DESARROLLO DE UN SISTEMA DE GESTIÓN BOTÁNICO SOBRE LOS RECURSOS DE EL RETIRO

TRABAJO FIN DE GRADO
CURSO 2022-2023

AUTOR
LUIS GABRIEL ROMÁN SANTILLÁN

DIRECTOR
ANTONIO SARASA CABEZUELO

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

DEVELOPMENT OF A BOTANICAL MANAGEMENT SYSTEM FOR THE RESOURCES OF EL RETIRO

FINAL DEGREE PROJECT REPORT
COMPUTER ENGINEERING DEGREE

AUTHOR
LUIS GABRIEL ROMÁN SANTILLÁN

DIRECTED BY
ANTONIO SARASA CABEZUELO

FACULTY OF COMPUTER SCIENCE
COMPLUTENSE UNIVERSITY OF MADRID

JUNIO 2023

AGRADECIMIENTOS

En primer lugar, me gustaría agradecer a mi tutor Antonio Sarasa Cabezuelo que me ha guiado y ha acompañado en esta nueva aventura, con sus consejos y sugerencias y sobre todo por su gran disponibilidad que ha mostrado.

Del mismo modo, me gustaría agradecer a mi familia su gran apoyo desde el inicio y durante la carrera hasta llegar a este punto.

Y, por último, a todas aquellas personas que he ido conociendo a lo largo de la carrera, ya que han hecho que el camino haya sido más sencillo y agradable.

Resumen

La idea del trabajo de fin de grado surge como una manera de crear una herramienta que facilite y permita tener una experiencia diferente a la tradicional de un mapa, es por ello que se ha desarrollado un sistema de gestión botánico sobre los recursos de El Retiro.

Para ello se ha tenido en cuenta que el Parque del retiro se puede visitar por medio de varias sendas que están compuestas de puntos de interés que cuentan una historia. Siguiendo esta línea se ha desarrollado una aplicación, la cual permite mostrar información junto con fotografías de los puntos y sendas.

Además, se ha diseñado funcionalidades de usuario que permiten moverse de forma libre por el parque mostrando los puntos según se va desplazando por él. Al igual que permite realizar trayectos hacia cualquier punto del parque.

Finalmente, para la implementación de este sistema se ha desarrollado una aplicación para dispositivos móviles, junto con una API que permite gestionar toda la información de la base datos del sistema.

Palabras Clave

Parque del Retiro, aplicación móvil para Android, sendas, puntos de interés, interactivo, localización en tiempo real.

Abstract

The idea of this project arises as a way to create a tool that facilitates and allows us to have a different experience from the traditional one: a map. That is why have developed a botanical management system for the resources of El Retiro.

For this take into account that The Retiro can be visited through several paths that are made up of interest points that tell a story. That is why have developed an application which allows you to display information with photographs of those points and paths.

In addition, has been designed user functions that allow users to move freely around the park, showing the points as we move through it. It also allows you to make trips to any point in the park.

Finally, for the implementation of this system, have developed an application for mobile devices, along with an API that allows managing all the information in the system database.

Keywords

Parque del Retiro, mobile application for Android, trails, Interest points, interactive, real-time location.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Capítulo 1 - Introducción

En el capítulo siguiente se explicarán las distintas razones por las que se ha seleccionado este proyecto, los objetivos de la aplicación y finalmente se describirá la estructura de la memoria junto con la metodología empleada.

1.1 Motivación

El parque del Retiro de Madrid es un espacio verde que se ubica dentro Madrid, el cual consta de 125 hectáreas y con mucha diversidad en su interior ya que alberga distintos tipos de jardines con temática variada, monumentos con gran carga histórica y también es un sitio de ocio y deporte preferido por los madrileños y turistas.

Por todo esto se ha pretendido digitalizar las sendas existentes que recorren las distintas ubicaciones del parque para que la visita sea mucho más interactiva.

Cada punto de interés cuenta con una historia y una serie de características, al igual que las sendas. Sin embargo, es más interactivo si durante la visita se puede ir relacionando los puntos con toda la información que le rodea.

Por otra parte, al usuario le puede surgir la necesidad de ir a algún punto que no esté contemplado dentro de las sendas, al igual que encontrar la fuente más cercana si estás haciendo deporte, por ese motivo se ha desarrollado una opción que ofrece una navegación libre.

Todo lo mencionado anteriormente es lo que ha llevado a diseñar e implementar una aplicación que recoge todas estas funcionalidades.

Se puede acceder al código desarrollado en el siguiente enlace, disponible solo para el personal de la Universidad Complutense de Madrid:

https://drive.google.com/drive/folders/1iGrCpHQxknzUa4nfUUOJUyoD2ZKgAhu4

1.2 Objetivos

El objetivo de este proyecto es el de hacer que el usuario experimente de forma más interactiva la visita por el parque del Retiro, además de la consulta de información referente al parque, para ello podemos desarrollarlo a través de los siguientes objetivos más específicos:

- Facilitar al usuario toda la información de los distintos puntos de interés del parque.
- Proporcionar una lista de sendas con información que están previamente definidas.
- Permitir la visualización de la ubicación en tiempo real del usuario.
- Seleccionar dentro del mapa el punto al cual el usuario quiere desplazarse.
- Desarrollar una función que proporcione información durante un trayecto hacia un punto de interés.
- Facilitar la creación de los puntos que pueden formar parte de las diferentes sendas.
- Diseñar una estructura para almacenar toda la información usando una base de datos SQL gestionada con phpmyAdmin.
- Implementar una interfaz que permita la comunicación con el servidor.

1.3 Estructura de la memoria

La memoria del proyecto se estructura de la siguiente manera.

- En el capítulo 1 se describe la motivación que ha llevado a la selección y desarrollo de este proyecto.
- En el capítulo 2 se mencionan algunas aplicaciones con funcionalidades similares a las de este proyecto.
- En el capítulo 3 aparecen las distintas herramientas, tecnologías y sus implementaciones dentro de la aplicación.
- En el capítulo 4 se hace referencia a los casos de uso del proyecto.
- En el capítulo 5 se explica la arquitectura de la aplicación además de cómo se realiza la persistencia de información, analizando en detalle cada elemento que componen la base de datos.
- En el capítulo 6 se describe el diseño, la funcionalidad y la implementación de la API de comunicación de la aplicación.
- En el capítulo 7 se plantea el trabajo futuro para desarrollar.

1.4 Metodología

El proyecto se ha realizado de acuerdo con un plan de proyecto y una organización que se describe en el siguiente apartado.

1.4.1 Fases del proyecto

Para llevar a cabo el proyecto, se hizo una planificación que incluyó una investigación y planteamiento de requisitos necesarios de la aplicación. Además, se incorporó el desarrollo de la gestión de usuarios, sistema del administrador, los servicios básicos de los visitantes y finalmente la gestión de las sendas. La Tabla 1 muestra la duración de cada etapa, y por medio de la Figura 1 se muestra el diagrama de Gantt del proyecto.

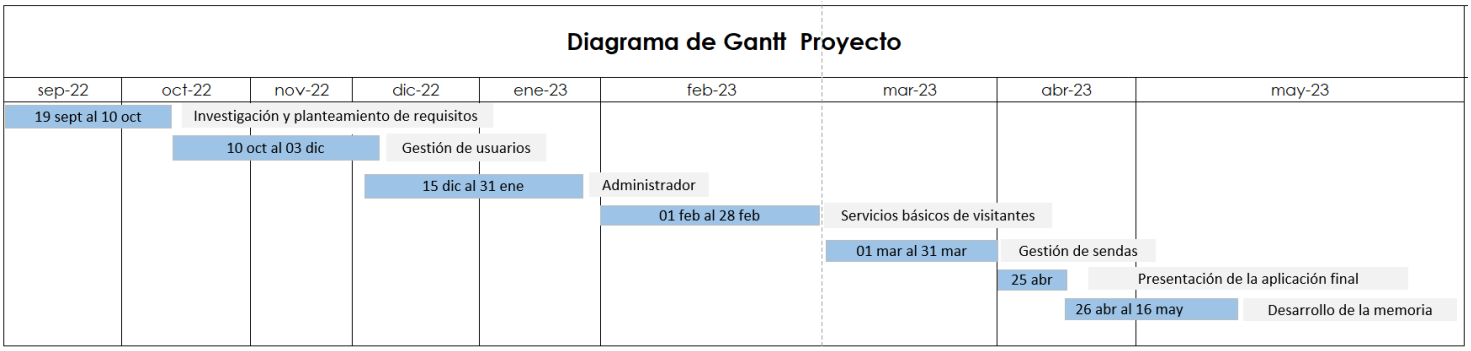


Figura 1. DIAGRAMA DE GANTT DEL PROYECTO

Etapa	Duración
Investigación y planteamiento de requisitos	19 de septiembre - 10 de octubre
Gestión de usuarios	10 de octubre - 03 de diciembre
Administrador	15 diciembre - 31 de enero
Servicios básicos de visitantes	01 de febrero - 28 de febrero
Gestión de sendas	01 de marzo -31 de marzo
Presentación de la aplicación final	25 de abril
Desarrollo de la memoria	26 de abril - 16 de mayo

Tabla 1. ETAPAS DEL DESARROLLO DEL PROYECTO

A continuación, se explica cada una de las fases que forman el proyecto.

Etapa 1: Investigación y planteamiento de requisitos

Esta primera etapa consistió en la recopilación de necesidades que presentaban los usuarios del parque del Retiro de Madrid, para ello se hizo una investigación de la distinta información recopilada y además basándose en las distintas herramientas que actualmente existen.

Etapa 2: Gestión de usuarios

En la segunda etapa, se desarrolló el diseño de todas las funcionalidades relacionadas con la gestión de usuarios, como el diseño de las tablas de usuarios de la base de datos y las funciones que permite modificar, crear y eliminar, junto con las funcionalidades como: iniciar sesión, cerrar sesión.

Etapa 3: Administrador

La etapa tres consistió en la implementación de las funcionalidades vistas en las que el administrador del sistema es el protagonista, referente a los puntos de interés y de las sendas.

Etapa 4: Servicios básicos de visitantes

En la cuarta etapa, se implementó todas las funcionalidades y vistas que hacen que la aplicación sea interactiva y en las que el usuario es el protagonista.

Etapa 5: Gestión de sendas

En la quinta etapa, se integró el uso de la localización en tiempo real dentro del proyecto, además de las funcionalidades que interactúan con ellas.

Etapa 6: Presentación de la aplicación final

En la sexta fase se realizó la presentación de la aplicación final a través de Google meet con el tutor. En donde se realizó un recorrido completo por las diferentes funcionalidades que presenta la aplicación.

1.4.2 Organización del trabajo

Durante la creación del proyecto se ha mantenido una metodología de trabajo iterativa incremental. Las distintas iteraciones han presentado una duración aproximada de un mes, seguida de una reunión en la que se mostraban las nuevas implementaciones y se recibía un feedback para ser corregido de cara a la siguiente reunión.

También se utilizó GitHub como herramienta de control de versiones que, debido al esquema de ramas que se ha implantado, permite desarrollar las nuevas funcionalidades por módulos.

Chapter 1: Introduction

The following chapter explains the different reasons why this Project has been chosen, the main objectives of the Application, the structure of the report and finally methodology used.

1.1 Motivation

The Retiro is a green space that is located within Madrid, which consists of 125 hectares with a lot of diversity inside because it has different types of gardens with varied themes, monuments with great historical load and It is also a place of leisure and play sports preferred by locals and tourists.

For all this, it has been intended to digitize the existing paths that run through the different locations of the park so that the visit is much more interactive.

Each interest point has a story and a series of features, as well as the paths. However, it is more interactive if during the visit you can relate the points with all the information that surrounds you.

On the other hand, the user may need to go to some point that is not contemplated within the paths or even find the nearest fountain if you are playing sports. For that reason, an option has been developed that offers free navigation.

Everything mentioned above is what has led to design and implement an Application that includes all these functionalities.

You can access the code developed in the following link, available only to the staff of the Complutense University of Madrid:

https://drive.google.com/drive/folders/1iGrCpHQxknzUa4nfUUOJUyoD2ZKgAhu4

1.2 Objectives

The main objective of this Project is to make the user visit the Retiro in a more interactive way in addition to being able to consult information regarding the Park. To do this, we can develop It through the following more specific objectives:

- Provide the user all the information related with the different interest points in the park.
- Provide a list of paths with information that are previously defined.
- Allow the visualization of the user's location in real time.
- Select within the map the point to which the user wants to move.
- Develop a function that provides information on a journey to an interest point.
- Facilitate the creation of points that can form part of the different paths.
- Design a structure to store all the information using a SQL database that is managed with phpmyAdmin.
- Implement an interface that allows communication with the server.

1.3 Structure of the report

The Project report is structured as follows:

- Chapter 1 describes the motivation that led to the selection and development of this Project.
- Chapter 2 mentions some Applications with similar functions to those that appear in this Project.
- Chapter 3 shows the different tools, technologies, and their implementations within the Application.
- Chapter 4 refers to the use cases of the Project.
- Chapter 5 explains the architecture of the Application and how information persistence is carried out, analyzing in detail each element that makes up the database.
- Chapter 6 describes the design, functionality, and the implementation of the Application communication API.

- Chapter 7 outlines the future work to develop.

1.4 Methodology

The Project has been carried out in accordance with a Project plan and an organization that is described in the following section.

1.4.1 Project phases

To carry out the Project, a planning was made including an investigation and an approach to the necessary requirements of the Application. In addition, the development of user management, the administrator system, the basic services and finally the management of the trails were incorporated. Table 1 shows the timeline of each stage and through Figure 1, the Gantt chart of the Project is shown.

Each of the phases that forms the Project is explained below.

Phase 1: Research and requirements approach

This first phase consisted in collecting the needs presented by the users of the Retiro in Madrid. For this, an investigation was made of the different information collected based on the different tools that currently exist.

Phase 2: User management

In the second phase, the design of all the functionalities related to user management was developed, such as the design of the user tables of the database and the functions that allow modify, create and delete, with the functionalities such as: log in and log out.

Phase 3: Administrator

Phase three consisted of the implementation of the functionalities in which the system administrator is the protagonist, referring to the interest points and paths.

Phase 4: Basic visitor services.

In the fourth phase, all the functionalities and views that make the Application interactive and in which the user is the protagonist were implemented.

Phase 5: Paths management

In the fifth phase, the use of real time location was integrated into the Project in addition to the functionalities that interact with them.

Phase 6: Presentation of the final Application

In the sixth phase, the final Application was presented through Google Meet with the teacher. In that presentation a complete tour of the different functionalities that the Application presents was carried out.

1.4.2 Work organization

During the creation of the Project, an incremental iterative work methodology has been maintained. The different iterations have lasted approximately one month, followed by a meeting in which the new implementations were shown, and feedback was received to be corrected for the next meeting.

GitHub was also used as a version control tool that, due to the branch scheme that has been implemented, allows the development of new functionalities by modules.

Capítulo 2 - Estado del arte

En el siguiente capítulo se muestra una serie de aplicaciones que se encuentran actualmente en el mercado para dispositivos ANDROID, IOS, web y poseen características similares a mi aplicación.

Citymapper: [1] Es una aplicación disponible en Android e ios y web desarrollada por citymapperlimited en 2011 cuyo objetivo principal es usar la localización en tiempo real y crear recorridos de un punto a otro, para ello se tiene que poner un punto de inicio y otro final. Y muestra las distintas posibilidades de trayecto que se pueden seguir.

ArroundMe: [2] Aplicación desarrollada por Flying Code en 2008 para Android e ios, cuyo objetivo es mostrar las diferentes ubicaciones que están cerca de la ubicación del usuario, las principales características son:

- Permite buscar los distintos tipos en los que se agrupan las ubicaciones disponibles, por ejemplo: farmacias, cines, restaurantes.
- Muestra una lista de elementos que cumplen el tipo seleccionado y al seleccionar muestra información relevante como: el número de contacto, acceso a la web e incluso permite realizar la ruta hasta la ubicación del punto seleccionado.
- Se da la opción de guardar en favoritos.
- Posee un apartado en el que el usuario puede colocar comentarios que se usan para la mejora del punto que se está consultando.

ZooAquarium: [3] Es la aplicación oficial del Zoo Aquarium de Madrid, está desarrollada por Mobil para dispositivos Android e IOs. Las principales características que posee son las siguientes:

- Posee una navegación por medio de un menú lateral, que permite el movimiento por toda la aplicación.
- Muestra un mapa en el que vienen prefijadas todas las ubicaciones de los puntos de interés del zoológico siendo estos seleccionables.

- Tiene fichas en las que muestra toda la información referente al punto seleccionado, además de una opción que te permite desplegar el mapa con la ubicación en tiempo real y remarcando la ubicación del punto seleccionado.

ParqueRetiroMadrid: [4] Es la aplicación oficial del Retiro de Madrid y permite una navegación muy intuitiva, ya que existen 3 apartados principales:

- Imprescindibles: es el encargado de recoger todos aquellos puntos que son interesante visitar durante tu visita por El Retiro, acompañado de una breve descripción y con la posibilidad de ver la ubicación en el mapa.
- Explora: recoge toda la información en familias genéricas y va acotando según navegas.
- Mapa: Muestra un mapa estático del Parque del Retiro en el coloca todos los puntos de interés dentro del parque.

Arboles del retiro: [4] Aplicación Android que recoge toda la información de la botánica del Parque del Retiro y muestra la lista de resultados según las preferencias que haya seleccionado el usuario en el menú principal. Una vez en el ítem de la lista, muestra una ficha que con una descripción y fotos del elemento.

Madrid Móvil:[5] Aplicación oficial de Madrid que permite interactuar con varios aspectos, pero en este caso cabe destacar el aparatado de fuentes y aseos, ya que utiliza la ubicación de estas y da la posibilidad a través de Google Maps de trazar una ruta e iniciar un recorrido hasta el punto seleccionado.

Tras la investigación y el análisis de las aplicaciones anteriormente nombradas, se han recogido aquellas funcionalidades más relevantes y sobre todo que mantengan el objetivo de este trabajo de fin de grado, que es el de hacer que las visitas de las personas sea lo más interactivas posible.

Para ello se tiene en cuenta una serie de funcionalidades principales que son: La localización en tiempo real, el manejo con datos sobre los distintos puntos y sobre todo que sea intuitiva para el usuario.

Capítulo 3 - Tecnologías y herramientas empleadas

3.1 Kotlin

[6]Es un lenguaje de programación creado en 2010 por Intellij IDEA de la mano de Dmitry Jemerov y más adelante en 2012 pasó a ser open source.

Sus principales características son que admite la programación funcional y orientada a objetos, proporciona una sintaxis similar a la de otros lenguajes: c#, java y puede trabajar sobre la JVM,JavaScript o sobre su LLVM

Este lenguaje de programación está enfocado en el desarrollo de aplicaciones móvil de forma nativa y estas son las ventajas que tiene sobre Java:

- Simple y moderno, lo que facilita el aprendizaje
- Solventa problemas

3.2 PHP

[7] [8] Es un lenguaje de programación creado en 1994 por Rasmus Lerdorf. Es código abierto que se encuentra en constante perfeccionamiento. Se ejecuta en el lado del servidor y se encarga de interpretar todas las sentencias para la conexión con la base de datos permitiendo el acceso a los datos.

Ventajas de usar php:

- Aprendizaje del lenguaje simplificado, ya que existe una gran cantidad de material didáctico que es proporcionado y además posee una comunidad de usuarios.
- Tiene una gran compatibilidad con sistemas de base de datos: Esto es importante ya que son piezas fundamentales dentro de la estructura de un sistema porque vamos a acceder de manera recurrente a la información que contienen.

3.3 Android Studio

[9] Es un entorno de desarrollo que fue creado en 2013 y se enfoca en el desarrollo de aplicaciones Android y sus principales características son:

- Sistema de compilación flexible
- Permite trabajar de forma fluida ya que permite renderizar los layouts en tiempo real.
- Posee herramientas como un gestor de git, un emulador que permite simular el funcionamiento del código.
- Compatibilidad con otros lenguajes como java, C++, NDK

3.4 PhpMyAdmin

[10]Es una herramienta creada en 1998, open source basada en php que sirve para administrar bases de datos relacionales como MySQL. Permite el acceso a la información de forma sencilla, ya que la distribuye en tablas previamente estructuradas de manera eficiente.

Además, se pueden ejecutar comandos SQL de búsqueda, inserción y eliminación de los datos de estas.

3.5 Vscode

[11] Es una herramienta open source, multiplataforma que se utiliza como editor de código fuente desarrollada por Microsoft.

Sus principales características son:

- Multiplataforma: Se puede utilizar para el desarrollo en distintos lenguajes con diversos objetivos, siendo muy polifacético.
- Uso de control de versiones ya que tiene compatibilidad con Git, permitiendo revisar las diferencias entre commits y archivos.

- Extensiones que permiten obtener un mayor grado de personalización a nivel de herramientas para el desarrollo de aplicaciones.

3.6 GitHub

[12]Es un portal creado en 2008 que permite alojar el código de las aplicaciones de forma remota, y tener acceso desde cualquier ubicación

Las principales características son:

- Administrar versiones del proyecto, permitiendo un control sobre los distintos cambios que se realicen a lo largo del desarrollo.
- Trabajar en equipo, ya que se permite las actualizaciones dentro del proyecto de forma simultanea

3.7 Mapbox

[13] [14]Es una plataforma que se basa en software libre creada en 2010, que provee de localización en tiempo real, además de otros productos como Mapas, navegación, Atlas entre otros y ofrece una gran variedad de estilos, tipos que pueden ser completamente personalizados.

- El funcionamiento se realiza por medio de llamadas a APIs o agregar SDKs al proyecto dependiendo del entorno en el que se esté desarrollando.
- La incorporación dentro de un proyecto es bastante intuitiva, ya que proporciona varios ejemplos, documentación y guías para ir agregándolo.
- Está disponible para la incorporación en proyectos Android como ios.

3.8 Postman

[15] Es una plataforma gratuita que ofrece la posibilidad de desarrollar y crear peticiones http para poder probar y verificar que el funcionamiento de una API es el deseado.

Sus principales ventajas son:

- Se puede usar de forma online o por medio de su aplicación de escritorio.
- Ofrece tutoriales, documentación y ejemplos con los que empezar a trabajar.
- La interfaz de usuario es bastante intuitiva lo que supone una gran ventaja para trabajar con ella.
- Tiene varios métodos y reconoce varios tipos de formatos para empaquetar los datos

3.9 Xampp

[16]Es una herramienta de software libre creado por Apache friends y cuenta con varias herramientas: un servidor, una base de datos entre otras que permite alojar y probar de forma local los datos de un desarrollo.

Sus principales ventajas son:

- Viene con todas las herramientas preconfiguradas y listas para ser utilizadas.
- Se puede utilizar sin necesidad de internet ya que se alojan en tu propio ordenador
- Su instalación es muy sencilla, ya que solo es descargar un ejecutable.

3.10 X10Hosting

[17]Es un servicio de alojamiento web gratuito en el que gracias a sus servidores se pueden alojar aplicaciones web, bases de datos en un servidor. Cuenta con una serie de características:

- Ancho de banda ilimitado.
- Almacenamiento en SSD, permite que la experiencia sea rápida.
- Permite conexiones Remotas.
- Muy intuitiva para trabajar ya que dispone de un gestor de ficheros.

Capítulo 4 - Especificación de requisitos

En este apartado se van a explicar los diferentes requisitos que actúan en la aplicación y para facilitar su comprensión se van a describir los diferentes actores que intervienen.

4.1 Actores

- Usuario sin registrar: Son aquellos actores que no se han registrado y por lo que tienen acceso a las funciones básicas de la aplicación.
- Usuario registrado: Son aquellos que se han registrado y por lo tanto tienen acceso a todas las funcionalidades que ofrece la aplicación.
- Administrador: Es el usuario que tienes los permisos pertinentes para gestionar toda la información con la que trabaja la aplicación.

La funcionalidad de la aplicación se ha agrupado en los siguientes módulos:

- Gestión de usuarios
- Gestión del sistema
- Servicios básicos de visitantes
- Gestión de rutas

4.2 Gestión de usuarios

En este módulo se ha concentrado toda la funcionalidad relacionada con el manejo de la información de los datos del usuario durante todo el ciclo útil de la aplicación. La Figura 2 contiene todo este módulo.

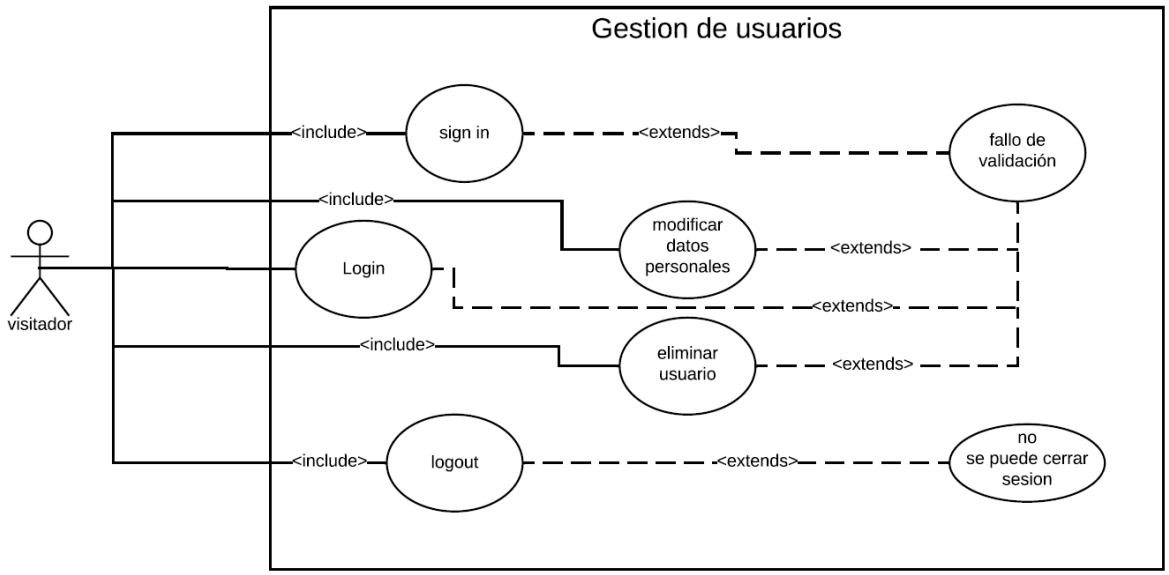


Figura 2. DIAGRAMA DE CASO DE USO GESTIÓN DE USUARIOS

A continuación, se describen los casos de usos pertenecientes a estes módulo.

Requisito	**Registro usuario**	
Identificador	1.1	
Prioridad	Alta	
Precondición	NA	
Descripción	Los Visitadores puede decidir si registrarse	
Entrada	Nombre, Apellidos, Nombre Usuario, Contraseña, Email, Edad	
Salida	NA	
Secuencia	**Paso**	**Acción**
	1	El usuario abre la aplicación, se le muestra en la pantalla principal la opción de registrarse o acceder
	2	Se muestra un formulario que recoge los distintos datos
	3	Se completa la información y se presiona “Crear”
	4	Se validan todos los campos
	5	El sistema lleva a la pantalla principal
	6	Se muestra un mensaje de bienvenida, explicando las funcionalidades.
Postcondición	Se ha registrado un usuario y pasa a reconocerse como visitador	
Excepciones	**Paso**	**Acción**
	3	En caso de fallar la validación de contraseña se mostrará un mensaje indicando el tipo de error
	3	En caso de fallar la validación de email se mostrará un mensaje indicando el tipo de error
	3	Si el nombre de usuario ya existe, se comunica a través de un mensaje
Comentarios	NA	
Actores	Usuario sin registrar, Admin	

Tabla 2. CASO DE USO REGISTRAR VISITADOR

Requisito	**Login**	
Identificador	1.2	
Prioridad	Alta	
Precondición	El visitador debe haberse registrado previamente	
Descripción	Los visitadores que se hayan registrado pueden iniciar sesión para poder realizar todas las acciones	
Entrada	Correo electrónico y contraseña	
Salida	Mensaje de Bienvenida	
Secuencia	**Paso**	**Acción**
	1	El visitador accede a la aplicación y en la ventana de bienvenida selecciona “Iniciar sesión”
	2	Se muestra un formulario en el que se pide la información necesaria para iniciar sesión
	3	Tras rellenar los campos se hace click “comenzar”
	4	El sistema valida toda la información
	5	Se muestra la pantalla inicial con todas las opciones
Postcondición	EL visitador ha iniciado sesión	
Excepciones	**Paso**	**Acción**
	4	La validación no se ha podido realizar por ende no se inicia sesión
Comentarios	NA	
Actores	Usuario Registrado, Admin	

Tabla 3. CASO DE USO LOGIN

Requisito	**Logout**	
Identificador	1.3	
Prioridad	Alta	
Precondición	El visitador ha de haber iniciado sesión	
Descripción	Los visitadores pueden salir de la sesión iniciada o poder navegar por la aplicación	
Entrada	Correo electrónico, id usuario	
Salida	Mensaje de despedida	
Secuencia	**Paso**	**Acción**
	1	El visitador accede a la aplicación, se muestra en la pantalla principal la opción de “cerrar sesión”
	2	El sistema cierra la sesión y lleva a la pantalla de inicio
Postcondición	Se ha cerrado la sesión	
Excepciones	**Paso**	**Acción**
	2	El sistema no puede cerrar sesión
Comentarios	NA	
Actores	Visitador registrado, admin	

Tabla 4. CASO DE USO LOGOUT

Requisito	**Eliminar usuario**	
Identificador	1.4	
Prioridad	Baja	
Precondición	El usuario ha de estar registrado y haber iniciado sesión en el sistema	
Descripción	Se permite que el usuario se de baja si ya no quiere usar la aplicación completa	
Entrada	Id usuario, correo electrónico	
Salida	Mensaje de Despedida	
Secuencia	**Paso**	**Acción**
	1	El usuario accede a la aplicación, se muestra la pantalla principal y dentro de configuración se hace click en “Eliminar cuenta”
	2	El sistema muestra en la pantalla un formulario en el cual se pide la contraseña
	3	El visitador introduce la contraseña y se verifica
	4	El sistema pide la confirmación
	5	Se borra el visitador del sistema
	6	El sistema lleva a la pantalla principal
Postcondición	Se elimina la cuenta del visitador del sistema	
Excepciones	**Paso**	**Acción**
	3	Muestra a través de un mensaje que la información no es correcta y pide que se vuelva a introducir
	4	Si no se pulsa la confirmación la corta el flujo
Comentarios	NA	
Actores	Visitador	

Tabla 5. CASO DE USO ELIMINAR USUARIO

Requisito	**Modificar datos personales**	
Identificador	1.5	
Prioridad	Baja	
Precondición	El usuario ha de estar registrado y haber iniciado sesión en el sistema	
Descripción	Se permite que el usuario realizar modificaciones de su información personal	
Entrada	Id usuario, correo electrónico	
Salida	Mensaje de cambios realizados	
Secuencia	**Paso**	**Acción**
	1	El usuario accede a la aplicación, se muestra la pantalla principal y dentro de configuración se hace click en “Modificar datos”
	2	El sistema muestra en la pantalla todos los datos del usuario que pueden ser modificados
	3	El sistema pide la confirmación
	4	Se arroja un mensaje
	5	Se recarga la página de datos personales
Postcondición	Los nuevos datos son guardados dentro del sistema	
Excepciones	**Paso**	**Acción**
	3	Se corta el flujo de la acción y se carga la pantalla de datos personales
Comentarios	NA	
Actores	Visitador	

Tabla 6. CASO DE USO MODIFICAR DATOS

4.3 Gestión del sistema

Este módulo concentra todas las funcionalidades a nivel de administrador en las que se realizan modificaciones sobre los datos del sistema y por lo tanto alteran al resto del sistema. En la Figura 3 se muestra el diagrama de los casos de uso de gestión del sistema correspondiente a este módulo.

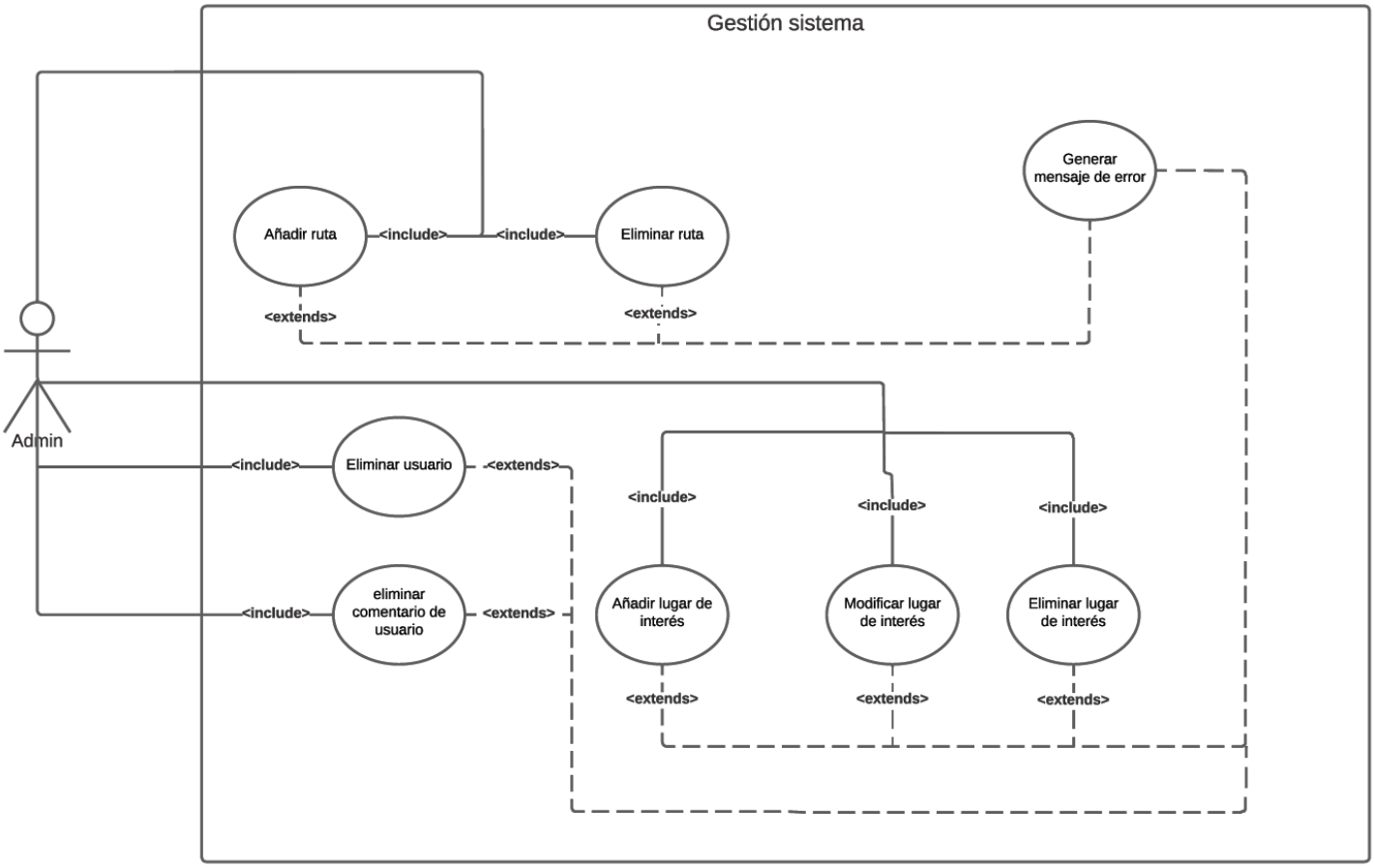


Figura 3. DIAGRAMA DE CASOS DE USO GESTIÓN DEL SISTEMA

<table>
<tr><td>Requisito</td><td colspan="3">Añadir ruta</td></tr>
<tr><td>Identificador</td><td colspan="3">4.1</td></tr>
<tr><td>Prioridad</td><td colspan="3">Alta</td></tr>
<tr><td>Precondición</td><td colspan="3">El Administrador debe haber iniciado sesión en la aplicación.</td></tr>
<tr><td>Descripción</td><td colspan="3">El Administrador puede agregar nuevas rutas a los itinerarios cuando sea necesario</td></tr>
<tr><td>Entrada</td><td colspan="3">IdAdministrador, idRuta</td></tr>
<tr><td>Salida</td><td colspan="3">NA</td></tr>
<tr><td rowspan="5">Secuencia</td><td>Paso</td><td colspan="2">Acción</td></tr>
<tr><td>1</td><td colspan="2">Se accede a la aplicación en modo administrador</td></tr>
<tr><td>2</td><td colspan="2">El administrador selecciona la opción de agregar Ruta.</td></tr>
<tr><td>3</td><td colspan="2">Se despliega un formulario que permite recopilar toda la información necesaria para crear la nueva ruta</td></tr>
<tr><td>4</td><td colspan="2">Guarda toda la información</td></tr>
<tr><td>Postcondición</td><td colspan="3">Agregar nueva información dentro del sistema</td></tr>
<tr><td rowspan="2">Excepciones</td><td>Paso</td><td colspan="2">Acción</td></tr>
<tr><td>3</td><td colspan="2">En caso de que la comprobación de datos falle se informa por medio de un mensaje al administrador</td></tr>
<tr><td>Comentarios</td><td colspan="3">NA</td></tr>
<tr><td>Actores</td><td colspan="3">administrador</td></tr>
</table>

Tabla 7. CASO DE USO AÑADIR RUTA

Requisito	**Eliminar ruta**	
Identificador	4.2	
Prioridad	Media	
Precondición	El Administrador debe haber iniciado sesión en la aplicación.	
Descripción	El Administrador puede eliminar nuevas rutas a los itinerarios cuando sea necesario	
Entrada	IdAdministrador, idRuta	
Salida	NA	
Secuencia	**Paso**	**Acción**
	1	Se accede a la aplicación en modo administrador
	2	El administrador selecciona la opción de eliminar Ruta.
	3	Se lanza un mensaje en el que el administrador debe confirmar la acción
	4	Guarda toda la información
Postcondición	Eliminar ruta del sistema	
Excepciones	**Paso**	**Acción**
	3	Si no se confirma en el mensaje, la acción se corta y no se realizan los cambios.
Comentarios	NA	
Actores	administrador	

Tabla 8. CASO DE USO ELIMINAR RUTA

Requisito	Eliminar usuario	
Identificador	4.3	
Prioridad	Media	
Precondición	El Administrador debe haber iniciado sesión en la aplicación.	
Descripción	El Administrador puede eliminar los usuarios que se encuentran registrados	
Entrada	IdAdministrador, idvisitador	
Salida	NA	
Secuencia	**Paso**	**Acción**
	1	Se accede a la aplicación en modo administrador
	2	El administrador selecciona la opción de eliminar usuario.
	3	Se lanza un apartado en el que se pide el motivo de la eliminación
	4	Se lanza un mensaje en el que el administrador debe confirmar la acción
	5	Guarda toda la información
Postcondición	Eliminar usuario del sistema	
Excepciones	**Paso**	**Acción**
	3	Si el apartado no se rellena no permite continuar
	4	Si no se confirma en el mensaje la acción se corta y se informa de un fallo
Comentarios	NA	
Actores	administrador	

Tabla 9. CASO DE USO ELIMINAR USUARIO

Requisito	**Eliminar comentario usuario**	
Identificador	4.4	
Prioridad	Media	
Precondición	El Administrador debe haber iniciado sesión en la aplicación.	
Descripción	El Administrador puede eliminar los comentarios realizados por usuarios que se encuentran registrados	
Entrada	IdComentario, idUsuario, idAdministrador	
Salida	NA	
Secuencia	**Paso**	**Acción**
	1	Se accede a la aplicación en modo administrador
	2	El administrador selecciona opciones de Foro.
	3	Seleccionará la opción de eliminar comentario foro
	4	Se lanza un apartado en el que se pide el motivo de la eliminación
	5	Se lanza un mensaje en el que el administrador debe confirmar la acción
	6	Guarda toda la información
Postcondición	El administrador habrá eliminado un comentario realizado por el usuario del sistema	
Excepciones	**Paso**	**Acción**
	4	En caso de no poner el motivo el flujo se corta, cancelando la operación
Comentarios	NA	
Actores	Administrador	

Tabla 10. CASO DE USO ELIMINAR COMENTARIOS USUARIO

Requisito	**Añadir lugar de interés**	
Identificador	4.5	
Prioridad	Media	
Precondición	El Administrador debe haber iniciado sesión en la aplicación.	
Descripción	El Administrador puede añadir los lugares al sistema	
Entrada	idLugar, idAdministrador	
Salida	NA	
Secuencia	**Paso**	**Acción**
	1	Se accede a la aplicación en modo administrador
	2	El administrador selecciona “opciones de lugares”.
	3	Seleccionará la opción de agregar sitio de interés
	4	Agrega toda la información por medio de un formulario
	5	Finaliza el registro
Postcondición	Se registrará un nuevo sitio al sistema	
Excepciones	**Paso**	**Acción**
	4	Si la validación de los campos falla, se comunica por medio de un mensaje
Comentarios	NA	
Actores	Administrador	

Tabla 11.CASO DE USO AÑADIR LUGAR DE INTERES

Requisito	**Modificar lugar de interés**	
Identificador	4.5	
Prioridad	Media	
Precondición	El Administrador debe haber iniciado sesión en la aplicación.	
Descripción	El Administrador puede modificar los lugares que se encuentran en el sistema	
Entrada	IdLugar	
Salida	NA	
Secuencia	**Paso**	**Acción**
	1	Se accede a la aplicación en modo administrador
	2	El administrador selecciona “opciones de lugares”.
	3	Seleccionará la opción de modificar sitio de interés
	4	Realizar las modificaciones de la información en los campos pertinentes
	5	Finaliza la modificación
Postcondición	Se modificará el registro dentro del sistema	
Excepciones	**Paso**	**Acción**
	4	En caso de que las modificaciones no cumplan las condiciones se muestra un mensaje informándolo
Comentarios	NA	
Actores	Administrador	

Tabla 12. CASO DE USO MODIFICAR LUGAR DE INTERES

Requisito	**Eliminar lugar de interés**	
Identificador	4.6	
Prioridad	Media	
Precondición	El Administrador debe haber iniciado sesión en la aplicación.	
Descripción	El Administrador puede eliminar los lugares que se encuentran en el sistema	
Entrada	IdLugar	
Salida	NA	
Secuencia	**Paso**	**Acción**
	1	Se accede a la aplicación en modo administrador
	2	El administrador selecciona “opciones de lugares”.
	3	Seleccionará la opción de eliminar sitio de interés
	4	Se lanza un mensaje de confirmación
	5	Finaliza la eliminación del lugar
Postcondición	Se elimina el lugar dentro del sistema	
Excepciones	**Paso**	**Acción**
	4	En caso de que la eliminación no se realice se muestra un mensaje informándolo
Comentarios	NA	
Actores	Administrador	

Tabla 13. CASO DE USO ELIMINAR LUGAR DE INTERÉS

4.4 Servicios básicos de visitantes

Este módulo recoge aquellas funcionalidades por las cuales el usuario puede obtener información, procesarla y finalmente mostrarla para que se pueda interactuar con ella de forma dinámica. En la Figura 4 se muestra el diagrama de casos de uso de los servicios de visitantes.

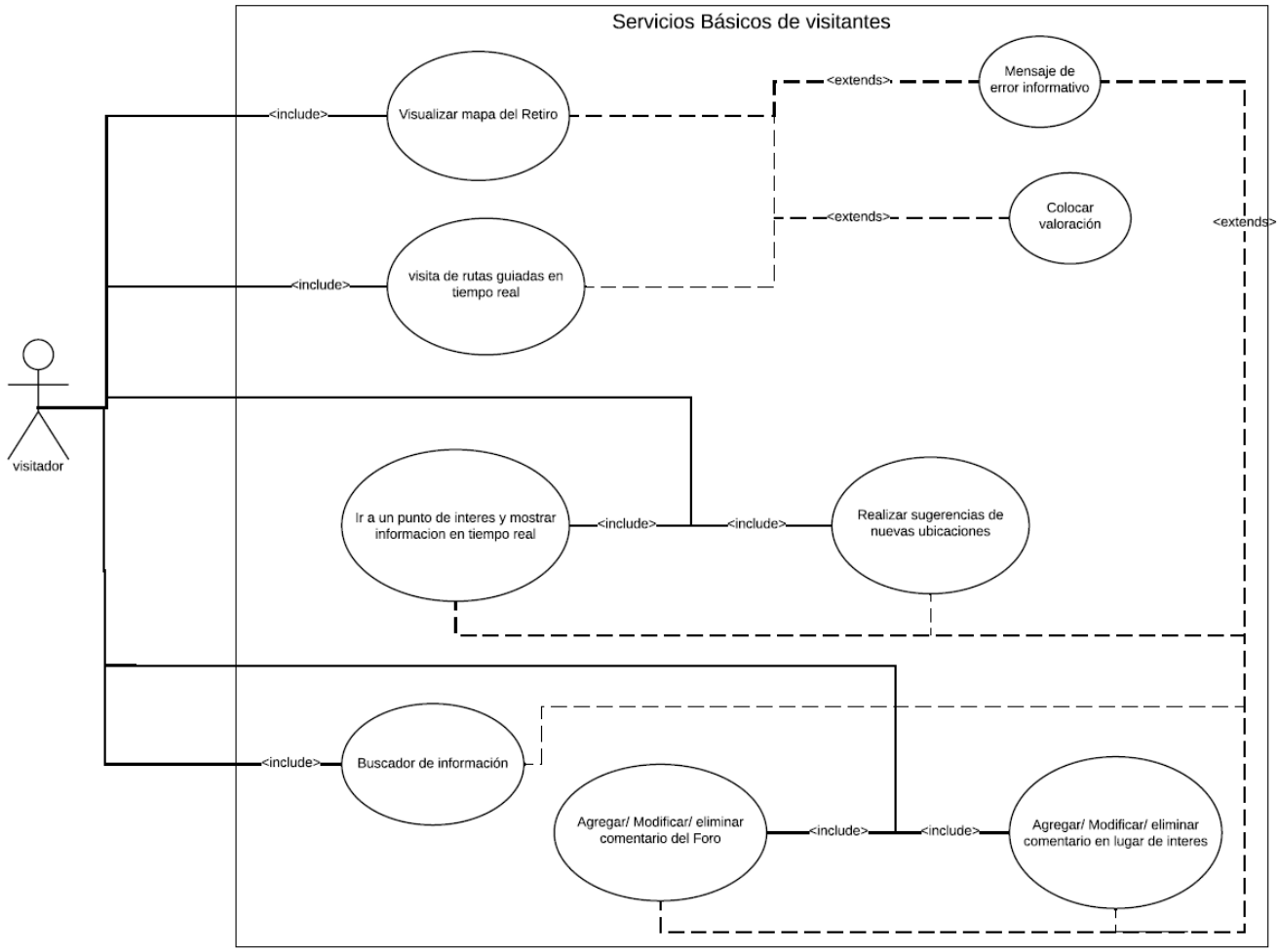


Figura 4. DIAGRAMA DE CASOS DE USO SERVICIOS VISITANTES

Requisito	**Visualizar Mapa del Retiro**	
Identificador	2.1	
Prioridad	Alta	
Precondición	El visitador debe estar registrado en el sistema o haber accedido a la aplicación.	
Descripción	Para el visitador que se ha registrado se va a desplegar un mapa en el que se muestran todas las rutas que dispone el parque pudiendo seleccionarlas. En cambio, un usuario sin registro solo podrá visualizar el mapa	
Entrada	Id usuario, localización	
Salida	NA	
Secuencia	**Paso**	**Acción**
	1	El visitador accede a la aplicación y hace click en la sección de ver mapa
	2	Para visitadores registrados se abre una ventana en la que se muestra un mapa interactivo con todas las rutas
	2.1	Selecciona ruta dentro del mapa
	2.3	Se muestra toda la información correspondiente a esa ruta
	2.4	El visitador puede dar en “empezar” o “salir”
	3	Para usuarios no registrados se despliega un mapa estático
Postcondición	El visitador acaba seleccionando una ruta de interés la cual puede seguir	
Excepciones	**Paso**	**Acción**
	2	En caso de haber seleccionado una de las rutas ofrecidas se lanza un mensaje de error
	5	No deja continuar e indica que ha de seleccionar una de las opciones
Comentarios	NA	
Actores	Visitador	

Tabla 14.CASO DE USO VISUALIZAR MAPA DEL RETIRO

Requisito	**Visualizar Mapa del Retiro**	
Identificador	2.2	
Prioridad	Alta	
Precondición	El visitador registrado debe haber iniciado sesión y seleccionado una de las rutas ofrecidas o haber accedido a la aplicación.	
Descripción	En el caso de que el visitador este registrado se va a guiar por donde debe ir para ver el contenido de la ruta. En caso de que no se encuentre registrado, solo podrá acceder a las indicaciones de forma escrita.	
Entrada	IdVisitador, localización	
Salida	NA	
Secuencia	**Paso**	**Acción**
	1	Se inicia la aplicación y mostrando la pantalla principal
	2	Después de seleccionar “ver rutas”
	3	Se abre una ventana en la que se muestra un mapa en el que se muestra la ubicación en tiempo real y el camino que debe seguir
	3.1	Una vez finalizada se pedirá una valoración
	3.2	Se llevará a la pantalla principal
	4	Se despliega la información para el usuario no registrado
Postcondición	El visitador ha interactuado en tiempo real con las distintas ubicaciones y ha disfrutado de una ruta.	
Excepciones	**Paso**	**Acción**
	2	Si no se selecciona ruta se informa con un mensaje y no deja continuar
	3	En caso de que se pierda la ubicación no permite continuar hasta que se haya recuperado
	3.1	Si no se pone una valoración se deja pasar y lleva al usuario a la pantalla principal.
Comentarios	NA	
Actores	Visitador	

Tabla 15. CASO DE USO VISUALIZAR MAPA DEL RETIRO

Requisito	**Como ir a un punto de interés y mostrar información en tiempo real.**	
Identificador	2.3	
Prioridad	Media	
Precondición	Visitador ha de haberse registrado e iniciado sesión previamente.	
Descripción	Permitirá indicar la ruta más corta a un punto de interés seleccionado por el visitador y tendrá la opción de ir consultando la información de los sitios por donde se encuentra en ese momento.	
Entrada	IdVisitador, localización, idPunto de interés.	
Salida	NA	
Secuencia	**Paso**	**Acción**
	1	Se inicia la aplicación y mostrando en la pantalla principal.
	2	Después se selecciona la opción de “ir a”.
	2.1	Si desea obtener información del punto en el que se encuentra se selecciona la opción “conoce más”.
	2.2	Muestra toda la información correspondiente al punto actual
	2.3	Al salir redirige a la pantalla en la que se encuentra la ruta
	3	Se abre una ventana en la que se muestra un mapa en el que se indica la ubicación en tiempo real y los puntos de interés más cercanos
	4	El visitador ha de seleccionar el punto que quiera y se despliega la ruta más corta
Postcondición	El visitador accede en tiempo real a la información del sistema	
Excepciones	**Paso**	**Acción**
	2.1	Si no existe ningún punto que pueda proporcionar información no se muestra la opción
	3	Si no se selecciona un punto que no se encuentra registrado se muestra un mensaje indicando que seleccione otro
Comentarios	NA	
Actores	Visitador	

Tabla 16. CASO DE USO GUIA PUNTO A PUNTO

Requisito	**Agregar/modificar/eliminar comentario en el foro**	
Identificador	2.4	
Prioridad	Alta	
Precondición	Visitador ha de haberse registrado e iniciado sesión previamente	
Descripción	Permite escribir en la comunidad, opiniones o datos relevantes sobre el parque	
Entrada	IdVisitardor	
Salida	NA	
Secuencia	**Paso**	**Acción**
	1	Se inicia la aplicación y muestra la pantalla principal
	2	Después se selecciona la opción de “foro”
	3	Se le habilitará una sección en la que poder escribir/modificar /eliminar
	4	El visitador guardará la información dando a “publicar”
Postcondición	El visitador podrá interactuar con la comunidad	
Excepciones	**Paso**	**Acción**
	4	Si no se ha podido guardar se informará por medio de un mensaje
Comentarios	NA	
Actores	Visitador	

Tabla 17. CASO DE USO AGREGAR/MODIFICAR/ELIMANAR COMENTARIO FORO

Requisito	**Agregar/modificar/eliminar comentario en lugares de interés**	
Identificador	2.5	
Prioridad	Alta	
Precondición	Visitador ha de haberse registrado e iniciado sesión previamente	
Descripción	Permite escribir en la comunidad, opiniones o datos relevantes sobre las ubicaciones	
Entrada	IdVisitardor	
Salida	NA	
Secuencia	**Paso**	**Acción**
	1	Se inicia la aplicación y muestra la pantalla principal
	2	Dentro de la sección de una ubicación tendrá la opción de comentar
	3	el visitador podrá agregar/modificar/eliminar información relevante respecto a la ubicación
	4	Hacer click en botón guardar
Postcondición	El visitador podrá compartir información relevante Respecto a las ubicaciones para que futuros usuarios	
Excepciones	**Paso**	**Acción**
	3	Si se ha producido algún error se comunicará por medio de un mensaje
Comentarios	NA	
Actores	Visitador	

Tabla 18.CASO DE USO AGREGAR/MODIFICAR/ELIMNAR LUGAR DE INTERES

Requisito	**Realizar sugerencias de nuevas ubicaciones**	
Identificador	2.6	
Prioridad	Alta	
Precondición	Visitador ha de haberse registrado e iniciado sesión previamente	
Descripción	Permite sugerir nuevas ubicaciones para su estudio y si tras su aprobación añadirlas a la ruta	
Entrada	IdVisitardor	
Salida	NA	
Secuencia	**Paso**	**Acción**
	1	Se inicia la aplicación y muestra la pantalla principal
	2	Dentro de la pantalla principal tendremos un apartado de sugerir
	3	Se mostrará un formulario que recoge la información necesaria para el estudio
	4	Dar a enviar
Postcondición	Haber realizado una participación dentro del sistema para agregar nuevos sitios	
Excepciones	**Paso**	**Acción**
	3	Si ha surgido algún fallo al guardar se informa con. Un mensaje y pidiendo que vuelva a escribir los datos
Comentarios	NA	
Actores	Visitador	

Tabla 19. CASO DE USO REALIZAR SUGERENCIAS DE NUEVAS UBICACIONES

Requisito	Buscador de información	
Identificador	2.7	
Prioridad	Alta	
Precondición	Visitador ha de haberse registrado o iniciado sesión previamente	
Descripción	Permite al visitador que se ha registrado buscar información de forma directa sin necesidad de acceder a las rutas	
Entrada	IdVisitador	
Salida	NA	
Secuencia	**Paso**	**Acción**
	1	Se inicia la aplicación y muestra la pantalla principal
	2	Dentro de la pantalla principal tendremos una barra de búsqueda
	3	Se introduce la consulta
	4	Se muestra la información
Postcondición	Acceder a información de forma directa	
Excepciones	**Paso**	**Acción**
	4	En caso de que no se obtengan resultados de la búsqueda se arroja un mensaje indicando una sugerencia
Comentarios	NA	
Actores	Visitador	

Tabla 20. CASO DE USO BUSCADOR DE INFORMACION

4.5 Gestión de rutas

En este módulo se concentran todas las funcionalidades relacionadas con las rutas a nivel de usuario y pueden ser de consulta, modificación, en la Figura 5 se muestra el diagrama de casos de uso de gestión de sendas.

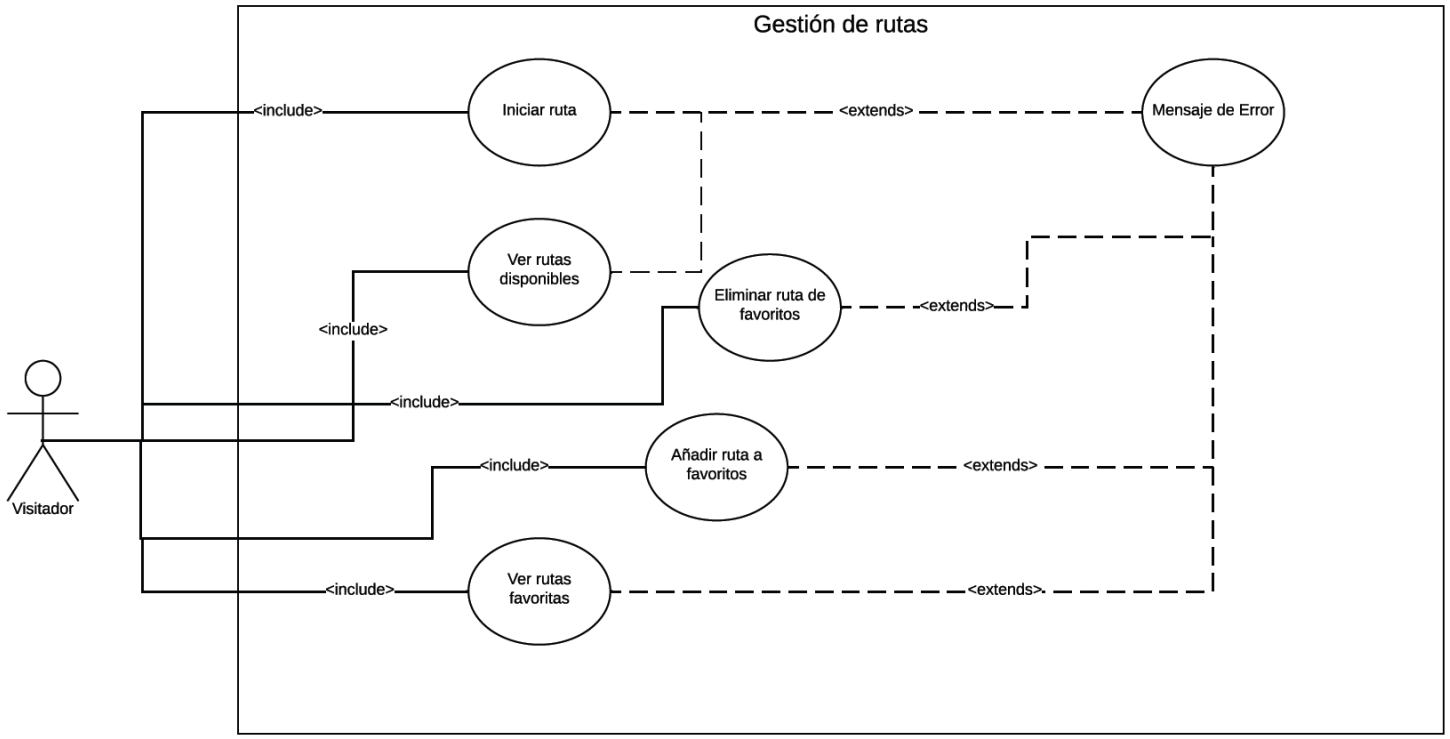


Figura 5. DIAGRAMA DE CASOS DE USO GESCION DE RUTAS

Requisito	**Ver rutas disponibles**	
Identificador	3.1	
Prioridad	Alta	
Precondición	Las rutas deben estar previamente registradas	
Descripción	Permite consultar las rutas que existen actualmente dentro del sistema	
Entrada	IdRuta	
Salida	NA	
Secuencia	**Paso**	**Acción**
	1	Se inicia la aplicación y muestra la pantalla principal
	2	Dentro de la pantalla principal tendremos una seleccionamos “rutas disponibles”
	3	Se muestran todas las rutas que se encuentran disponibles en ese momento.
Postcondición	Ver de forma clara todas las rutas que dispone el sistema	
Excepciones	**Paso**	**Acción**
	3	Si no se disponen de rutas que mostrar se comunican por medio de un mensaje:” Aún no disponemos de rutas que mostrar”.
Comentarios	NA	
Actores	Visitador	

Tabla 21. CASO DE USO VER RUTAS DISPONIBLES

Requisito	**Iniciar ruta**	
Identificador	3.2	
Prioridad	Alta	
Precondición	La ruta seleccionada debe estar registrada	
Descripción	Una vez seleccionada la ruta se puede comenzar a visitar el parque	
Entrada	IdRuta	
Salida	NA	
Secuencia	**Paso**	**Acción**
	1	Se accede a la aplicación y una vez haya seleccionado una ruta
	2	Se muestra toda la información al respecto de esta
	3	Se muestra y se hace click en el “Iniciar ruta”
	4	Se despliega un mapa en el que se muestra todo el recorrido de la ruta
Postcondición	Empezar una ruta de las opciones del visitador	
Excepciones	**Paso**	**Acción**
	2	En caso de no haber encontrado ninguna ruta se muestra un mensaje de fallo
Comentarios	NA	
Actores	Visitador	

Tabla 22.CASO DE USO INICIAR RUTA

Requisito	**Añadir ruta a Favorito**	
Identificador	3.3	
Prioridad	Media	
Precondición	La ruta debe de estar registrada en el sistema	
Descripción	Se va a permitir al visitador que añada las rutas que más le hayan gustado a una lista personal	
Entrada	idRuta	
Salida	NA	
Secuencia	**Paso**	**Acción**
	1	Se accede a la aplicación y una vez haya seleccionado una ruta
	2	Se muestra toda la información al respecto de esta
	3	Se muestra “añadir a favoritos” y se hace click y la ruta se agregará a la lista del visitador
	4	Se lleva a la ventana principal
Postcondición	Agregar ruta dentro de las opciones del visitador	
Excepciones	**Paso**	**Acción**
	3	Si no se ha podido agregar a favoritos, se mostrará un mensaje informando del fallo
Comentarios	NA	
Actores	Visitador	

Tabla 23. CASO DE USO AÑADIR RUTA A FAVORITO

Requisito	**Ver rutas favoritas**	
Identificador	3.4	
Prioridad	Media	
Precondición	Las rutas deben de estar registradas en el sistema	
Descripción	Se va a permitir al visitador listar todas las rutas que tiene guardadas	
Entrada	idRuta	
Salida	NA	
Secuencia	**Paso**	**Acción**
	1	Se accede a la aplicación y una vez haya seleccionado una ruta
	2	Se muestra “ver rutas favoritas” y se hace click
	3	Se despliega una lista en la cual se muestran todas las rutas con un pequeño detalle
Postcondición	Eliminar ruta dentro de las opciones del visitador	
Excepciones	**Paso**	**Acción**
	3	Si no tiene rutas favoritas se comunica por medio de un mensaje: “Aún no tienes rutas favoritas”
Comentarios	NA	
Actores	Visitador	

Tabla 24. CASO DE USO VER RUTAS FAVORITAS

Requisito	**Eliminar Ruta de Favoritos**	
Identificador	3.5	
Prioridad	Media	
Precondición	La ruta debe de estar registrada en el sistema	
Descripción	Se va a permitir al visitador que se eliminen rutas que ya no quiera conservar.	
Entrada	idRuta	
Salida	NA	
Secuencia	**Paso**	**Acción**
	1	Se accede a la aplicación y una vez haya seleccionado una ruta
	2	Se muestra “ver rutas favoritas” y se hace click
	3	Se despliega una lista en la cual puede eliminar la ruta deseada, seguida de una confirmación
	4	Se lleva a la ventana principal
Postcondición	Eliminar ruta dentro de las opciones del visitador	
Excepciones	**Paso**	**Acción**
	3	Si la eliminación ha fallado se genera un mensaje de error
Comentarios	NA	
Actores	Visitador	

Tabla 25. CASO DE USO ELIMINAR RUTA DE FAVORITOS

Capítulo 5 - Arquitectura de la aplicación y modelo

En el siguiente capitulo se procede a explicar la arquitectura y el modelo de datos que presenta la aplicación.

5.1 Arquitectura de la aplicación

Para el desarrollo de la aplicación se ha utilizado una arquitectura cliente-servidor que es un modelo de diseño software de los más utilizados y se diferencian en 2 partes: los demandantes de un servicio (Clientes) y los proveedores (Servidores) de este. El funcionamiento es muy simple ya que cada vez que un cliente necesite utilizar un servicio, realiza una petición al servidor que se encarga de atenderla, procesarla y devolver una respuesta con la información necesaria para satisfacer la petición.

Dentro de la arquitectura de la aplicación se implementa en la parte del servidor una interfaz de programación de aplicaciones (API) [13]. Se trata de un conjunto de definiciones y protocolos que permiten generar un nivel de abstracción a la hora de acceder a un servicio, permitiendo de esta manera:

- Tener una mayor facilidad de intercambio de información ya que el único requisito es realizar una correcta preparación de estos.
- Además de una mayor interoperabilidad, ya que puede ser utilizadas desde distintos dispositivos.

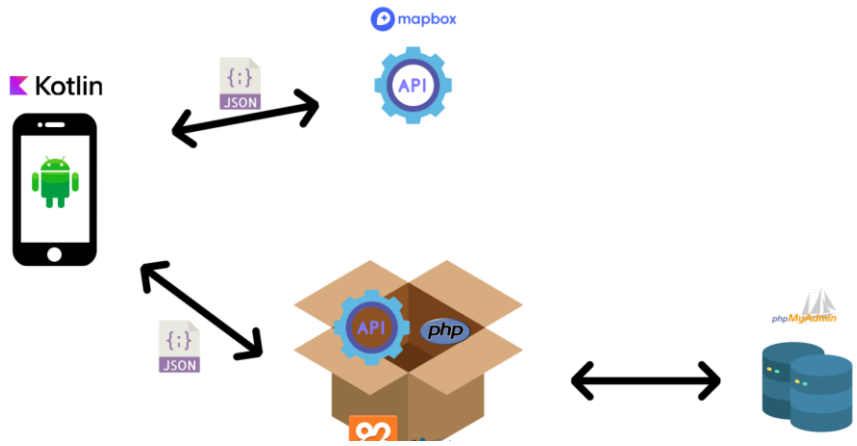


Figura 6. AQUITECTURA DEL SISTEMA

En la Figura 6 se plasma la arquitectura de cliente-servidor que se ha implementado.

Por la parte del cliente existe como entrada una aplicación cuyo lenguaje de programación es Kotlin que por medio de peticiones accede a servicios de las API´s, la de mapbox proporciona toda la información necesaria para poder trabajar con la localización del GPS y la otra es la que se ha desarrollado para el proyecto, que posee los distintos servicios que procesan las peticiones de tipo HTTP y devuelven la información pertinente

5.2 Elementos de la arquitectura

5.2.1 Aplicación móvil

Esta parte comprende lo denominado como cliente y se trata de una aplicación Android que sigue el modelo MVVM de Kotlin, reflejado en la Figura 7.

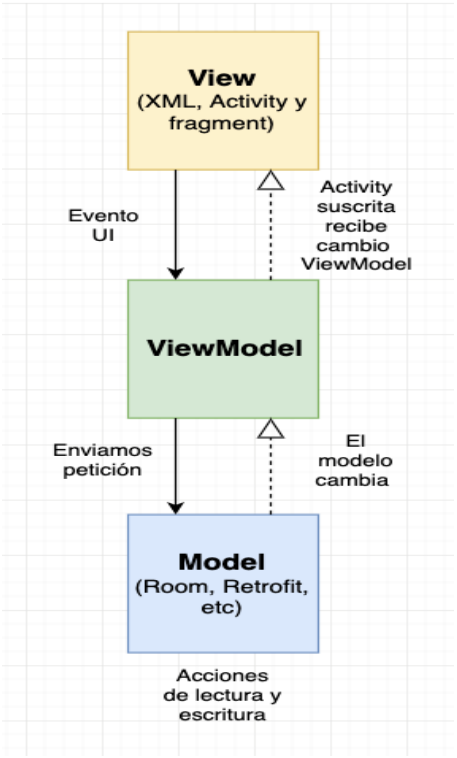


Figura 7. MODELO MVVM

A su vez realiza peticiones a la API correspondiente para llevar a cabo las distintas actividades.

5.2.2 APIs

- API creado: El cliente se comunica con los servicios que se encargan de verificar y proporcionar la información de los distintos elementos utilizados que se encuentran en la base de datos y funciona por medio de peticiones HTTP. Esta se encuentra implementada en PHP, se aloja de forma local con XAMPP y de forma remota en X10Hosting.
- Mapbox API: Es un API REST la cual utiliza el cliente para realiza todas las peticiones a servicios relacionados con la localización, mapas e indicaciones.

5.3 Diagrama de la base de datos

En la siguiente sección se procede a mostrar y explicar la estructura que se ha seguido para almacenar la información dentro del sistema, se puede ver en Figura 8.

Se ha empleado un sistema relacional gestionado por PhpMyAdmin que se ha alojado en un servidor. En el esquema se pueden observar las distintas tablas, con sus respectivas claves primarias, foráneas y las relaciones que se establecen entre ellas.

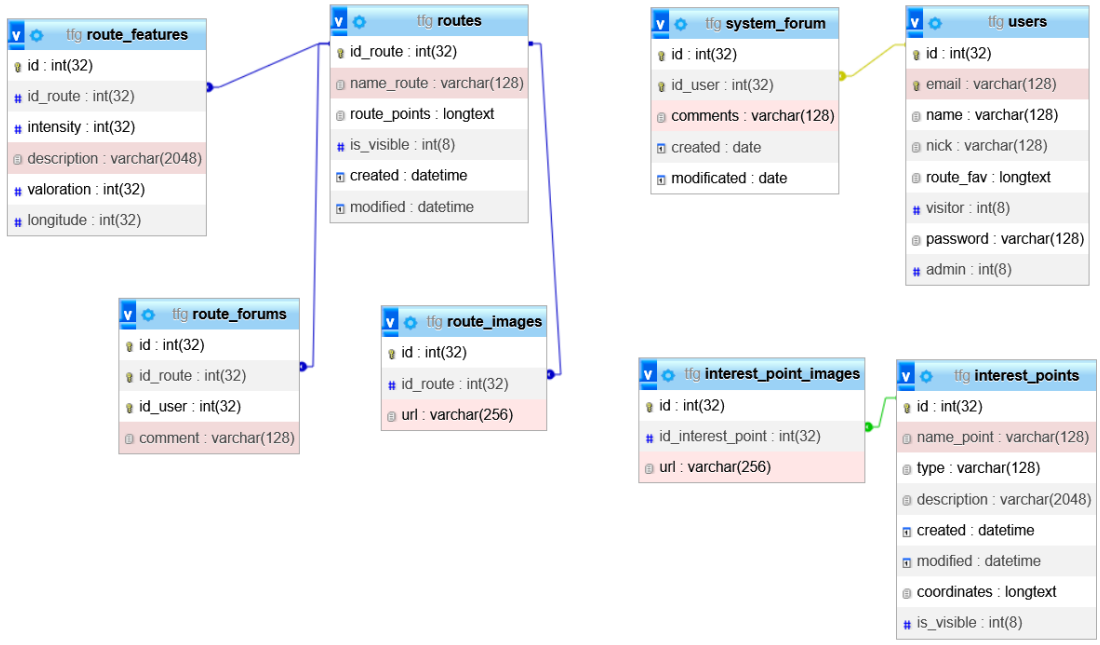


Figura 8. DIAGRAMA DE LA BASE DE DATOS

Todo este conjunto da como resultado el siguiente esquema:

- **Puntos de interés:**

  Son aquellas ubicaciones que se quieren resaltar dentro del parque. Para ello se ha creado esta tabla, en la cual se va a guardar toda la información referente a las ubicaciones junto con sus respectivas fotos que se vayan agregando.

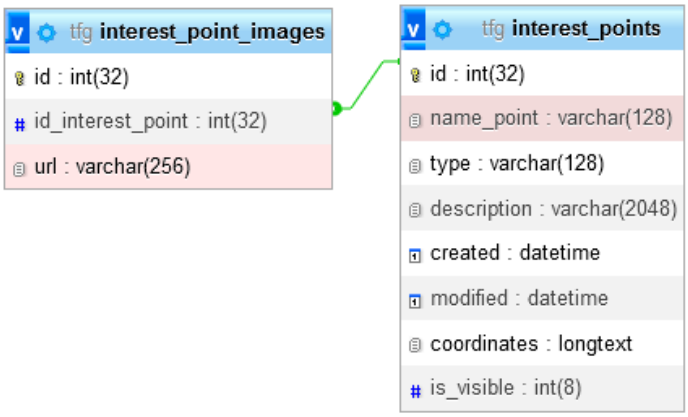


Figura 9. DIAGRAMA E.R DE PUNTOS DE INTERÉS

Dentro de la tabla de la Figura 9, se encuentra la tabla Interst_point en la que destacan las siguientes columnas:

- Id: Es el identificador del punto y este es único dentro del sistema.
- Name_point: Es el nombre por el cual se va a hacer referencia al punto en la aplicación y también es único.
- Type: Recoge el tipo de punto de interés, pudiendo ser: Monumento, Fuente, Restaurante o Árbol.
- Created y modified: Son marcas de tiempo de creación y modificación, para comprobaciones.
- Coordinates: Almacena la longitud y latitud del punto creado
- Is_visible: Es el encargado de indicar si un punto es visible al usuario.

A continuacion en la Figura 10se muestra un ejemplo de información que es almacenada.

	id	name_point	type	description	created	modified	coordinates	is_visibl
Copiar Borrar	31	Puerta de la independencia	Monumento	La función shareIn muestra un SharedFlow, un flujo...	2023-01-29	2023-04-2	-3.688076,40.419205	1

Figura 10. EJEMPLO PUNTO DE INTERES

En la tabla de interest_point_images de la Figura 9 destacan las siguientes columnas:

- ❖ Id: es el identificador de la imagen dentro del sistema
- ❖ Id_interest_point: Recoge el identificador del punto de interés al que corresponde
- ❖ url: almacena la ruta dentro del servidor donde se aloja la imagen

A continuacion, en la Figura 11 se muestra un ejemplo de información almacenada.

Figura 11.URLS DE IMÁGENES EN BASE DE DATOS

- **Sendas:**

  Son los diferentes itinerarios que pueden llegar a formar los puntos de interés que se encuentran registrados en el sistema. La Figura 12 refleja el diagrama de base de datos de las sendas.

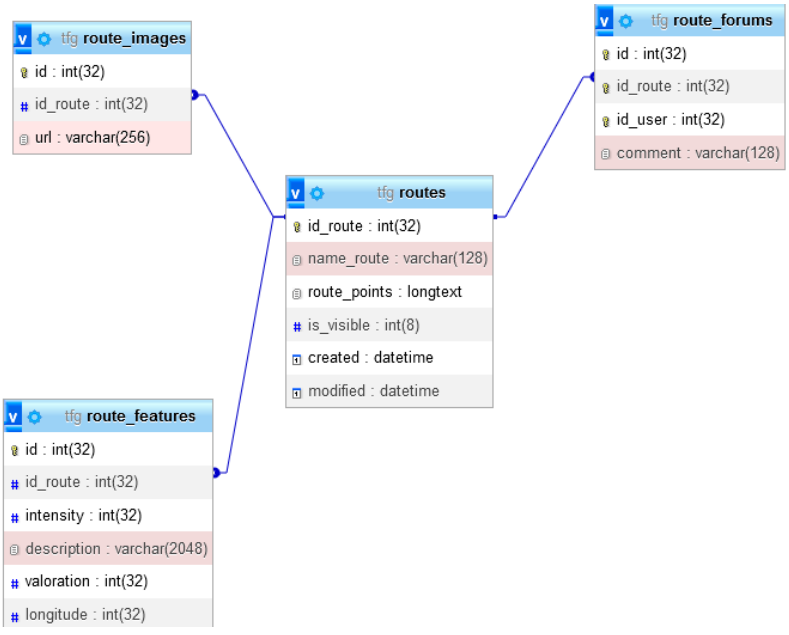


Figura 12. DIAGRAMA E.R DE SENDAS

Dentro de la Figura 12 se puede encontrar la tabla de routes que es la encargada de almacenar la información referente a las sendas dentro del sistema, compuesta por:

- Id_route: Recoge el identificador de las diferentes sendas que se van a ir creando y es único
- Name_route: Almacena el nombre de la ruta que se va a utilizar en la aplicación y también es único.
- Route_points: Es la columna encargada de guardar una colección de identificadores de puntos de interés que la forman.
- Is_visible: Es la encargada de indicar si una ruta es visible al usuario.
- Created y modified: Son marcas de tiempo que indican la creación y la modificación de la senda.

A continuación, en la Figura 13 se indica un ejemplo de la información.

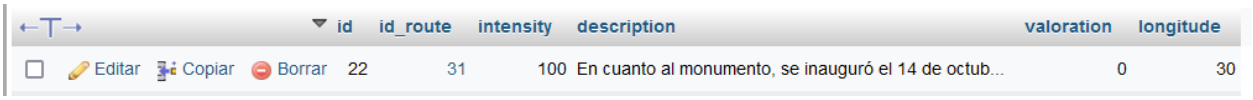
	id_route	name_route	route_points	is_visible	created	modified
Copiar Borrar	31	Los jardines más antiguos	31,34,35,36,37,38,39,4	1	2023-02-04	2023-03-18 16:0
Copiar Borrar	53	Los jardines más antiguos 2	58,59,60,61,92	1	2023-02-04	2023-03-18 13:3

Figura 13. EJEMPLO PUNTO DE SENDA

En la tabla que se muestra en la Figura 12 que recibe el nombre route_features se van a guardar todas las características de las sendas y consta de las columnas:

- Id: Identificador único que recibe la senda una vez registrada.
- Id_route: Es el índice que indica a que senda pertenece la información que contiene.
- Intesity: Guarda la dificultad con la que ha sido catalogada la senda.
- Description: Almacena una breve descripción de la senda guarda
- Valoration: Es valor numérico que irá guardando el promedio de las valoraciones realizadas por el usuario.
- Longitude: Guarda la longitud total de la que está formada la senda.

En la Figura 14, se muestra un ejemplo de datos almacenados.



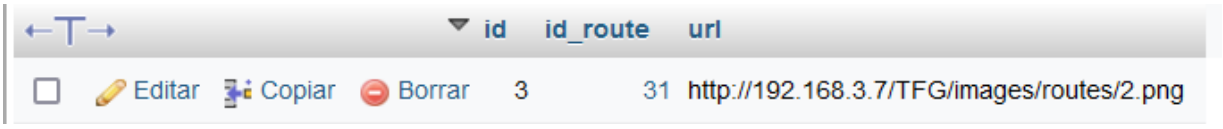
	id	id_route	intensity	description	valoration	longitude
Editar Copiar Borrar	22	31	100	En cuanto al monumento, se inauguró el 14 de octub...	0	30

Figura 14. EJEMPLO DE CARACTERISTICAS EN BASE DE DATOS

En la tabla con el nombre de route_images que se ve en la Figura 12, almacena toda la información referente a las fotos de las sendas y está formada por:

- Id: Es el identificador de la imagen dentro del sistema.
- Id_route: Recoge el identificador del punto de interés al que corresponde.
- url: Almacena la ruta dentro del servidor donde se aloja la imagen.

En la Figura 15, se muestra un ejemplo de datos almacenados.



	id	id_route	url
Editar Copiar Borrar	3	31	http://192.168.3.7/TFG/images/routes/2.png

Figura 15. URLS DE IMÁGENES DE SENDAS EN BASE DE DATOS

Dentro de la Figura 12 se muestra la tabla routes_forum que es la encargada de guardar todos los comentarios que se puedan ir realizando dentro de la aplicación.

- **Usuarios:** En la Figura 16 se muestra la tabla que refleja toda la información que se recopila para registrar a un usuario en el sistema. A continuación, se muestran los atributos:
  - Id: contiene el identificador que se le asigna en el registro y es único.
  - Email: Es correo con el que se registra el usuario siendo también único.
  - Name: Guarda el nombre del usuario con el que se produjo el registro.

- Route_fav: Contiene una lista con todos los identificadores de las sendas que ha ido agregando el usuario a favoritos.
- Visitor: Indica si el usuario ha realizado al menos una visita
- Password: Guarda la contraseña cifrada.
- Admin: Indica si el usuario tiene permisos de administrador para gestionar los datos de la aplicación.

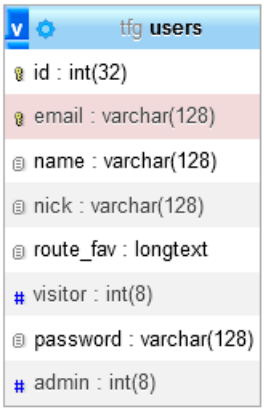


Figura 16. ESTRUCTURA DATOS DE USUARIO

En la Figura 17, se muestra un ejemplo de datos almacenados.

	id	email	name	nick	route_fav	visitor	password	admin
Editar Copiar Borrar	108	admin	lucas	admin1	ruta1,ruta2,ruta3,ruta10	1	$2y$10$k1bRx/6cBzwn2z7rlBjrPu/yNN46JR9igBRkJ0.wS3K...	1
Editar Copiar Borrar	133	luis@luis.com	luis	nombre 1	Los jardines más antiguos	1	$2y$10$rXgoKsigDBbZZeyR5S8tUez5t.ADlzR8vu	0

Figura 17. EJEMPLO DE USUARIO EN BASE DE DATOS

- **Foro del sistema:**

Es la tabla en la que se recogerá todos los comentarios realizados por los usuarios que se hayan registrado en el sistema sobre el funcionamiento de la aplicación. La Figura 18 muestra su estructura.

- ❖ Id_user: recoge el id del usuario que ha realizado la aportación.
- ❖ Comments: guarda la información que ha escrito el usuario.
- ❖ Created y modifcated: fechas que recogen las marcas de tiempo.

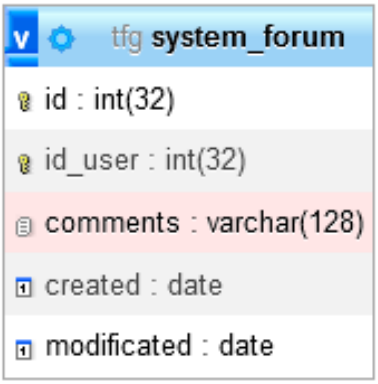


Figura 18. ESTRUCTURA DATOS DE FORO DEL SISTEMA

Capítulo 6 - Diseño de la aplicación

A continuación, se va a tratar el diseño y la implementación de las funcionalidades más importantes de dentro de la aplicación. Seguido de la explicación de la implementación de la API.

La aplicación final está destinada hacia todos los públicos, por este motivo se ha intentado plantear un estilo muy intuitivo y sencillo.

Para llevar a cabo esta tarea se han utilizado distribuciones en formato .xml que se basa en ir añadiendo componentes que, junto a kotlin, permite generar y desarrollar las interfaces de usuario.

6.1 Distribuciones y estilos

Para las distribuciones se ha tenido en cuenta que en todas las pantallas no se sobrecarguen ni con información ni disparadores de acciones, ya que de esa forma se consigue que los flujos de usuario sean lo más claros y simples, como se ve en la Figura 19.

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="22dp"
    android:layout_gravity="clip_horizontal"
    android:layout_marginTop="5dp"
    android:enabled="false"
    android:inputType="textPersonName"
    android:textSize="12sp"
    android:scrollbarAlwaysDrawHorizontalTrack="false"
    android:scrollHorizontally="false"
    android:text="Nombre"
    android:textColor="#040404 "
    android:verticalScrollbarPosition="defaultPosition" />

<EditText
    android:id="@+id/nameRoute"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_marginTop="30dp"
    android:enabled="false"
    android:text="Nombre:"
    android:textSize="13sp"

    android:textColor="#000000"
    app:layout_constraintBottom_toBottomOf="parent"
```

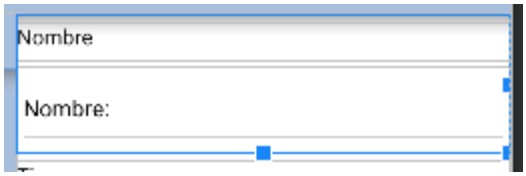


Figura 19. ESTRUCTURA CONTENEDOR DE INFORMACION

Para los estilos de los botones se han creado plantillas para que todos sean del mismo tamaño, color y forma, lo que queda reflejado en la Figura 20.

Figura 20. PLANTILLAS PARA BOTONES Y TITULOS

De igual forma, los colores de la aplicación que se ha seleccionado pertenecen a una paleta de colores que siguen la misma línea (se pueden ver en la Figura 21):

Figura 21. PALETA DE COLORES DE LA APLICACION

Por otra parte, en la Figura 22 se muestra la estructura de aquellas secciones en las que hay un listado de características. Se ha utilizado la representación de una card que contiene un textView con el identificador y un editText en el que se va a volcar la información.

```
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="1dp"
    android:orientation="vertical">
```

Figura 22. CONFIGURACION DE CARD

También se ha mantenido la misma línea para la búsqueda de iconos y su colocación, ya que todos han seguido una línea negra gruesa que forma el contorno del icono, como se muestran en la Figura 23.

Figura 23.FORMATO DE ICONOS UTILIZADOS

6.2 Funcionalidad de la aplicación

A lo largo de este apartado se van a describir las principales funcionalidades que presenta la aplicación. Para ello se ha de hacer una subdivisión en apartados, los cuales contienen Modelo, View y ViewModel (MVVM).

6.2.1 Entidades principales

Son las tres entidades encargadas de la recepción de los datos procedentes de las acciones realizadas por el usuario y su preparación para ser enviados como una petición al servidor.

A su vez se encargan de la interpretación de la respuesta y de su guardado para su procesamiento dentro de la aplicación.

- dbUsers: Contiene todos los métodos que guardan cualquier relación con el usuario, como se muestra en la Figura 24.

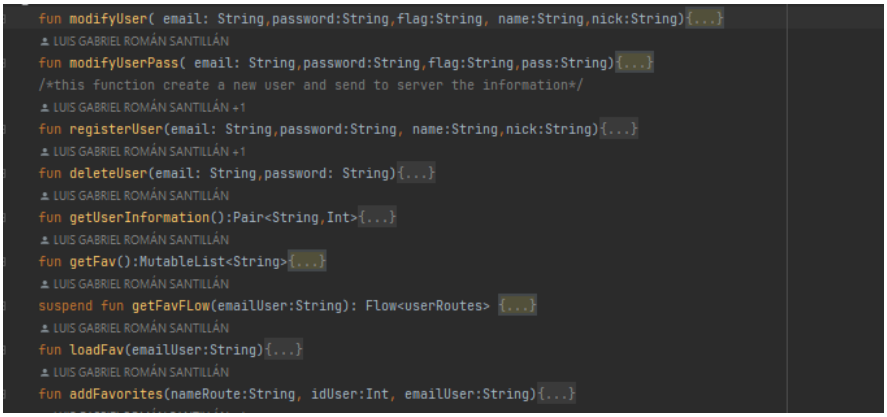


Figura 24. METODOS DE DBUSERS

- dbInterestPoints: Contiene los métodos que tienen relación con los puntos de interés, como se muestra en la Figura 25.

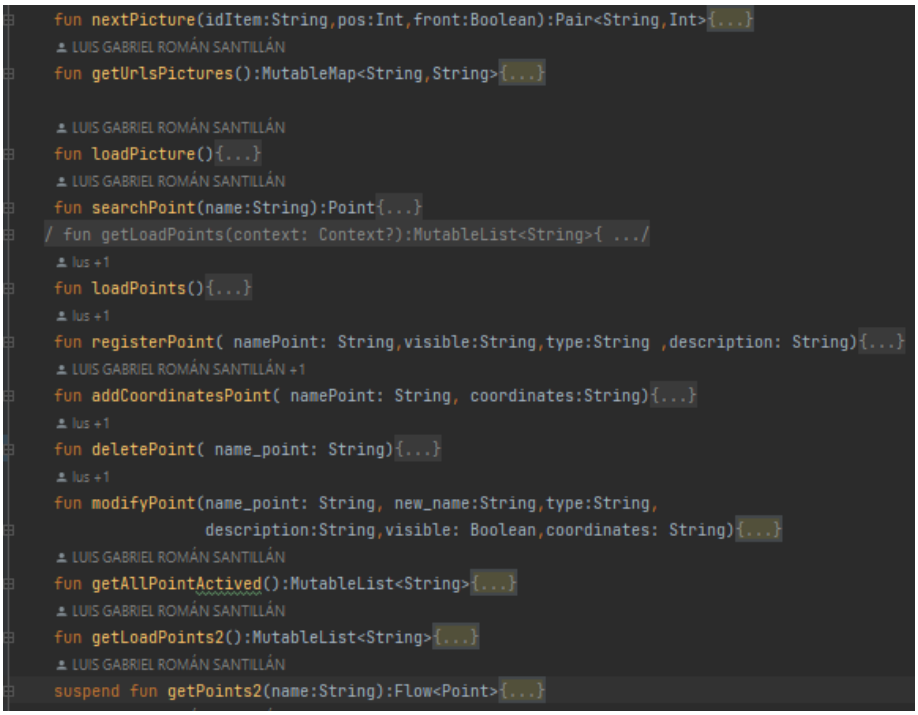


Figura 25. METODOS DE DBINTERESTPOINT

- dbRoutes: Recoge las funciones que tienen relación con las sendas, como aparece en la Figura 26.

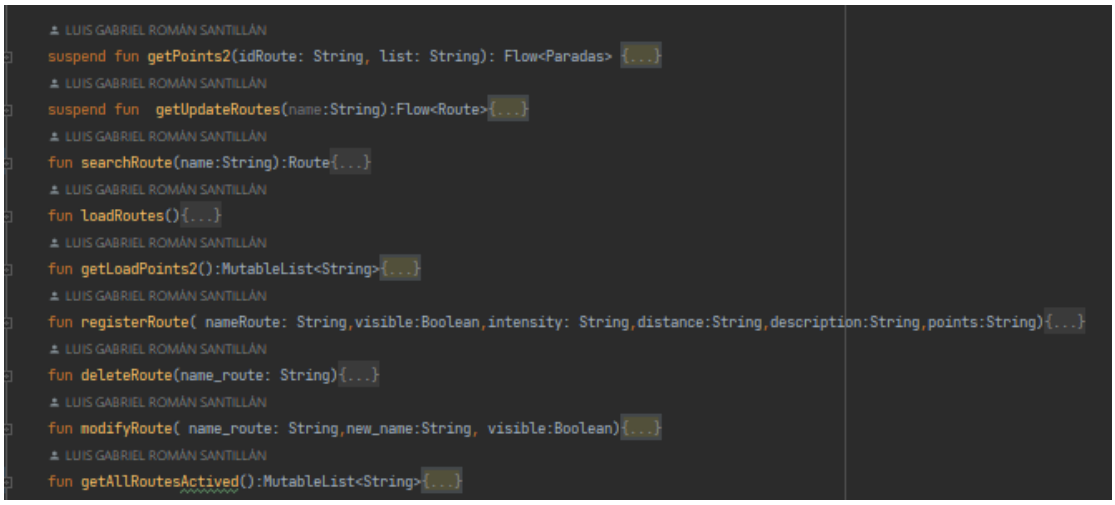


Figura 26. METODOS DE DBROUTES

6.2.2 Vistas de usuario

Para realizar acciones de usuario como eliminación de cuenta, modificaciones de datos y contraseña, seguimos el siguiente esquema:

ConfigurationViewModel: Es el encargado de interpretar los datos y enviarlos a la entidad correspondiente, como son datos de usuario usamos dbUser para hacer las peticiones al servidor.

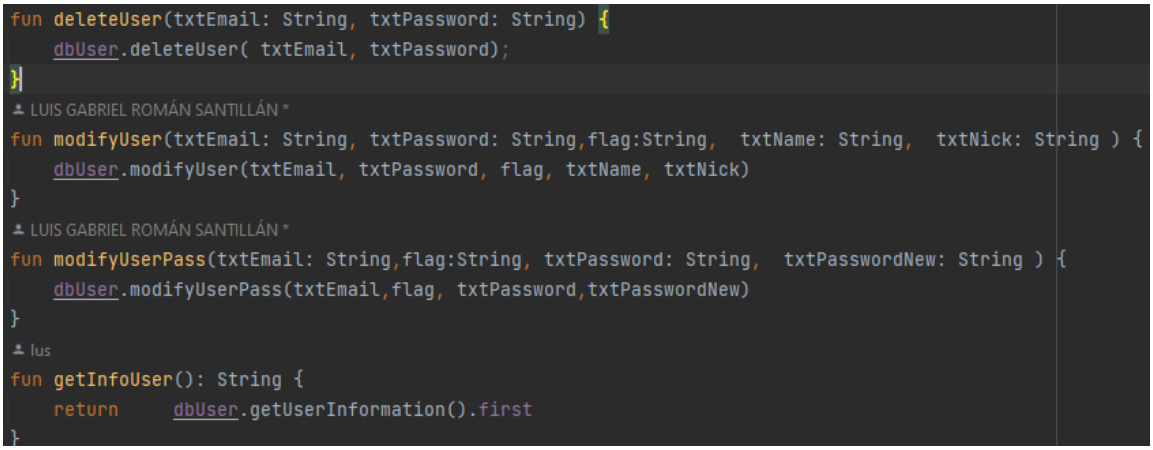


Figura 27. EJEMPLO CONFIGURATIONVIEWMODEL

Seguido de los fragments, que contienen la recepción de las acciones de usuario. Están DeleteFragment que se encarga de la eliminación de un perfil, ModifyFragment de los

distintos cambios a nivel de datos como el nombre, contraseña, que se pueden ver en la Figura 28 y finalmente ModifyPassword que capta los cambios de contraseña, ver Figura 29. A continuación, se muestran algunos ejemplos de código, en los que se refleja la creación de la vista con los distintos elementos dentro de cada una y cómo reacciona e interactúa con el Viewmodel.

```
        _binding = FragmentModifyBinding.inflate(inflater, container,  attachToParent: false)
        val root: View = binding.root
        viewModel=ConfigurationViewModel()
        val textView: TextView = binding.TextHeader
        configurationviewModel.textHeader.observe(viewLifecycleOwner) {  it: String!
            textView.text = it
        }

        val txtName=root.findViewById<TextInputLayout>(R.id.idNameModify)
        val txtNick=root.findViewById<TextInputLayout>(R.id.idNickModify)
        val txtPassword=root.findViewById<TextInputLayout>(R.id.password)
        val buttonCancel=binding.idCancelBtnModify
        val buttonSave=binding.idSaveBtnModify
        buttonSave.setOnClickListener{...}
        buttonCancel.setOnClickListener{...}


        return root
    }
```

Figura 28. EJEMPLO DE MODIFYFRAGMENT

```
    _binding = FragmentPasswordBinding.inflate(inflater, container,  attachToParent: false)
    val root: View = binding.root
    viewModel=ConfigurationViewModel()
    binding.textInfo.text = "A continuación te dispones a realizar cambios sobre tu contraseña"
    val buttonCancel=binding.idCancelBtnModify
    val buttonSave=binding.idSaveBtnModify
    val txtPassword=root.findViewById<TextInputLayout>(R.id.password)
    val txtPasswordnew=root.findViewById<TextInputLayout>(R.id.idPasswordModify)
    val txtPasswordConf=root.findViewById<TextInputLayout>(R.id.idPasswordConfirmModify)
    buttonSave.setOnClickListener{...}
    buttonCancel.setOnClickListener{...}
    return root
}
```

Figura 29. EJEMPLO DE MODIFYPASSWORD

6.2.3 Vistas de punto de interés

Para la realización de acciones en las que se ven implicados los puntos de interés se ha implementado la clase, ***InteresPointViewModel***:

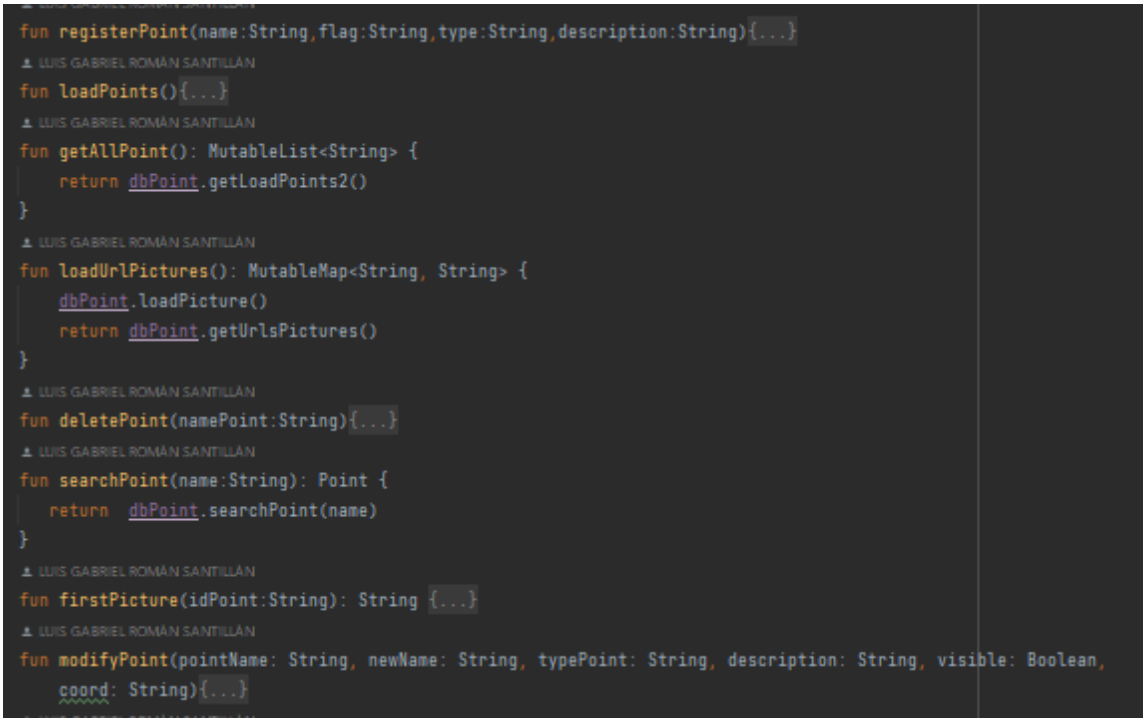


Figura 30. EJEMPLO INTERESTPOINTMODEL

Seguido de los fragments que generan las distintas vistas que representan las acciones y comprobaciones referentes a las acciones que se pueden hacer con un punto de interés como, por ejemplo: CreateInterestPoint para la creación de un punto, se puede (ver en la Figura 31); ModifyInterestPoint para modificarlo (ver Figura 32); ShowInterestPoint para generar la vista en la cual se muestra toda la información del punto de interés (ver Figura 33).

A continuación, se muestra algunos fragmentos de código de las partes mencionadas:

```
buttonCreatePoint.setOnClickListener { it: View!
    validateParametres(namePoint,typePoint,descripText)
    if(namePoint.editText?.text.toString().isNotEmpty() && typePoint.editText?.text.toString().isNotEmpty()
            && descripText.text.toString().isNotEmpty()){
        var name =  namePoint.editText?.text.toString()
        var type=typePoint.editText?.text.toString()
        var description=descripText.text.toString()
        var visible:Boolean= isVisible.isChecked
        namePoint.error=null
        descripText.error=null
        typePoint.error=null
        pointViewModel.registerPoint(name,visible.toString(),type,description)
        val result = name
         name=""
         type=""
        descripText.setText("")
        setFragmentResult( requestKey: "namePoint", bundleOf( ...pairs: "namePoint" to result))
        this.parentFragment?.findNavController()?.navigate(R.id.action_createInterestPoint_to_createPointFeatures)
    }
```

Figura 31. EJEMPLO DE CREATEINTERESTPOINT

```
_binding = FragmentModifyInterestPointBinding.inflate(inflater, container, attachToParent: false)
val root: View = binding.root
val Data=binding.data
val ImageContainer= binding.fotos
viewModel = InterestPointViewModel()
val NamesData= Data.findViewById<ScrollView>(R.id.scrollData)
var namePoint = NamesData.findViewById<TextInputLayout>(R.id.idNewNamePoint)
var typePoint = NamesData.findViewById<TextInputLayout>(R.id.idTipo)
var longitude= NamesData.findViewById<TextInputLayout>(R.id.coordLongitude)
var latitud= NamesData.findViewById<TextInputLayout>(R.id.coordLatitude)
var description= binding.receipt
val isVisible = Data.findViewById<Switch>(R.id.switchVisibleButtonP)
val buttonModifyPoint = binding.idModifyPointBtn
val buttonLocationPoint = binding.buttonLocation
```

Figura 32. EJEMPLO DE MODIFYINTERESTPOINT

```
    val item = adapter.getItem(position)
    posItem=position
    tv1.text = item
    var point =pointViewModel.searchPoint(item.toString())
    namePoint.editText?.setText(point.getName())
    typePoint.editText?.setText(point.type)
    latitud.editText?.setText(point.getLatitude().toString())
    longitude.editText?.setText(point.getLongitude().toString())
    description.setText(point.desc)
```

Figura 33 EJEMPLO DE SHOWINTERESTPOINT

6.2.4 Vistas Senda

Para la realización de acciones en las que se ven implicados las sendas se ha implementado la clase, ***RoutesViewModel:***

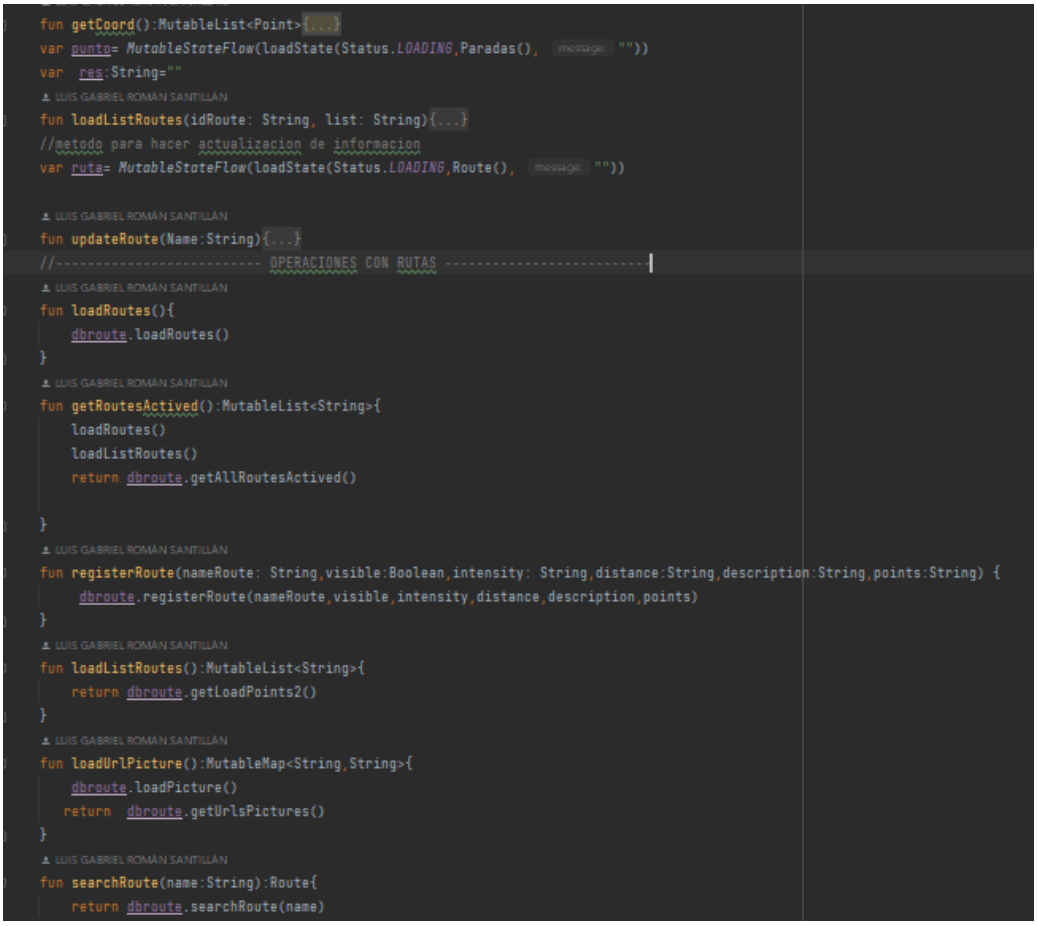
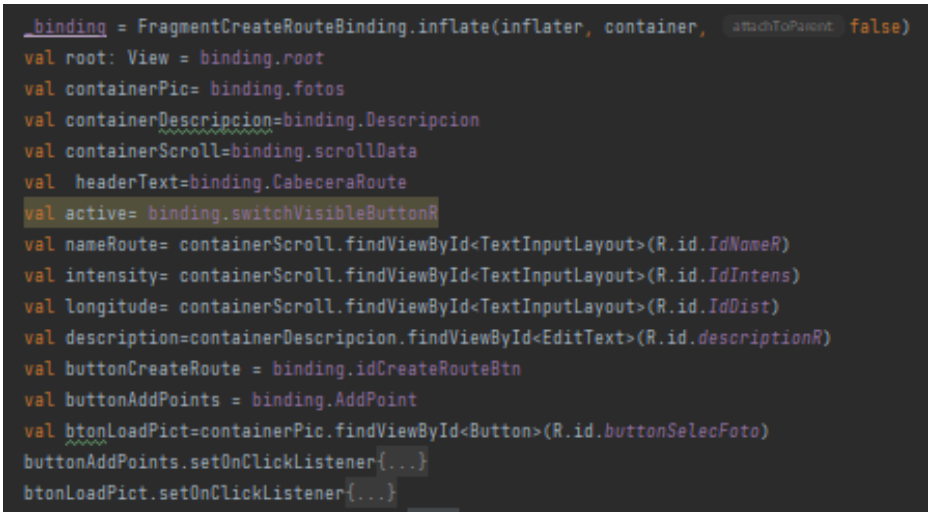


Figura 34. EJEMPLO DE ROUTESVIEWMODEL

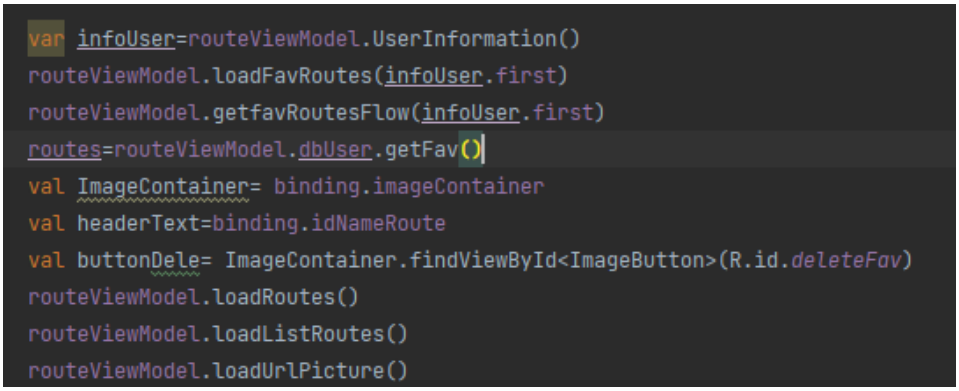
Además, se han implementado los fragments que se encargan de la creación de rutas (CreateRouteFragment) contenida en la Figura 35, mostrar toda la información referente a las Sendas (ShowRoutes) se muestra en la Figura 36 y la modificación de esta (ModifyRouteFragment) se puede ver en la Figura 37.

Todos los fragmentos poseen métodos que permiten realizar comprobaciones y recogen las interacciones que puede realizar el usuario con la aplicación.



```
_binding = FragmentCreateRouteBinding.inflate(inflater, container, attachToParent: false)
val root: View = binding.root
val containerPic= binding.fotos
val containerDescripcion=binding.Descripcion
val containerScroll=binding.scrollData
val  headerText=binding.CabeceraRoute
val active= binding.switchVisibleButtonR
val nameRoute= containerScroll.findViewById<TextInputLayout>(R.id.IdNameR)
val intensity= containerScroll.findViewById<TextInputLayout>(R.id.IdIntens)
val longitude= containerScroll.findViewById<TextInputLayout>(R.id.IdDist)
val description=containerDescripcion.findViewById<EditText>(R.id.descriptionR)
val buttonCreateRoute = binding.idCreateRouteBtn
val buttonAddPoints = binding.AddPoint
val btonLoadPict=containerPic.findViewById<Button>(R.id.buttonSelecFoto)
buttonAddPoints.setOnClickListener{...}
btonLoadPict.setOnClickListener{...}
```

Figura 35. EJEMPLO CREATEROUTEFRAGMENT



```
var infoUser=routeViewModel.UserInformation()
routeViewModel.loadFavRoutes(infoUser.first)
routeViewModel.getfavRoutesFlow(infoUser.first)
routes=routeViewModel.dbUser.getFav()
val ImageContainer= binding.imageContainer
val headerText=binding.idNameRoute
val buttonDele= ImageContainer.findViewById<ImageButton>(R.id.deleteFav)
routeViewModel.loadRoutes()
routeViewModel.loadListRoutes()
routeViewModel.loadUrlPicture()
```

Figura 36. EJEMPLO DE SHOWROUTES

```
    .setTitle("Modificacion de la senda") AlertDialog.Builder!
    .setMessage("Se va a proceder a realizar la modificación de la senda")
    .setNegativeButton( text: "Cancelar") { view, _ ->
        Toast.makeText(activity, text: "Cancel button pressed", Toast.LENGTH_SHORT).show()
        view.dismiss()
    }
    .setPositiveButton( text: "Continuar") { view, _ ->
        routeViewModel.modifyRoute(headerText.text.toString(),
            newNameRoute.editText?.text.toString(),visible)
        routeViewModel.updateRoute(newNameRoute.editText?.text.toString())
        view.dismiss()
       this.parentFragment?.findNavController()?.navigate(R.id.action_modifyRouteFragment_to_nav_routes2)
    }
    .setCancelable(false)
    .create()
dialog.show()
```

Figura 37. EJEMPLO DE MODIFYROUTEFRAGMENT

6.2.5 Vistas de las utilidades

En este apartado se va a tratar una serie de ejemplos que se han utilizado para la representación de una ubicación dentro del mapa o subir imágenes al servidor.

- ***LoadPictures:*** Se va a encargar de realizar la creación de todos los elementos que forman parte de la vista. Además, recibe un indicador y un nombre que permite gestionar la subida, como se ve en la Figura 38.

```
setFragmentResultListener( requestKey: "flagC") { requestKey, bundle ->
    param1= bundle.getString( key: "flagC")
}
setFragmentResultListener( requestKey: "nameC") { requestKey, bundle ->
    param2=bundle.getString( key: "nameC")
}
```

Figura 38. EJEMPLO LOADPICTURES

- ***PointLocation:*** Es el fragment encargado de realizar la representación de la localización del usuario y las coordenadas de un punto de interés dentro de un mapa, Estas se reciben a través de SetFragementResultListener bajo la key coord. Se puede ver en la Figura 39 la carga el mapa en el fragment.

```
mapView = binding.mapView
mapView?.getMapboxMap()?.loadStyleUri(
    Style.MAPBOX_STREETS,
    object : Style.OnStyleLoaded {
        override fun onStyleLoaded(style: Style) {
            addAnnotationToMap()
        }
    }
)
```

Figura 39. EJEMPLO POINTLOCATION

6.3 API

A lo largo de este capítulo se va a explicar la estructura y el funcionamiento de la API que se ha utilizado para gestionar las peticiones realizadas por la aplicación y acceder a la información de la base de datos.

El esquema general para devolver los resultados a la aplicación y es el siguiente:

Data[“code”], Data [“menssage”], Data[“user”] siendo este último opcional.

Todos los ficheros utilizan el método crearBD() que se muestra en la Figura 40, que junto con las credenciales se encargan de establecer la conexión con la base de datos para que se puedan realizar las consultas pertinentes.

```php
function crearBD(){
  $hostname='localhost';
  $database='jxrnaqsp_TFG';
  $username='jxrnaqsp_usuario';
  $password='Migatotomas12';

  mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
  $mysqli = mysqli_connect('localhost', 'root', '', 'tfg');
  /* Set the desired charset after establishing a connection */
  mysqli_set_charset($mysqli, 'utf8mb4');


  return $mysqli;
```

Figura 40. CREACION BASE DE DATOS

Dentro de la carpeta User se agrupan todos los ficheros que se encargan de gestionar y administrar toda la información referente al usuario.

6.3.1 Registrar_usuario.php

Se encarga de recibir datos como el email, password, name, nick, visitor y realizar las consultas necesarias para poder registrar un nuevo usuario en el sistema, como se muestra en la Figura 41.

```php
$passEnctrypted=password_hash($usu_password, PASSWORD_DEFAULT);
$sql="INSERT INTO users (email,name,nick,visitor,password,admin)
      VALUES ('$usu_email','$usu_name','$usu_nickname',0,'$passEnctrypted',0)";
if (mysqli_query($mysqli, $sql)) {
  $data["code"]=2000;
  $data["message"]="El Usuario se ha registrado correctamente";
} else {
  $data['code']=2002;
  $data['message']="Problema al registar el usuario";
}
}
```

Figura 41. REGISTRO DE USUARIO

6.3.2 Validar_usuario.php

Es el encargado de realizar la comprobación de las credenciales de usuario tras recibirlas. Al usar la función password_verify() permite hacer una comprobación con la contraseña cifrada.

A su vez se devuelve toda la información del usuario como respuesta, ver figura 42.

```php
$id=$fila['id'];
$name=$fila['name'];
$email=$fila['email'];
$nick=$fila['nick'];
$admin=$fila['admin'];
$password=$fila['password'];
$routes=$fila['route_fav'];
$cliente= array('id'=> $id, 'name'=> $name, 'email'=> $email, 'nick'=> $nick,
          'password'=>$password,'admin'=> $admin, 'routesFav'=>$routes);
```

Figura 42. COMPROBACION DE USUARIO

6.3.3 Cargar_rutas_fav.php

Se utiliza para realizar la carga de todas rutas que tiene el usuario como favoritas, para ello solo es necesario recibir el email del usuario y realizar la consulta como se muestra en la Figura 43.

```php
$usu_email=$_POST['email'];
$sentencia;
$data=array();
$sentencia = $mysqli->query("SELECT * FROM users WHERE email='$usu_email'");
    if($sentencia && $sentencia->num_rows < 1){···
    }
    else{
        $fila = $sentencia->fetch_assoc();
            $ruta= $fila['route_fav'];
            $data['code']=2000;
            $data['message']=$ruta;
    }
```

Figura 43.CARGAR RUTAS FAVORITAS

6.3.4 Actualizacion_favoritas.php

Se encarga de actualizar las rutas favoritas cada vez que se agregue o elimine una senda dentro del sistema. Para ello recibe el email, id del usuario seguido de una lista que incluye los identificadores con las nuevas rutas, como se muestra en la Figura 44.

```php
    $sql="UPDATE users SET route_fav='$usu_routes' where email='$usu_email'
            and id='$usu_id'";
```

Figura 44. ACTUALIZACION DE SENDAS FAVORITAS

Dentro de la carpeta interest_point se encuentran aquellas funciones que interactúan con los datos de los puntos de interés que se encuentran en la base de datos.

6.3.5 Crear_punto.php

Recibe el nombre, tipo, descripción y un atributo que indica si es visible o no, además en el momento de la creación se agrega un campo que agrega una marca de tiempo que indica la creación y de la última modificación, como se muestra en la Figura 45.

```php
$name_point=$_POST['name_point'];
$type= $_POST['type'];
$description= $_POST['description'];
$visible= $_POST['is_visible'];
```

Figura 45.CREACION DE PUNTO DE INTERES

6.3.6 ConsultarImagen.php

Es un método creado para cargar todas las urls de las imágenes pertenecientes a los puntos de interés, Para ello es necesario recibir el identificador del punto que se está visualizando y utilizarlo como en la Figura 46.

```
while ( ($fila = mysqli_fetch_array($sentencia))!=NULL){

    $i=0;
    $idPoint= $fila['id'];
    $datos1= $idPoint. " ";
    $sentencia1 = $mysqli->query("SELECT * FROM interest_point_images where id_interest_point='$idPoint'");

    $urls='';
    while ( ($fila2 = mysqli_fetch_array($sentencia1))!=NULL){
        if($i==$sentencia1->num_rows-1){
            $urls.=$fila2['url'];
        }
        else{
            $urls.=$fila2['url'] .';';
        }
        $i++;
    }
```

Figura 46. CONSULTAR IMAGEN

6.3.7 Todos_los_puntos.php

Realiza la preparación de todos los datos en una variable como la que aparece en la Figura 47, para posteriormente devolverlos al iniciar la aplicación.

```
$i=0;
while ( ($fila = mysqli_fetch_array($sentencia))!=NULL){
    $id='';
    $id.=$fila['id'].'-'.$fila['name_point'] ;
    $info=array("idName"=>$id,"desc"=> $fila['description'],"visible"=> $fila['is_visible'],"type"=> $fila['type'],
                "coordinates"=>$fila['coordinates']);
    if($i<$sentencia->num_rows-1){
        $datos.=json_encode($info)."_";
    }
    else{
        $datos.=json_encode($info);
    }
    $i++;
}
```

Figura 47. CARGA DE TODOS LOS PUNTOS

6.3.8 UploadImage.php

Se encarga de recibir bitmap y los datos pertinentes para realizar el registro de una foto, y por medio del parámetro $_post["carpeta"] se indica donde se debe registrar ya que puede ser de un punto de interés o una senda, generando la ruta correspondiente como en la Figura 48.

```php
$idFoto= $fila['id'];
$idFoto+=1;
$path = "../images/$carpeta/$idFoto.png";
$url="images/$carpeta/$idFoto.png";
$actualpath = "http://192.168.3.7/TFG/$url";
if($carpeta=="routes"){
    $nuevaRuta = $con->query("select id_route from routes where name_route='$name'");
    if($nuevaRuta && $nuevaRuta->num_rows < 1){
        $flag=false;
    }
    else{
        $filaId = $nuevaRuta->fetch_assoc();
        $id=$filaId['id_route'];
        $sql = "INSERT INTO route_images (id_route,url) VALUES ('$id','$actualpath')";
    }
}
```

Figura 48.CARGA DE IMAGEN

En la carpeta routes se encuentran todos los ficheros que realizan las consultas, modificaciones, eliminación de las sendas en la base de datos.

6.3.9 Crear_ruta.php

Recoge los datos que se introducen en la aplicación como el nombre de la ruta, la intensidad, descripción, visibilidad y se crea un nuevo registro en la base de datos, como en la Figura 49.

```php
$fechaActual = date('y/m/d H:i:s');
$fechaModificada=$fechaActual;
$sql="INSERT INTO routes (name_route,route_points,is_visible,created,modified)
    VALUES ('$name_route',$pointsJson,$is_visible,'$fechaActual','$fechaModificada' )";
if (mysqli_query($mysqli, $sql)) {
    $sentenciaId = $mysqli->query("select * from routes order by id_route desc limit 1");
    $fila = $sentenciaId->fetch_assoc();
    $idRoute= $fila['id_route'];

    $sqlFeatures="INSERT INTO route_features (intensity,id_route,description,valoration,longitude)
      VALUES ('$intensity','$idRoute','$description','$valoration','$distance' )";
    if (mysqli_query($mysqli, $sqlFeatures)) {
        $data["code"]=2000;
```

Figura 49. CREAR RUTA

6.3.10 CoordenadasPuntos.php*:*

Es el encargado de registrar los puntos de interés que forman la senda tras su registro con la información mínima, previamente mencionada, como se puede ver en la Figura 50.

```
if(!empty($coordinates)){
    $newCoordinates=   $coordinates;
}
else{
    $newCoordinates= $fila['coordinates'];
}
$sql="UPDATE interest_points SET  coordinates ='$newCoordinates',
            modified ='$modified' where name_point='$name_point'";

if (mysqli_query($mysqli, $sql)) {
```

Figura 50. GUARDAR COORDENADAS

6.3.11 ConsultarImagen.php:

Reúne todas las url's pertenecientes a las sendas y las devuelve a la aplicación para que puedan ser utilizadas, para ello realiza la consulta que se muestra en la Figura 51.

```php
while ( ($fila = mysqli_fetch_array($sentencia))!=NULL){
    $i=0;
    $idPoint= $fila['id_route'];
    $datos1= $idPoint. " ";
    $sentencia1 = $mysqli->query("SELECT * FROM route_images where id_route='$idPoint'");
    $urls='';
    while ( ($fila2 = mysqli_fetch_array($sentencia1))!=NULL){
        if($i==$sentencia1->num_rows-1){
            $urls.=$fila2['url'];
        }
        else{
            $urls.=$fila2['url'] .';';
        }
        $i++;
    }
    if($i!==0){
        $conj++;
        $imagesURLs[$idPoint]=$urls;
        $datos1.=$urls ;
    }
    if( $j>0 && $sentencia1->num_rows>0){
        $datos.=",";
        $datos.=$datos1;
    }
    else if( $sentencia1->num_rows>0) {
        $datos.=$datos1;
        $datos.="";
    }
    $j++;
}
```

Figura 51. CONSULTAR IMAGEN

6.4 Estructura de almacenamiento de las imágenes:

Todas las fotos que se suben a través de la aplicación se guardan en la ruta ./TFG/Images/ y, dependiendo del contenido que tiene el parámetro $_post[“carpeta”] mencionado en el punto 6.3.2 Punto de interés -> UploadImage.php, se va a guardar en la subcarpeta points o routes en formato .png.

A su vez existe una foto que recibe el nombre de imagenPrincipal.png, la cual se usa para situaciones en las que se produce algún fallo de carga con alguna url.

6.5 Uso de Mapbox

Es un API que permite a la aplicación el uso y la creación de mapas, para ello primero hay que realizar el registro en la web, seguido de la obtención de un token que se ha de utilizar para realizar las peticiones una vez agregado el SDK para Android.

Para la instalación dentro del proyecto se ha de agregar los sdk de mapbox, con las versiones correspondientes, como se ve en la Figura 52. Además, hay que configurar las credenciales para que se autorice las llamadas a los distintos servicios, como se ve en la Figura 53.

```
implementation 'com.mapbox.maps:android:10.11.1'
implementation "com.mapbox.navigation:android:2.10.1"
```

Figura 52. INSTALACION SDK MAPBOX

```
dependencyResolutionManagement {
    repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)
    repositories {
        google()
        mavenCentral()
        maven {
            url 'https://api.mapbox.com/downloads/v2/releases/maven'
            authentication {
                basic(BasicAuthentication)
            }
            credentials {
                // Do not change the username below.
                // This should always be `mapbox` (not your username).
                username = "mapbox"
                // Use the secret token you stored in gradle.properties as the password
                password =SDK_REGISTRY_TOKEN
```

Figura 53. CONFIGURACION CREDENCIALES MAPBOX

A continuación, se muestra en la Figura 54, como se ha de crear la autentificación para el uso del sdk dentro del proyecto. Y en la Figura 55 se muestra la definición de las API´s a los cuales se pueden realizar las llamadas de los servicios.

```
    private fun initNavigation() {
        MapboxNavigationApp.setup(
            NavigationOptions.Builder(this.requireActivity())
                .accessToken("pk.eyJ1IjoibHVpc3RmZyIsImEiOiJjb6V0cnhtOXIxamI1M3FwNHdw...")
```

Figura 54. INICIALIZACION DE SERVICIOS MAPBOX

```
val routeLineOptions = MapboxRouteLineOptions.Builder(this.requireActivity())
    .withRouteLineBelowLayerId( layerId: "road-label-navigation")
    .build()
routeLineApi = MapboxRouteLineApi(routeLineOptions)
routeLineView = MapboxRouteLineView(routeLineOptions)
val routeArrowOptions = RouteArrowOptions.Builder(this.requireActivity()).build()
routeArrowView = MapboxRouteArrowView(routeArrowOptions)
```

Figura 55. USO DE APIS MAPBOX

Capítulo 7 - Conclusiones y trabajo futuro

En este apartado se va a realizar, por un lado, un análisis y una reflexión sobre la aplicación que se ha implementado y, por otro lado, se va a exponer una serie de funcionalidades que se plantean para futuras versiones.

7.1 Conclusiones

A lo largo de este Trabajo de Fin de Grado se ha ido creando una aplicación que reúne distintas formas de visitar El parque del Retiro de Madrid, manteniendo siempre el objetivo de ser interactiva y sobre todo fácil de utilizar. Para ello se ofrece la información de los distintos elementos junto con la gran diversidad de sendas existentes incorporando siempre la ubicación del usuario.

Durante el desarrollo del código se ha mantenido una estructura sencilla facilitando de esta manera la comprensión y a su vez intentado tener presente que sea escalable de cara a futuro.

Por otra parte, el uso de Mapbox facilita la incorporación de los mapas junto con el uso de la ubicación en tiempo real, siendo estos una parte fundamental del trabajo y que proporcionan esa faceta interactiva.

Para el desarrollo de este trabajo se han estructurad dos ideas principales: por un lado, utilizar aquellas herramientas que están al alcance de todos y que son gratuitas y, por otro lado, hacer uso de los conocimientos adquiridos durante las diferentes asignaturas de la carrera, como son Sistemas Web, Base de datos y las diferentes asignaturas de programación.

Gracias a todo esto, se ha podido crear esta aplicación Android, que recibe el nombre de Visita el Retiro.

7.2 Trabajo futuro

A lo largo del trabajo han ido surgiendo nuevas ideas que pueden encajar con la aplicación y se pueden dejar de cara futuras versiones, como son las siguientes:

- **Implementación de foros del sistema:** Consistiría en la creación de un mecanismo que permita reflejar todas las opiniones de aquellos usuarios que usen la aplicación generando feedbacks constantes que serían tratados. De esta manera se permitiría una evolución constante de la aplicación.
- **Gestión del foro de los puntos de interés y de las sendas:** Actualmente el usuario puede visitar los puntos uno por uno sin tener nada más que la información registrada por el administrador, pero con este mecanismo se ofrece al usuario la posibilidad de leer las opiniones que han dejado registradas otras personas previamente. Dando así al usuario la opción de realizar visitas basándose en un criterio diferente.
- **Realizar sugerencias de nuevas ubicaciones:** Esta funcionalidad permitiría al usuario sugerir aquellas ubicaciones que consideren que puedan ser relevantes para mejorar la visita por el parque. Estas sugerencias han de ser previamente estudiadas y aprobadas por el administrador antes de que formen parte del sistema.
- **Buscador de información:** Es una herramienta que agilizaría de forma considerable la navegación por los distintos elementos que componen el sistema y añadiría una faceta interactiva con los datos de la aplicación. Ya que actualmente debes ir por los apartados uno a uno de puntos de interés o sendas.
- **Mejoras y correcciones:** Buscar distintas estructuras de datos que mejoren la interpretación y comunicación entre cliente-servidor. Además de otras tecnologías como un servicio de hosting que permita una mayor velocidad de conexión para el uso en exteriores.

Chapter 7 - Conclusions and future work

In this part of the Project is goes to present, on the one hand, an analysis and a reflection on the Application that has develop and, on the other hand, goes to expose some functionalities that are proposed for future versions.

7.1 Conclusions

Throughout this Project, have been creating an application that brings together different ways of visiting El Retiro of Madrid, always maintaining the main objective of being interactive and above all, easy to use. For this, the information of the different parts is offered with the diversity of the paths, always including the location of the user.

During the development of the code, have maintained a simple structure, making it easier to understand and at the same time trying to keep in mind that It is scalable for the future.

In addition to this, the use of Mapbox facilitates the incorporation of the maps with the use of the location in real time, being an essential part of the Project and providing that interactive part.

For the development of this Project, has used two main ideas: on the one hand, have use those free tools that are available to everyone and, on the other hand,make use of the knowledge and everything learned during all the subjects of this degree, such as: Web Systems, Database and the different programming subjects.

Thanks to all this, I have been able to create this Android Application called "Visita el Retiro".

7.2 Future work

Throughout the Project have come up with new ideas that can be used with the Application and can be left for future versions, such as:

- **Implementation of system forums:** It involves the creation of a mechanism that allows viewing all the opinions of the users who use the Application, generating constant feedback. This would allow an effective evolution of the Application.
- **Management of the interest points forum and paths:** currently the user can visit the points one by one without having anything more than the information, but with this mechanism the user could read the opinions that other people have previously registered. This gives the user the option to make visits based on a different opinion.
- **Make suggestions for new locations:** This functionality would allow the user to suggest those locations that they consider relevant to improve the visit of El Retiro. These suggestions must be previously studied and approved by the administrator before they become part of the system.
- **Information search engine:** It is a tool that would considerably speed up the navigation through the different elements that form the system. I would add an interactive facet with the Application data, since currently you must go through the sections of interests points or paths one by one.
- **Improvements and corrections:** It consists in looking for different data structures that improve the interpretation and communication between client and server. In addition to other technologies such as a hosting service that allows a higher speed connection for outdoor use.

BIBLIOGRAFÍA

[1] citymapperlimited, «Citymapper,» [En línea]. Available: https://citymapper.com/?lang=es.

[2] «AroundMe,» [En línea]. Available: https://en.wikipedia.org/wiki/AroundMe.

[3] Mobil, «ZOO AQUARIUM Madrid,» [En línea]. Available: https://www.zoomadrid.com/prepara-tu-visita/informacion-relevante/nuestra-app.

[4] A. d. Madrid, «Centro de información y Educación Ambiental,» [En línea]. Available: https://diario.madrid.es/cieaelretiro/descubre-el-retiro-con-tu-movil-apps-para-usar-en-el-parque/.

[5] A. d. Madrid, «Bienvenidos a Madrid,» [En línea]. Available: https://www.esmadrid.com/otras-apps-madrid.

[6] A. Leiva, «Devexperto.com,» 2021. [En línea]. Available: https://devexperto.com/que-es-kotlin-y-para-que-sirve/.

[7] «PHP,» 2023. [En línea]. Available: https://www.php.net/manual/es/intro-whatis.php.

[8] I. d. Souza, «Rockcontent,» 2020. [En línea]. Available: https://rockcontent.com/es/blog/php/.

[9] G. Developers, «Developers,» [En línea]. Available: https://developer.android.com/studio/intro?hl=es-419.

[10] «PhpMyAdmin,» 2023. [En línea]. Available: https://www.phpmyadmin.net.

[11] F. Flores, «OpenWebinars,» 2022. [En línea]. Available: https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/.

[12] D. Camacho, «Platzi,» 2021. [En línea]. Available: https://platzi.com/blog/que-es-github-como-funciona/.

[13] P. V. Climent, «MappingGIS,» 2023. [En línea]. Available: https://mappinggis.com/2019/11/que-productos-y-servicios-ofrece-mapbox/.

[14] S. L. Mora, «Digital55,» 2023. [En línea]. Available: https://digital55.com/que-es-mapbox/.

[15] R. keepCoding, «keepCoding,» 2023. [En línea]. Available: https://keepcoding.io/blog/que-es-postman/.

[16] «Digital Guide IONOS,» 2023. [En línea]. Available: https://www.ionos.es/digitalguide/servidores/herramientas/instala-tu-servidor-local-xampp-en-unos-pocos-pasos/#:~:text=XAMPP%20es%20una%20distribución%20de,de%20programación%20Perl%20y%20PHP..

[17] A. Gómez, «Hosting,» 2023. [En línea]. Available: https://hostinggratis.xyz/opiniones/x10hosting.

[18] «Interfaz de programación de aplicaciones,» [En línea]. Available: https://es.wikipedia.org/wiki/Interfaz de programacion de aplicaciones.

APÉNDICES

Apéndice A - Guía de uso

En este apéndice se explica cómo se puede utilizar la aplicación desde el punto de vista de un administrador, un usuario registrado y uno no autentificado. A continuación, se van a explicar aquellos usos que son comunes a todos los usuarios.

Pantalla inicial

Lo primero que se ve al acceder a la aplicación es una pantalla inicial (ver Figura 56) en la que se encuentra 2 botones que sirven para informar por medio de un pop-up que es un punto y una senda.

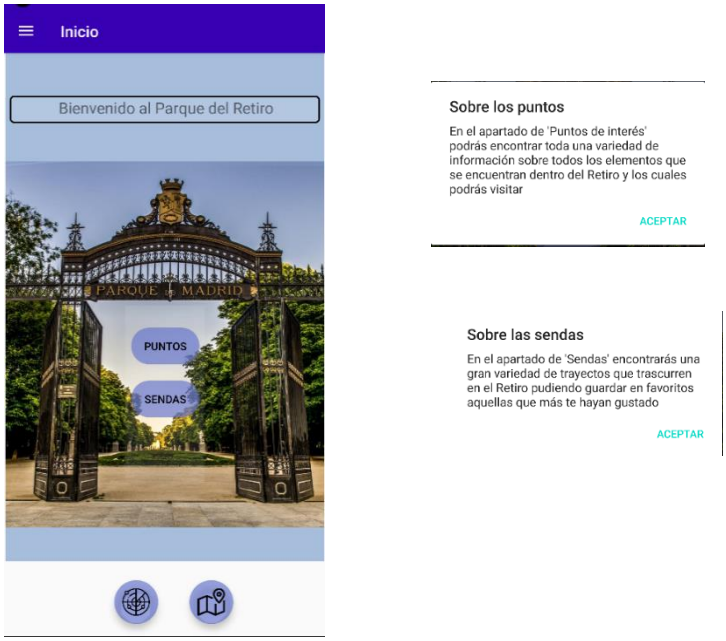


Figura 56. PANTALLA INCIAL

Además, en la parte inferior de la pantalla se encuentran 2 botones que sirven de acceso directo a la función de radar y a la de navegación libre.

Función Radar

Pulsando el botón de Radar de la pantalla principal despliega una pantalla en la que se puede ver un mapa que contiene la ubicación en tiempo real del usuario además de todos aquellos puntos que es encuentran registrados en el sistema y se encuentren a una distancia inferior a 100 metros, como se muestra en la Figura 57.

Figura 57. FUNCION RADAR

Navegación libre

Pulsando el botón de navegación libre se despliega una pantalla que contiene la ubicación actual del usuario, pero a diferencia del radar aquí se encuentran todos los puntos que estén registrados en el sistema. Además, existe una funcionalidad que permite por medio de una pulsación larga generar las instrucciones en tiempo real para llegar al punto seleccionado, todo se encuentra reflejado en la Figura 58 .

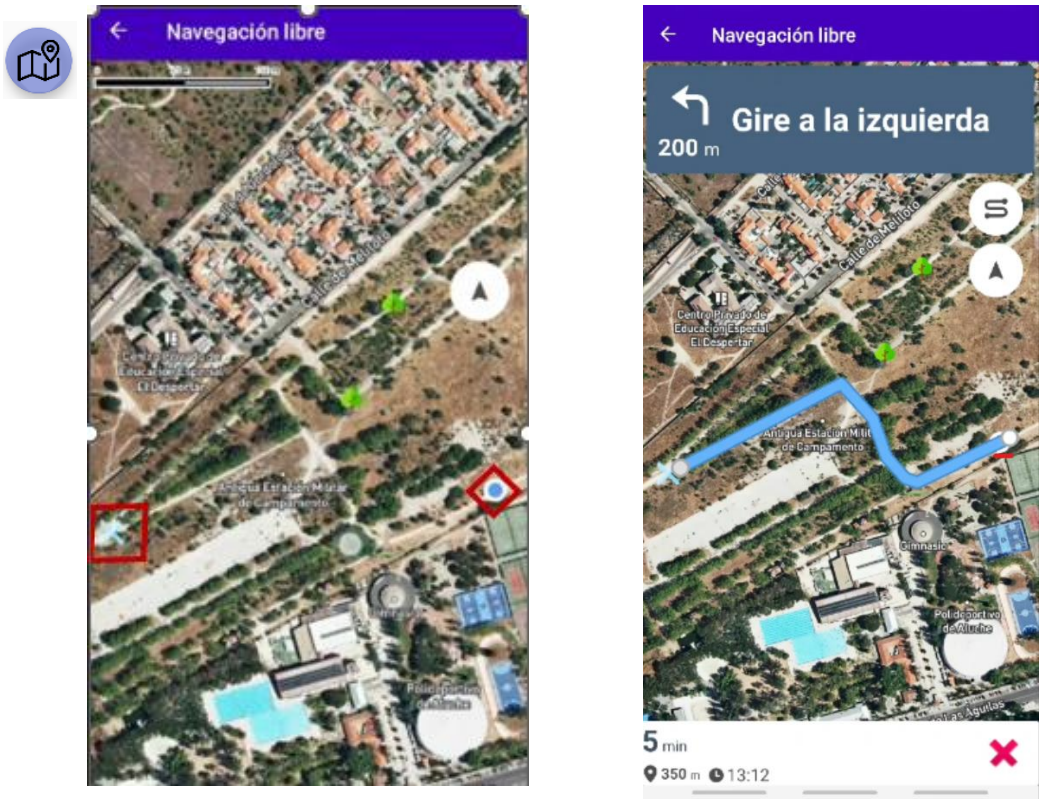


Figura 58. FUNCION NAVEGACION LIBRE

Menú lateral

Dentro de la pantalla de inicio se encuentra un botón que despliega el menú vertical que contiene varios accesos de la aplicación, como se puede observar en la Figura 59.

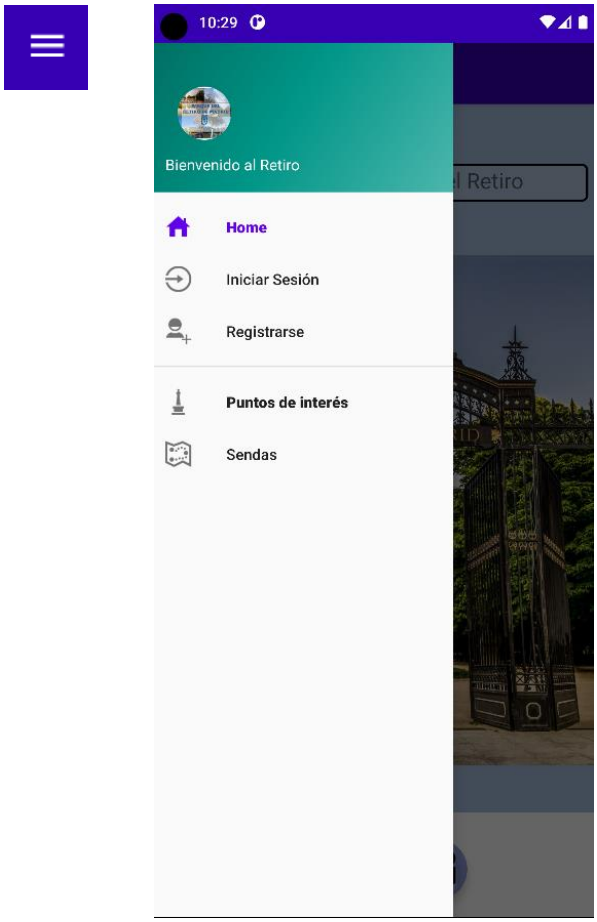


Figura 59. MENU LATERAL

Dentro de usuarios no registrados están las siguientes

Registrarse

A través de la siguiente pantalla que se puede ver en la Figura 60 el usuario que no está registrado puede completar el formulario que recoge el email, Nick, nombre y contraseña. Pudiendo hacer 2 acciones, enviarlo o cancelarlo. Si se decide enviar se realizan una serie de comprobaciones de tal forma que si algo falla en la verificación se pondrá en rojo indicando el fallo. En caso de cancelar borra el contenido que se ha escrito en las cards del formulario, permitiendo volver a empezar el registro.

Figura 60. REGISTO USUARIO

Visualizar los puntos de interés

Para poder acceder a los puntos de interés lo primero que se muestra es una pantalla principal en la que se encuentra la selección de Ver puntos. Al seleccionarlo se abre una vista en la cual aparece un conjunto de funcionalidades que se recogen en la Figura 61 como, por ejemplo:

- ítem izquierdo: es una lista de elementos la cual aparecen todos los puntos y según se vaya seleccionando toda la información se va actualizando.
- Flechas: Permiten navegar entre las distintas imágenes que dispone el punto de interés en la base de datos
- Cuadro de datos: en la parte inferior derecha se encuentra un apartado en el cual se puede ver toda la información como el nombre, tipo, descripción y la localización del punto seleccionado.

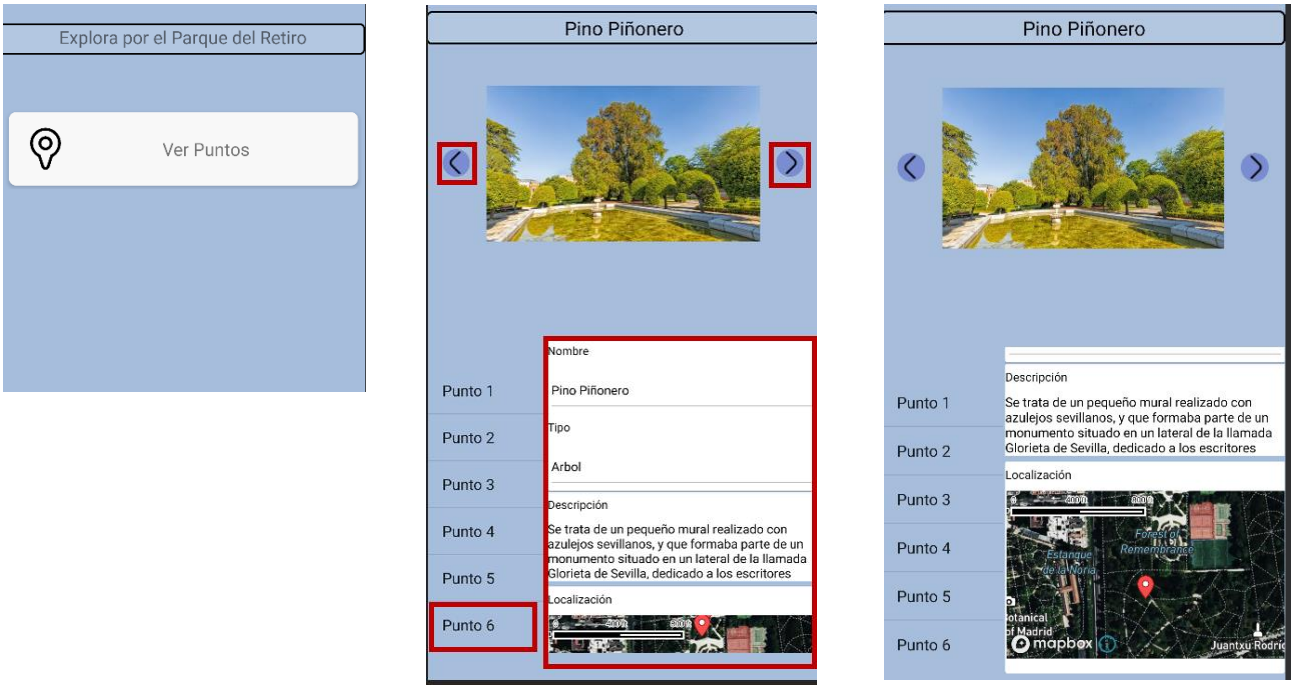


Figura 61. VER PUNTO USUARIO NO RESGISTRADO

Visualizar las sendas

Para acceder a la visualización de las sendas, lo primero que se puede ver es una pantalla principal en con un botón. Al presionarlo redirige a otra en la cual existen varias funcionalidades, que se recogen en la Figura 62:

- Item izquierdo: Es una lista de elementos la cual aparecen todas las sendas y según se vaya seleccionando toda la información se va actualizando.
- Flechas: Permiten navegar entre las distintas imágenes que dispone la senda en la base de datos.
- Cuadro de datos: En la parte inferior derecha se encuentra un apartado en el cual se puede ver toda la información como el nombre, intensidad, distancia, descripción y al final un botón "ver ruta".

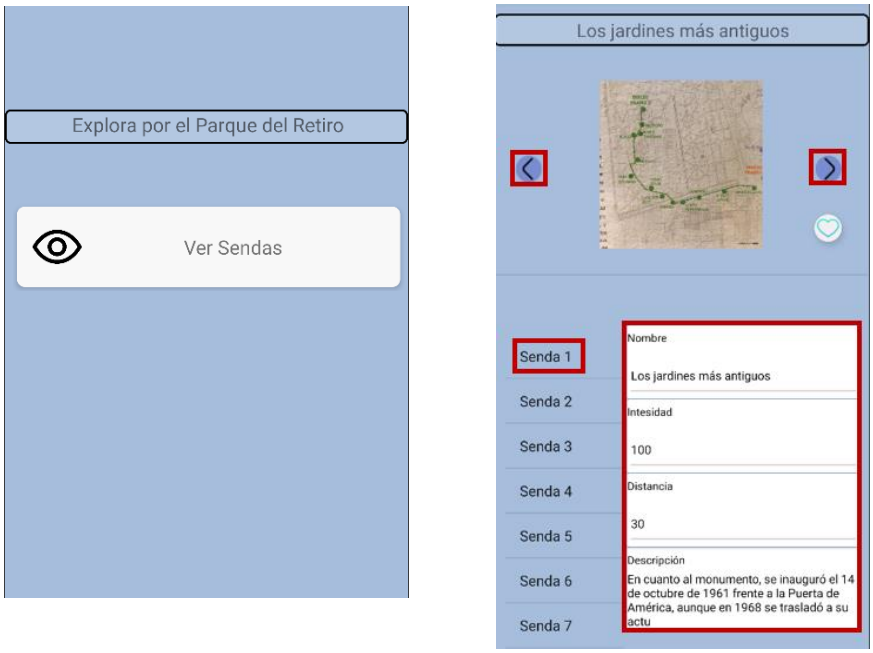


Figura 62. VER CARACTERISTICAS DE SENDA USUARIO NO REGISTRADO

Al seleccionar el botón de Ver senda muestra un pop-up el cual indica que se va a visualizar un mapa (ver Figura 63).

- Continuar: Permite avanzar y visualizar el mapa de la ruta con todos los puntos que la contienen.
- Cancelar: Corta el avance desapareciendo el pop-up y manteniendo la pantalla de información.

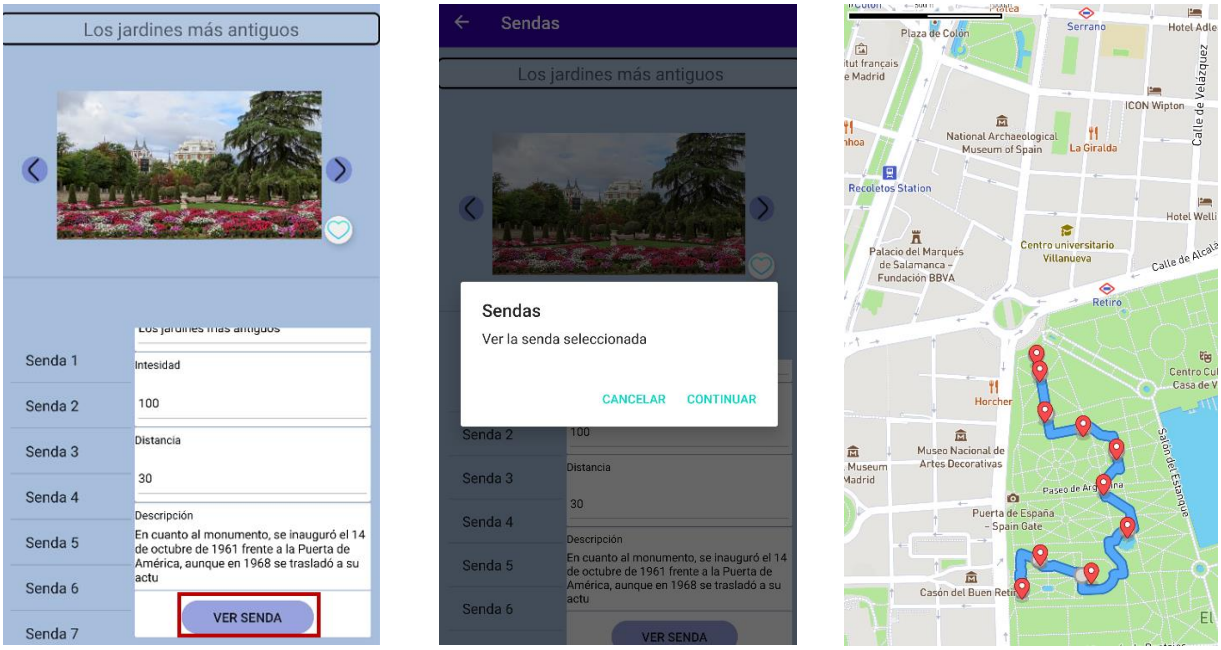


Figura 63. VISUALIZACION DE SENDA

Al presionar el botón del corazón muestra un mensaje en la parte inferior de la pantalla indicando que esa función solo está permitida para usuarios registrados, como se ve en la Figura 64.

Figura 64. AGREGAR A FAVORITOS SIN ESTAR REGISTRADO

Las funcionalidades comunes dentro de los usuarios registrados tenemos:

Inicio de Sesión

En la siguiente pantalla (ver Figura 65) se muestra un formulario con dos campos:

- Email: Se introduce el correo electrónico con el que se ha registrado el usuario dentro del sistema.
- Contraseña: En este campo se introduce la contraseña del usuario y se puede consultar la contraseña por medio del icono de la derecha

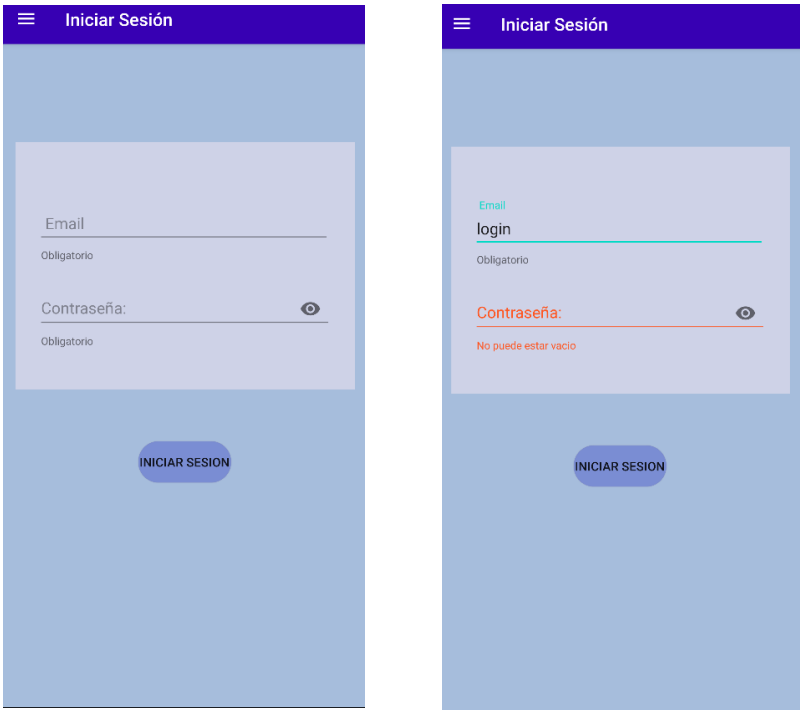


Figura 65. INICIO DE SESION

Configuración

Como punto de partida de este apartado se encuentra una pantalla principal en la que tenemos los accesos a las funciones que modifican los datos con los que el usuario se registró, como se muestra en la Figura 66.

Figura 66. VISTA CONFIGURACION

Modificar Datos: A través de esta vista que se puede ver en la Figura 67, se puede realizar modificaciones del nombre y nick. Una vez se presione el botón de guardar se realiza la comprobación de los campos.

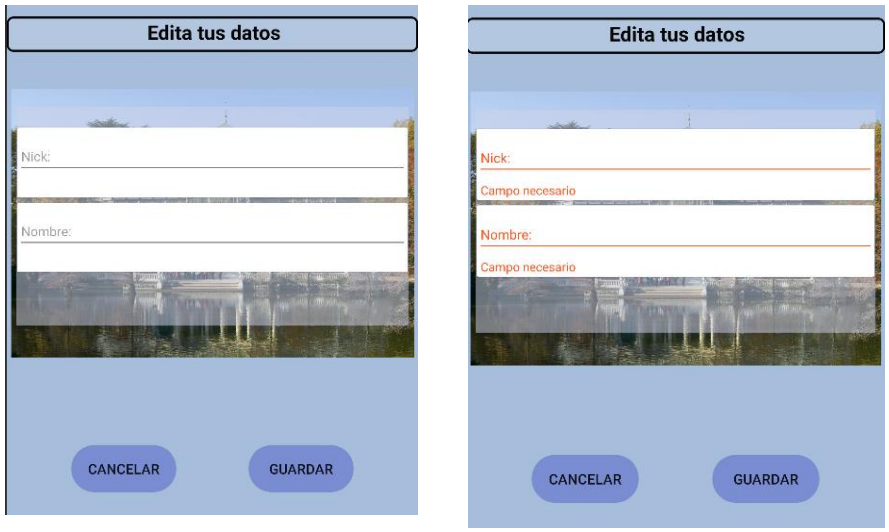


Figura 67. MODIFICACION DE DATOS USUARIO

Darse de baja: Introduciendo el email de registro y la contraseña el usuario puede darse de baja, se puede ver en la Figura 68.

Figura 68. DARSE DE BAJA

Cambiar contraseña: En la Figura 69 se muestra la vista en la que se pueden realizar modificaciones de la contraseña. Se introducen los nuevos datos seguido de la contraseña.

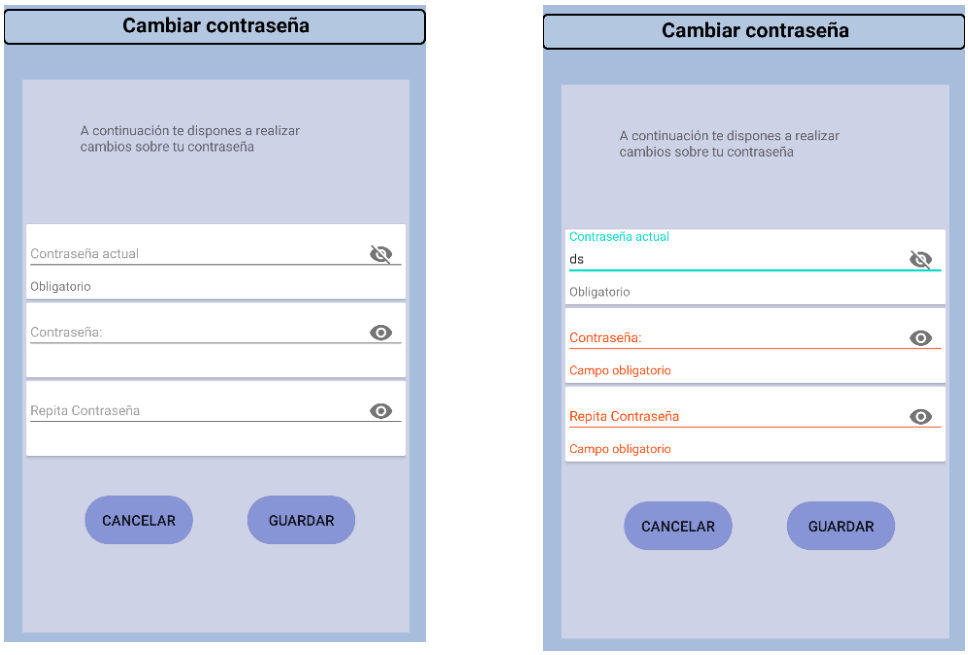


Figura 69. CAMBIO DE CONTRASEÑA

A continuación, se van a indicar las funcionalidades que pueden realizar los usuarios registrados.

Visualizar los puntos de interés

En este caso la única diferencia que existe respecto a los usuarios sin registrar es que aparece un nuevo icono que desbloquea la funcionalidad punto a punto.

Una vez desplegado el mapa se muestra la ubicación en tiempo real del usuario, en la Figura 70 se muestran los siguientes botones:

- Play (1): Permite iniciar el trayecto hacia la ubicación del punto de interés.
- Ver trayecto (2): Hace zoom alejando para poder visualizar la ruta completa.
- Centrar (3): Acerca la imagen para ver la ruta más cerca.
- Cancelar (4): Cancelar la navegación.

Figura 70. NAVEGACION A PUNTO A PUNTO

Visualizar las sendas

En este caso la única diferencia con el respecto a un usuario sin registrar es que se puede agregar sendas a favoritos y por ello en la pantalla principal de sendas aparece una nueva sección en la que irán apareciendo todas las sendas que se hayan agregado, como se ve en la Figura 71.

Figura 71. VISTA PRINCIPAL DE SENDAS

Al seleccionar Ver Sendas se despliega la pantalla en la que se muestra toda la información del punto de interés, aparece un icono de corazón si rellenar. Al presionarlo se despliega una alerta con dos opciones, como se ve en Figura 72:

- Continuar: Agrega la senda actual a la lista de favoritos del usuario, volviendo a la pantalla de información de senda, pero con la diferencia de que el icono de favoritos ahora es rojo.
- Cancelar: No agrega a la lista y vuelve a la pantalla de información de la senda.

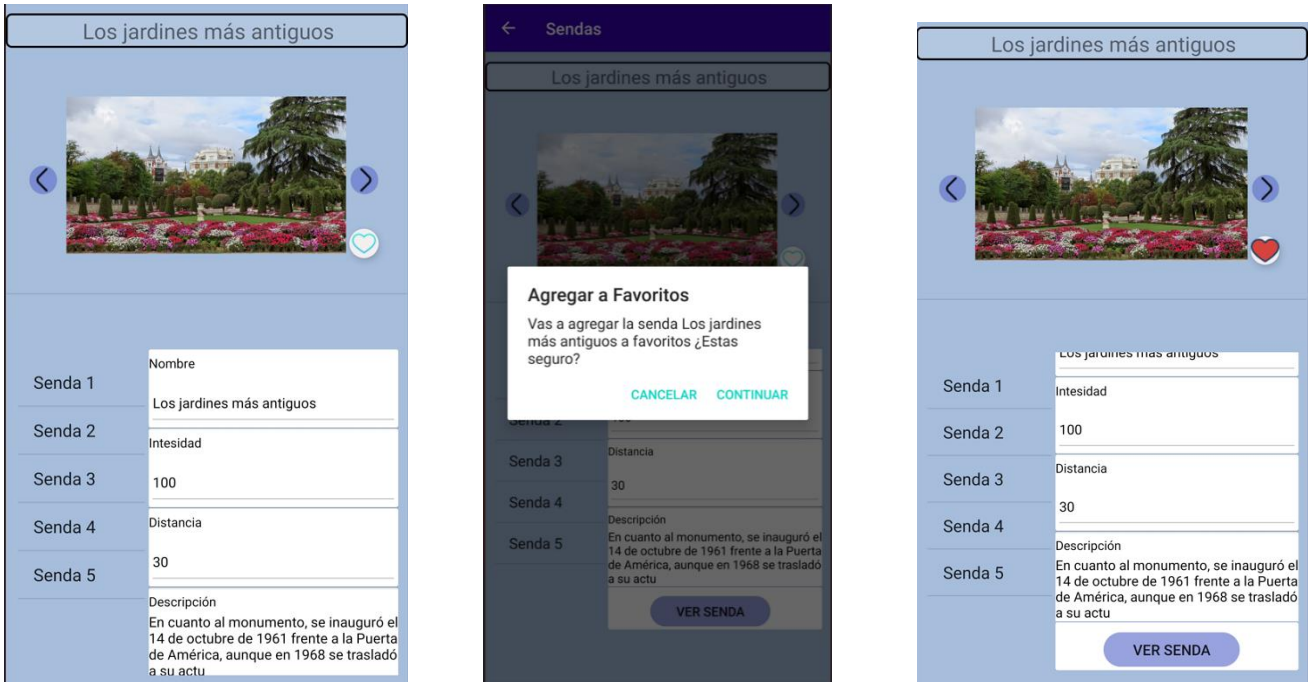


Figura 72. AGREGAR A FAVORITOS USUARIO REGISTRADO

Por otra parte, en la Figura 73 se muestra una vista en la cual se encuentra los siguientes elementos:

- Ítems favoritos: Lista de sendas que han sido agregadas a favoritos
- Eliminar ítem: Icono con forma de papelera cuya función es eliminar la senda

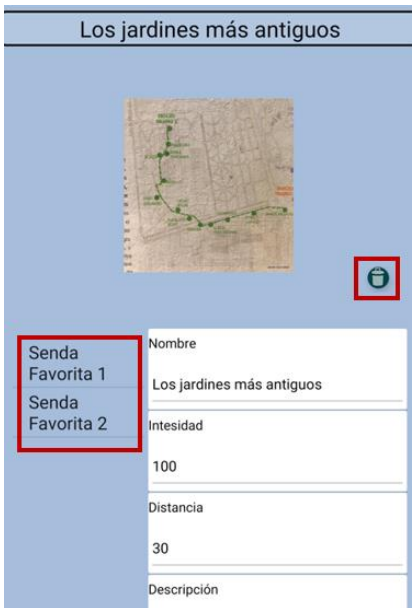


Figura 73. ELIMINAR SENDA DE FAVORITOS

Visualizar y administrar los puntos de interés

Dentro de la Figura 74 se encuentran una pantalla principal con las funcionalidades de administrador:

Figura 74. VISTA PRINCIPAL DE PUNTOS DE INTERES

Tras seleccionar Crear Punto se despliega un formulario en el que se tiene que rellenar con los datos que se van a ver en el sistema en caso de que estén vacíos, se informan por medio de mensajes.

Al continuar se debe añadir las coordenadas del punto, pueden consultar si es correcta por medio del botón de ver ubicación.

En caso de querer vaciar los campos se debe presionar el botón cancelar.

Este flujo se muestra en la Figura 75.

Figura 75. CREACION DE PUNTO DE INTERES

Al guardar la ubicación se desbloquea la opción de subir fotos, que se puede consultar en la Figura 76.

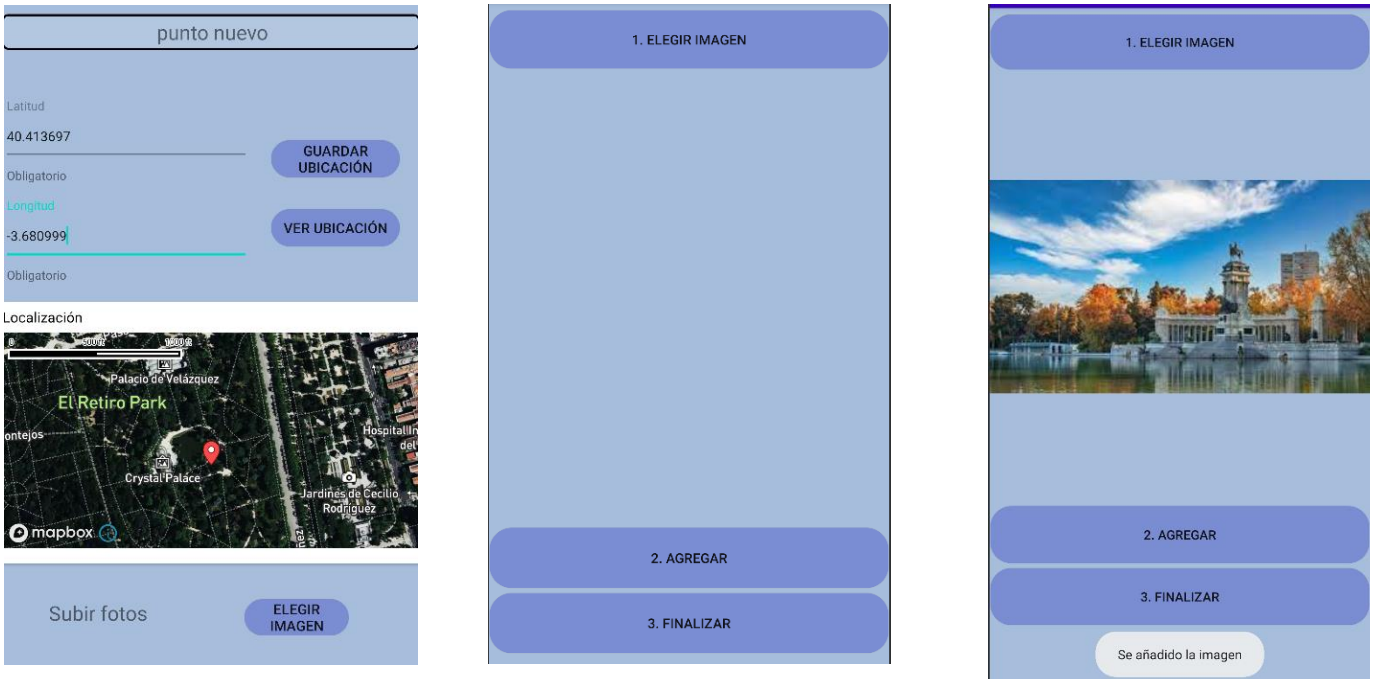


Figura 76. SUBIDA FOTOS DE PUNTO DE INTERES

La opción Modificar Punto que se muestra en la Figura 77 abre una vista en la que existen varios elementos: lista de ítems que permiten ir cambiando de punto; un cuadro de datos en el que se coloca toda la información y un apartado en el que se puede subir más fotos.

El funcionamiento es el siguiente: una vez se haya seleccionado un ítem se pueden realizar los cambios pertinentes y cuando se haya finalizado se debe dar al botón guardar.

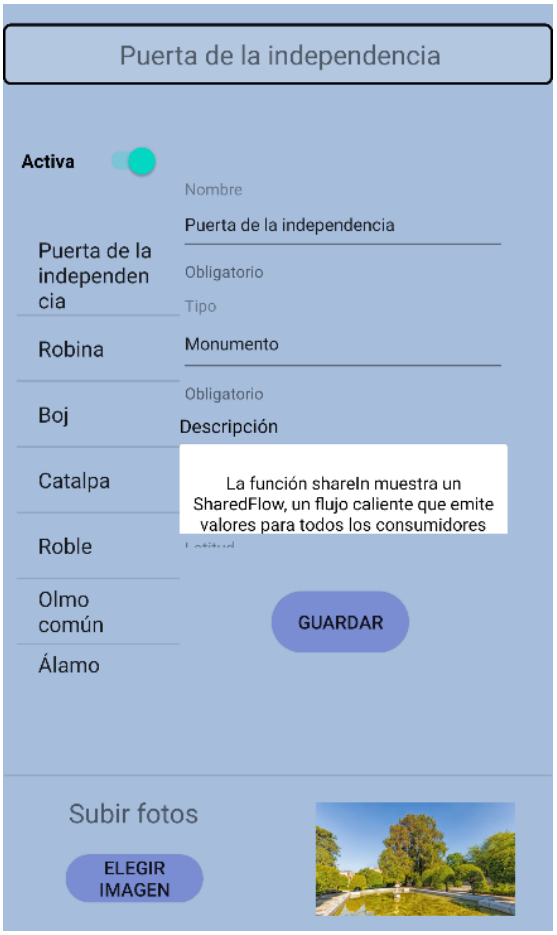


Figura 77. MODIFICACION DE PUNTO DE INTERES

En la opción Eliminar punto se encuentra una lista con todos los puntos que contiene el sistema. Primero se selecciona el punto seguido del botón eliminar.

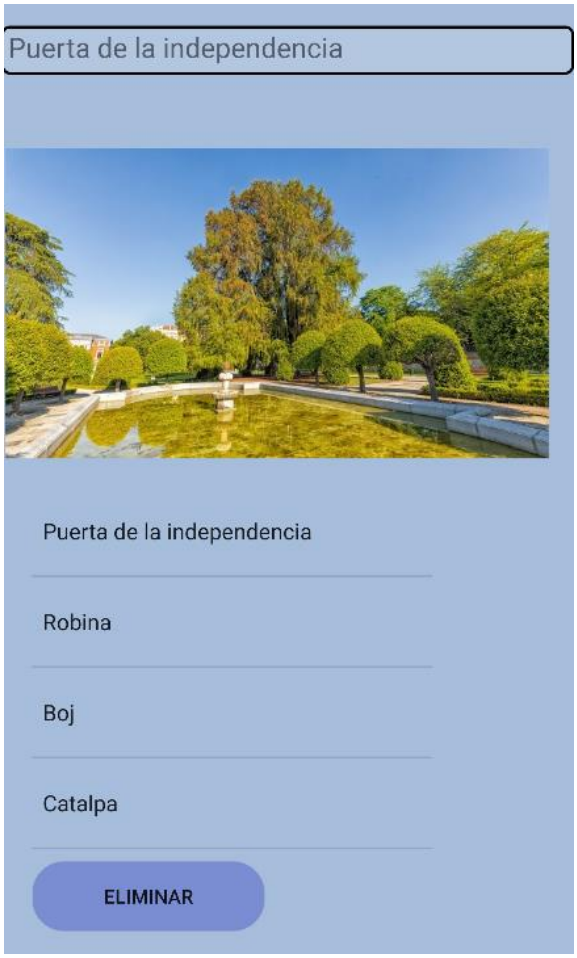


Figura 78. ELIMINACION PUNTO DE INTERES

Visualizar y administrar las sendas

En la Figura 79 se muestran las funcionalidades en las que las sendas son las protagonistas desde el punto de vista del administrador:

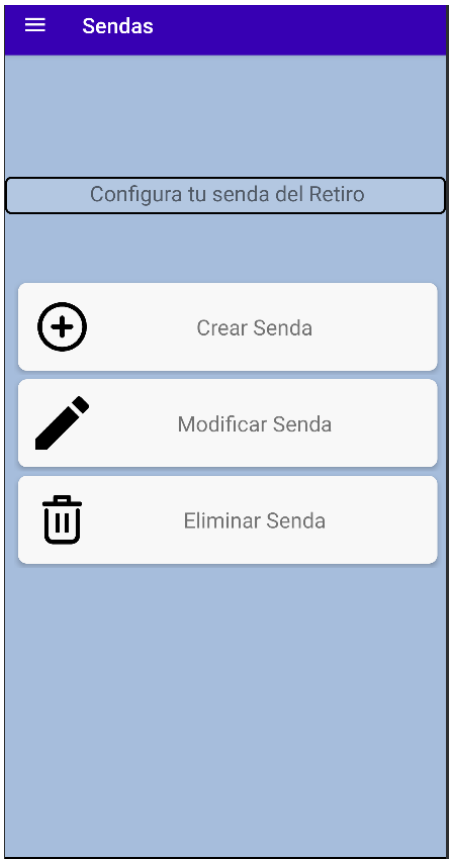


Figura 79. VISTA PRINCIPAL DE SENDAS DE ADMINISTRADOR

Con la opción de crear senda que se puede ver en la Figura 80, se muestra un formulario en el que lo primero es agregar los puntos que van a formar parte de la senda, para ello se muestran todos los puntos disponibles y un cuadro de información en el que se encuentra el tipo y la localización. Al acabar se presiona el botón finalizar, volviendo de esta manera al formulario de datos de la senda en el que tendremos que completarlo.

Finalmente creamos la ruta con el botón crear ruta, que desbloquea en la parte inferior de la pantalla la opción de subir fotos. Cuyo comportamiento es igual que en la creación de puntos.

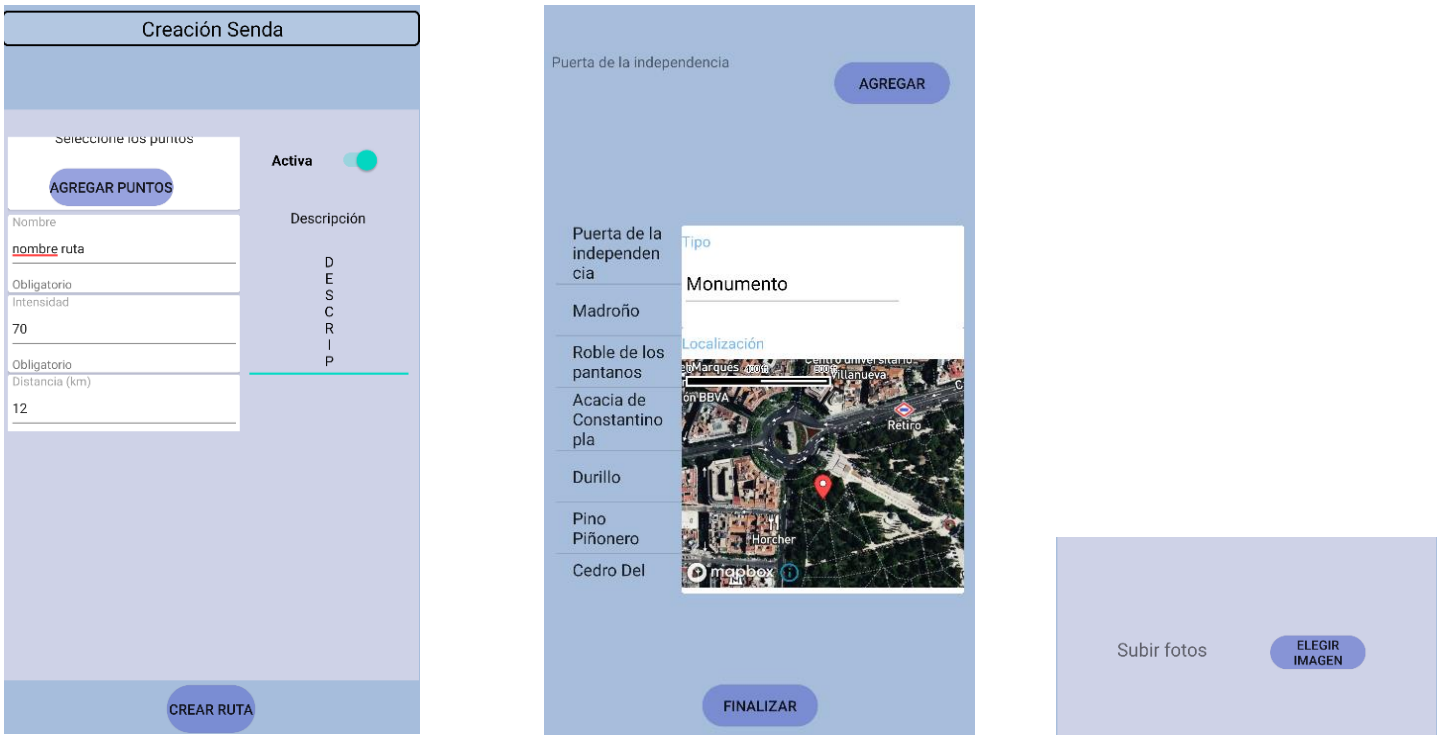


Figura 80. FLUJO DE CREACION DE SENDA

En la Figura 81 se muestra la pantalla de modificación de Senda, en la que se selecciona una y se realizan los cambios necesarios y al finalizar se presiona el botón modificar senda.

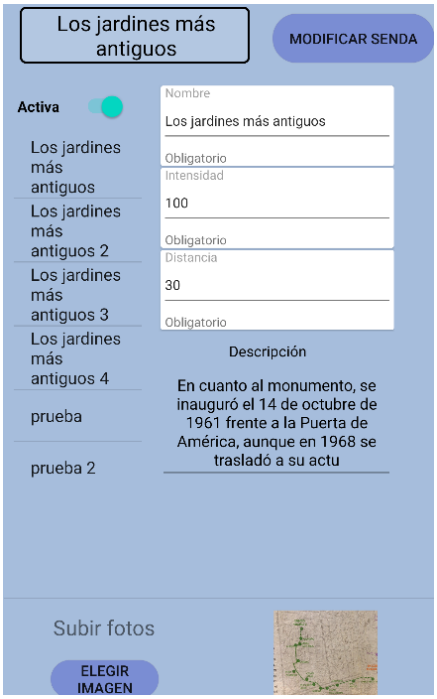


Figura 81. MODIFICACION DE SENDA

Para la opción de eliminar senda que se muestra en la Figura 82, se ha de seleccionar un ítem de la lista y continuación presionar el botón eliminar.

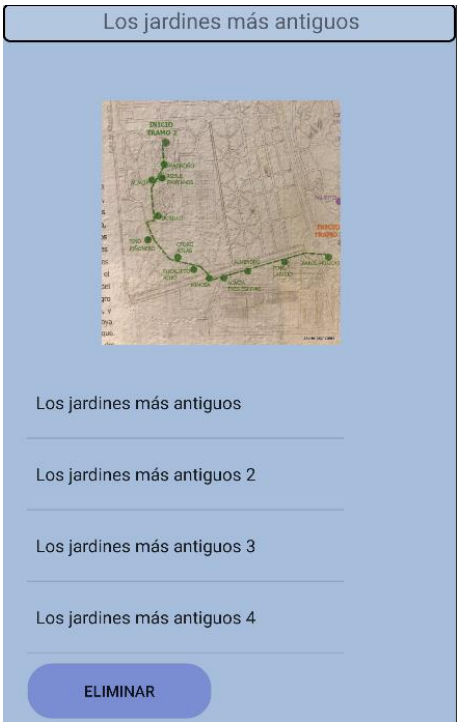


Figura 82. ELIMINACION DE SENDA

Apéndice B - Guía de instalación de la aplicación

Instalación entorno local

A lo largo de este apartado se va a explicar los pasos necesarios para la instalación y el correcto funcionamiento de la API y de la aplicación móvil.

Para la API es necesario tener la aplicación XAMPP instalada en la ruta "c:/xampp", ya que contiene un servidor apache que permite la ejecución de código php con la versión 8.2.0 o superior y el gestor de base de datos phpmyAdmin con la versión 5.2.0 o superior.

A continuación, se debe descargar el código del repositorio de GitHub y copiar el contenido de la carpeta ./backend de la en ruta "C:/xampp/htdocs/TFG/". Esto permite ejecutar el servidor y que pueda gestionar las peticiones.

El paso siguiente es crear la base de datos a través de un navegador, para ello se debe abrir la siguiente ruta https://localhost/phpmyadmin/index.php, crear un base de datos con el siguiente nombre TFG e importar el fichero BDTFG.sql que se encuentra en la carpeta ./BD dentro de los ficheros descargados, esto cargará la estructura de la base de datos.

Para la visualización de todo el código en Android Studio hay se debe importar el contenido de la carpeta ./tfg

Instalación aplicación en móvil

Finalmente hay que configurar el dispositivo en el que se va a probar, lo primero que se debe hacer es comprobar que tiene habilitada la instalación de orígenes desconocidos. Además, se tiene que volcar el fichero aplicacionLocal.apk que se encuentra en la carpeta ./aplicación del repositorio al dispositivo e instalarlo.

Una vez realizados estos pasos ya se puede usar la aplicación en un entorno local. Un dato importante es que no se puede crear administradores desde la aplicación por lo que hay que acceder a la tabla users de la base de datos y a través del gestor poner el atributo admin a 1.

Cabe resaltar que también se ha preparado un entorno en un servidor externo en https://x10hosting.com/login con el usuario: luisromantfg@gmail.com y la contraseña: AplicacionTFG2023 que ofrece un gestor de ficheros y de base de datos que se accede por medio del portal de la opción de hosting control panel la cual se puede acceder desde el dashboard.

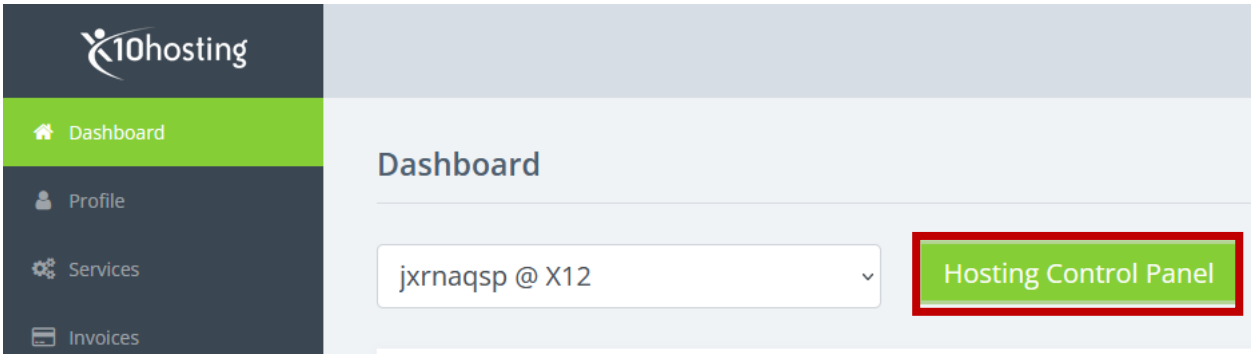


Figura 83. DASHBOARD HOSTING REMOTO

Los ficheros del servidor se pueden consultar dentro del gestor de ficheros en la siguiente ruta: public_html/TFG/

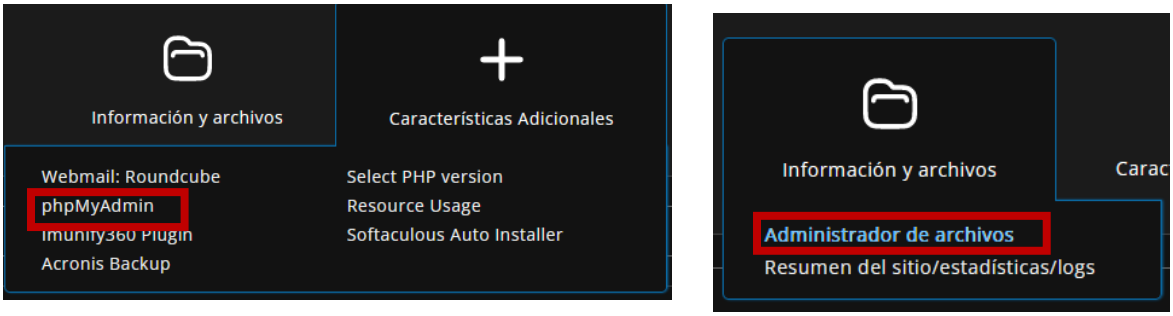


Figura 84. ADMINISTRADOR DE ARCHIVOS Y PHPMYADMIN

Del mismo modo se proporciona el fichero aplicaciónExterna.apk que apunta a este entorno y permite su uso sin necesidad de instalar xampp.