura 82. Podrá ver su lista de itinerarios con su nombre, información y todos los lugares presentes en el itinerario. Además, en la parte inferior podrá eliminar el itinerario de su lista de favoritos pulsando sobre el botón quitar, tal como se muestra en la figura 83, o pulsar sobre el botón empezar para iniciar la ruta del itinerario, donde se le redirigirá a la pantalla del mapa donde deberá pulsar el botón iniciar, tal como se muestra en la figura 80. Se mostrará la ruta paso a paso del itinerario, tal como se muestra en la figura 81.

☰ Mis itinerarios

Mi favorito

Nombre del itinerario
Itinerario principal

Información del itinerario
Itinerario que pasa por los lugares más importantes del parque.
Saber más ▼

Lugares

Casa de la Vieja



Tiene dos alturas, accediéndose al piso alto por dos escaleras interiores y otra exterior. En el piso bajo se encuentra el llamado gabinete de musgo, con las paredes y los asientos de sus ocho sillas estaban recubiertas de musgo, la cocina con

Figura 82: Mostrar itinerarios favoritos

☰ Mis itinerarios



Bonita escultura con varios leones

El abejero



Rodeado de un gran jardín

EMPEZAR QUITAR

Figura 83: Eliminar itinerario favorito