

Desarrollo de un sistema de servicios turísticos por geolocalización



TRABAJO DE FIN DE GRADO GRADO EN INGENIERÍA INFORMÁTICA

Realizado por:

David Rodríguez Marco

Director:

Antonio Sarasa Cabezuelo

Departamento de Sistemas Informáticos y Computación

Facultad de Informática

Universidad Complutense de Madrid

Madrid, septiembre 2018

Dedicatoria

“La belleza de un destino no es llegar hasta él, sino disfrutar del camino que te ha llevado a lograrlo”

Hay muchas metas en la vida, unas tardan más y otras tardan menos en lograrse, pero el objetivo de todas ellas es aprender del camino que te lleva hasta ellas. Aprender de los errores buscando soluciones. Ser decidido, creativo y realista a la hora de fijar la vista hacia un reto que se desea cumplir.

Después de todos estos años, uno echa la vista atrás para ver su evolución, sus cambios. Valora los momentos, tanto buenos como malos, porque al fin y al cabo el conjunto de ellos ha sido lo que le ha traído hasta el día de hoy.

Cuando uno comienza una aventura lo hace siempre con una mochila, una mochila llena de recursos, unos más valiosos que otros. Entendiendo esa carga se aprende a disfrutar del camino. Añadiendo o quitando esos recursos se llega a un equilibrio.

Quiero dedicarle el trabajo de todos estos años y este último a mi familia, a mis amigos de la universidad y a los de siempre. A todas las personas que han estado y están en mi vida, pues cada una de ellas me ha traído hasta aquí.

Gracias de corazón, por el apoyo y la dedicación. Y aunque esta aventura termina, empiezan nuevas metas que conquistar.

Agradecimientos

Quiero agradecer a mi tutor de Trabajo de Fin de Grado y profesor de la carrera Antonio Sarasa Cabezuelo el apoyo que me ha dado, desde la primera reunión, ayudándome a elegir un trabajo que se adaptara a lo que estaba buscando, aportándome ideas y dejándome libertad para ser creativo a lo largo del proyecto.

Quiero agradecer a la Universidad Complutense de Madrid y a la Facultad de Informática la ayuda que me ha dado durante todos estos años. A cada profesor que me ha ayudado con un tema o con alguna duda referente a tecnologías.

Y en último lugar, quiero agradecer a todas las personas que me han ayudado a realizar las pruebas de la aplicación móvil. Usando la app y dándome ideas y reportes de su uso.

A todas las personas e instituciones, gracias.

Resumen

Saber ubicarnos está a la orden del día. Saber dónde estamos para poder conocer todo lo que tenemos a nuestro alrededor es algo indispensable en nuestra rutina diaria. La informática y la tecnología constituye un papel fundamental a la hora de realizar estas acciones. Mediante la geolocalización podemos saber dónde está un sitio, cuánto tardaremos en llegar hasta el, etc.

El presente Trabajo de Fin de Grado hace uso de la tecnología de la geolocalización presente en los dispositivos, para descubrir los lugares emblemáticos de una ciudad, es decir, los monumentos que hay cerca del usuario con el fin de conocerlos más de cerca.

En el trabajo se han desarrollado dos aplicaciones interrelacionadas. Una aplicación que permite el mantenimiento de una página web desde la cual se puede acceder los monumentos gestionados por el sistema. El usuario de esta aplicación puede realizar operaciones tales como la introducción, modificación o eliminación de datos, descripciones, elementos multimedia o documentos relacionados con los monumentos gestionados.

Por otro lado, se tiene una app Android dónde el usuario puede recuperar información acerca de los monumentos gestionados por el sistema. Para hacer más atractivo el proceso, se ha implementado el acceso a la información como una acción de “descubrimiento”. En este sentido, el usuario irá descubriendo solo los monumentos que se encuentren en un radio máximo de 1km, de manera que sólo si el usuario de la app camina, podrá descubrir los monumentos que hay en la ciudad.

Esta memoria incluye todo el proceso de creación de la aplicación “GeoTulistApp”. Su especificación, arquitectura, diseño, implementación, algoritmos, conclusiones y trabajo futuro.

Palabras clave

Descubrir monumentos, Valorar lugares, Geolocalización, Aplicación web, Aplicación móvil, Android

Abstract

Knowing how to locate yourself is the order of the day. Knowing where we are to know everything we have around us is essential in our daily routine. Technology and IT have a fundamental role when carrying out these actions. Through geolocation we can know where a place is, how long it will take to get there, etc.

This Final Degree Paper uses technology of geolocation present in the devices to discover emblematic places of a city, in other words, the monuments that are close to the user in order to know them more closely.

In this paper, two interrelated applications have been developed. An application that allows the maintenance of a website from which you can access the monuments managed by the system. The user of this application can perform operations such as the introduction, modification or removal of data, descriptions, multimedia elements or documents related to the monuments managed.

On the other hand, there is an Android app where the user can retrieve information about the monuments managed by the system. To make the process more attractive, access to information has been implemented as a "discovery" action. This way, the user will discover only the monuments that are within a maximum radius of 1km, so that only if the user of the app walks, he can discover the monuments that are in the city.

This paper includes the entire process of creating the "GeoTulistApp" application. Its specification, architecture, design, implementation, algorithms, conclusions and future work.

Keywords

Discover monuments, Rate places, Geolocation, Web application, Mobile application, Android

Índice

Resumen	4
Abstract.....	5
Índice de figuras	9
1. Introducción	11
1.2 Estructura de la memoria	12
1.3 Propósito y objetivos.....	14
1.4 Estado del arte	15
1.1 Introduction	16
1.2.1 Memory structure	17
1.3.1 Purpose and objectives	19
1.4.1 State of the art	20
2. Especificación de la aplicación	21
2.1 Funcionalidades de la Aplicación Web	21
2.1.1 Casos de uso de la Aplicación Web.....	21
2.2 Funcionalidades de la Aplicación Móvil.....	24
2.2.1 Casos de uso de la Aplicación Móvil	24
3. Tecnología empleada	29
3.1 Tecnologías de la Aplicación Web	29
3.2 Tecnologías de la Aplicación Móvil.....	30
3.3 Otras tecnologías.....	30
4. Arquitectura de la aplicación.....	31
5. Modelo de datos	32
6. Diseño de la aplicación	38
6.1 Diseño de la Aplicación Web	38
6.2 Diseño de la Aplicación Móvil.....	44
7. Implementación de la aplicación.....	52
7.1 Implementación de la Aplicación Web	52
7.2 Implementación de la Aplicación Móvil.....	57
7.3 Base de datos	66
7.4 Algoritmos	66
8. Conclusiones y trabajo futuro	69
8.1 Conclusiones	69
8.1 Conclusions.....	69

8.2 Trabajo futuro	70
8.2 Future work	71
9. Aportaciones	72
Bibliografía	73
Apéndices.....	74
Apéndice 1: Manual de instalación	74
Apéndice 2: Manual de usuario.....	75
Código de las aplicaciones.....	78
Aplicación web	78
Aplicación móvil	78

Índice de figuras

FIGURA 1 - CASOS DE USO APLICACIÓN WEB.....	21
FIGURA 2 - LOGIN DE LA APLICACIÓN WEB.....	22
FIGURA 3 - ALTA DE UN LUGAR.....	22
FIGURA 4 - ALTA DE UN LUGAR AVANZADO.....	23
FIGURA 5 - GESTIÓN DE COMENTARIOS.....	24
FIGURA 6 - CASOS DE USO APLICACIÓN MÓVIL.....	25
FIGURA 7 - REGISTRO Y LOGIN DE LA APP.....	25
FIGURA 8 - MENÚ PRINCIPAL Y DESCUBRE.....	26
FIGURA 9 - VER LUGAR Y VER MULTIMEDIA.....	27
FIGURA 10 - VALORAR, VER PERFIL Y VER COMENTARIOS.....	27
FIGURA 11 - ARQUITECTURA.....	31
FIGURA 12 - BASE DE DATOS DE LA APLICACIÓN.....	32
FIGURA 13 - TABLA DE USUARIOS.....	33
FIGURA 14 - TABLA DE LUGARES.....	33
FIGURA 15 - TABLA DE USUARIOS - LUGARES.....	34
FIGURA 16 - TABLA DE VALORACIONES.....	34
FIGURA 17 - TABLA DE IMÁGENES.....	35
FIGURA 18 - TABLA DE DOCUMENTOS.....	35
FIGURA 19 - TABLA DE AUDIOS.....	35
FIGURA 20 - TABLA DE COMENTARIOS.....	36
FIGURA 21 - TABLA DE GESTIÓN.....	36
FIGURA 22 - TABLA DE INCIDENCIAS.....	36
FIGURA 23 - VISTA DE LOGIN DE LA APLICACIÓN WEB.....	38
FIGURA 24 - VISTA DE INICIO DE LA APLICACIÓN WEB.....	39
FIGURA 25 - VISTA DE MONUMENTOS DE LA APLICACIÓN WEB.....	39
FIGURA 26 - AÑADIR LUGAR.....	40
FIGURA 27 - AÑADIR LUGAR COMPLETO.....	40
FIGURA 28 - AÑADIR MONUMENTO CON CAMPOS COMPLETOS.....	41
FIGURA 29 - VER MONUMENTO Y ACTUALIZAR ELEMENTOS MULTIMEDIA.....	41
FIGURA 30 - ELIMINAR MONUMENTO/S.....	42
FIGURA 31 - COMENTARIOS DE LOS MONUMENTOS.....	42
FIGURA 32 - VALORACIONES DE LOS MONUMENTOS.....	43
FIGURA 33 - USUARIOS DE LA APLICACIÓN.....	43
FIGURA 34 - INCIDENCIAS DE LA APLICACIÓN.....	44
FIGURA 35 - VISTA DE LOGIN DE LA APP.....	44
FIGURA 36 - VISTA DE REGISTRO DE LA APP.....	45
FIGURA 37 - VALIDACIÓN DE LOS FORMULARIOS DE LOGIN Y REGISTRO.....	46
FIGURA 38 - PANTALLA PRINCIPAL DE LA APP.....	46
FIGURA 39 - MENÚ LATERAL DE LA APP.....	47
FIGURA 40 - VISTA DE MIS MONUMENTOS EN LA APP.....	47
FIGURA 41 - VISTA DE USO DE UBICACIÓN Y DESCUBRE MONUMENTOS EN LA APP.....	48
FIGURA 42 - VISTA DE UN MONUMENTO EN LA APP.....	48
FIGURA 43 - VISTAS DE VALORAR Y COMENTAR UN MONUMENTO.....	49
FIGURA 44 - VISTA DE ELEMENTOS MULTIMEDIA DE UN MONUMENTO.....	50
FIGURA 45 - VISTA DE MI PERFIL DE LA APP.....	50
FIGURA 46 - VISTA DE CERRAR SESIÓN DE LA APP.....	51
FIGURA 47 - CÓDIGO DE LOGIN.....	52
FIGURA 48 - CÓDIGO DE INICIO SECCIONES.....	53
FIGURA 49 - CÓDIGO DE ACTUALIZAR DATOS.....	53

FIGURA 50 - CÓDIGO DE OBTENER MONUMENTOS	54
FIGURA 51 - CÓDIGO DE ELIMINAR LUGARES.....	54
FIGURA 52 - CÓDIGO DE VER MONUMENTO	54
FIGURA 53 - CÓDIGO DE OBTENER MULTIMEDIA	55
FIGURA 54 - CÓDIGO DE OBTENER VALORACIONES	55
FIGURA 55 - CÓDIGO DE ALTA DE LUGAR	56
FIGURA 56 - ESTRUCTURA APP	57
FIGURA 57 - PERMISOS APP.....	57
FIGURA 58 - ESTRUCTURA CLASES JAVA	57
FIGURA 59 - ESTRUCTURA RES	58
FIGURA 60 - SHARED PREFERENCES	58
FIGURA 61 - REALIZAR LOGIN.....	58
FIGURA 62 - REALIZAR REGISTRO	59
FIGURA 63 - MENÚ APP	59
FIGURA 64 - BOTONES INICIO.....	60
FIGURA 65 - URL OBTENER MONUMENTOS DE USUARIO	60
FIGURA 66 - MÉTODO GETJSON.....	61
FIGURA 67 - VOLCADO EN LISTVIEW.....	62
FIGURA 68 - CÓDIGO REALIZAR VALORACIÓN Y COMENTARIOS	63
FIGURA 69 - CÓDIGO VER MONUMENTO	63
FIGURA 70 - CÓDIGO BOTONES MULTIMEDIA Y VALORACIÓN	64
FIGURA 71 - CÓDIGO VER PERFIL	65
FIGURA 72 - CÓDIGO CERRAR SESIÓN.....	65
FIGURA 73 - CONEXIÓN CON BBDD	66
FIGURA 74 - OBTENER COORDENADAS	67
FIGURA 75 - CAMBIO DE LOCALIZACIÓN.....	67
FIGURA 76 - ACTIVIDAD SENSOR GPS.....	68
FIGURA 77 - OBTENER COORDENADAS USUARIO.....	68
FIGURA 78 - COMPARACIÓN DE COORDENADAS	68

1. Introducción

Este Trabajo de Fin de Grado surgió con la idea de realizar una aplicación enfocada al turismo que aprovechara las últimas tecnologías que permiten explotar la geolocalización. Así, la idea principal es motivar a los usuarios a explorar los diferentes lugares de una ciudad para descubrir monumentos emblemáticos y poder acceder a información acerca de ellos gestionada a través del sistema creado. El sistema consta de dos aplicaciones interrelacionadas y con dos objetivos diferentes.

La parte de la aplicación web permite a un administrador enriquecer la base de datos con la información de los monumentos y lugares que gestiona el sistema. La información podrá incluir una descripción del monumento o lugar, los horarios de apertura y cierre, imágenes que se mostrarán en la aplicación móvil, documentos de información como pdfs, excels o archivos de sonido con descripciones acerca de los monumentos, audio guías y otra información referente a los monumentos. Además, desde la aplicación web se podrán gestionar tanto la información de los monumentos y lugares como la información referente a los usuarios. Así mismo, se podrá acceder a estadísticas y valoraciones realizadas por los usuarios de la aplicación.

Por otro lado, la aplicación móvil permite a un usuario explotar la información de los monumentos almacenada en la base de datos. Haciendo uso del sensor de posicionamiento del dispositivo móvil el sistema comprobará que lugares y monumentos hay cerca del usuario. El sistema ofrece al usuario los distintos lugares que puede encontrar, ahorrando tiempo y facilitando su búsqueda.

El campo de la geolocalización es muy extenso y en este trabajo se aborda la herramienta con un fin específico: *Encontrar monumentos cercanos dada una posición*. Una gran ventaja de la geolocalización es poder determinar en tiempo real la ubicación en la que el usuario se encuentra. Además, permite conocer los lugares que hay cerca y explorar su información

La manera en la que se enfoca este proyecto es el de motivar a las personas a salir a la calle, descubrir, conocer y aprender de todo lo que tienen a su alrededor. Haciendo que se compita de una forma sana, conociendo incluso a otros usuarios a medida que se exploran los distintos monumentos. Aporta conocimiento sobre los lugares que hay en una ciudad apostando por descubrir lugares a pie y conociendo la zona en la que están.

1.2 Estructura de la memoria

La memoria se estructura de la siguiente manera:

- **Introducción**

En este apartado se explica de forma breve de que trata el proyecto, se realiza un resumen y se establece la estructura básica de cómo está organizado el proyecto.

- **Especificación**

En este apartado se trata la especificación de cada una de las funcionalidades de las dos aplicaciones implementadas.

- **Tecnología empleada**

En este punto de la memoria se describen las herramientas y tecnologías utilizadas en el desarrollo del proyecto.

- **Arquitectura de datos**

En este apartado se trata cómo está organizada la aplicación a nivel de arquitectura, y los elementos que interactúan entre sí.

- **Modelo de datos**

En este apartado se describe la estructura completa de la base de datos, las tablas que la componen y las relaciones que hay entre ellas.

- **Diseño e Implementación**

En este apartado se trata el desarrollo a nivel de diseño, los patrones utilizados, bocetos, y herramientas utilizadas para ambas aplicaciones. Además, se explica en detalle la implementación del proyecto, cómo se ha construido a nivel interno, el desarrollo a nivel de código, las relaciones entre las aplicaciones, y los algoritmos utilizados.

- **Conclusiones y Trabajo futuro**

En este apartado de la memoria se plantean las conclusiones obtenidas a partir de la realización del proyecto y las líneas de trabajo futuro que existen en el proyecto.

- **Aportaciones**

En este apartado se explican en detalle las aportaciones del creador del proyecto y cuales han sido las decisiones tomadas.

- **Bibliografía**

En este apartado se establecen las referencias utilizadas en la memoria del proyecto.

- **Apéndices**

En el último apartado de la memoria se adjunta el manual de usuario y el manual de instalación.

1.3 Propósito y objetivos

El objetivo principal de este Trabajo de fin de Grado es el desarrollo de un sistema formado por una aplicación web y una aplicación móvil que permita a los usuarios descubrir los monumentos y lugares que hay cerca de ellos. Para ello se aprovecha las posibilidades que ofrece la geolocalización para conseguirlo. Este objetivo se puede refinar en los siguientes objetivos más específicos:

- Crear una aplicación web que permita a un usuario administrador mantener toda la información que el sistema gestiona: añadiendo, actualizando o borrando esta información.
- Desarrollar una aplicación móvil que explote toda la información de la base de datos y muestre a los usuarios la información sobre los distintos lugares y monumentos que existen en la aplicación.
- La aplicación web permitirá al administrador ver y gestionar en tiempo real toda la información acerca de los usuarios registrados, las valoraciones y visitas de monumentos.
- La aplicación móvil será intuitiva y fácil de usar de manera que el usuario se sienta cómodo y pueda usarla sin problemas.

1.4 Estado del arte

En general las aplicaciones que hay en el mercado son simplemente guías de ciudades en las que se indican distintos lugares y monumentos, pero sin usar de manera intensiva las posibilidades que ofrece la geolocalización. En la investigación que se ha realizado se han encontrado diversas apps que hacen uso de la geolocalización y muestran información de monumentos.

- **Pokemón Go [10]:**

Aplicación que hace uso de la geolocalización para capturar criaturas a través de la realidad aumentada. Se explora la ciudad en busca de nuevas criaturas que coleccionar.

- **Google Goggles [11]:**

Aplicación que muestra información sobre productos y monumentos a través de la cámara del móvil.

- **Myooni Guía turística global [12]:**

Guía turística para explorar los monumentos que hay alrededor del mundo. Pudiendo conocer los que hay en otros continentes.

- **Visit A City [13]:**

Aplicación para crear itinerarios y establecer los monumentos que se quieren ver en una ciudad.

- **Monumento Guida Turística [14]:**

Aplicación que muestra una extensa lista de monumentos con información e imágenes.

1.1 Introduction

This Final Degree Project came up with the idea of making an application focused on tourism that takes advantage of the latest technologies that allow exploiting geolocation. Thus, the main idea is to motivate users to explore different places of a city to discover emblematic monuments and be able to access information about them managed through the created system. The system consists of two interrelated applications and with two different goals.

The part of the web application allows an administrator to enrich the database with the information of the monuments and places managed by the system. The information may include a description of the monument or place, the opening and closing times, images that will be displayed on the mobile application, information documents such as pdfs, excels or sound files with descriptions about the monuments, audio guides and other information referring to the monuments. In addition, from the web application you can manage both the information of the monuments and places as well as the information referring to the users. Also, you can access statistics and assessments made by users of the application.

On the other hand, the mobile application allows an user to exploit the information of the monuments stored in the database. By making use of the positioning sensor of the mobile device, the system will verify which places and monuments are close to the user. The system offers the user different places they can find, saving time and facilitating their search.

The geolocation field is very extensive and in this project the tool is addressed with a specific purpose: Find nearby monuments given a position. A great advantage of geolocation is being able to determine in real time the location in which the user is located. In addition, it allows you to know the places nearby and explore their information.

The way in which this project is focused is to motivate people to go out on the street, discover, know and learn from everything they have around them. Making it competition in a healthy way, even getting to know other users as the different monuments are explored. It improves knowledge about places that are in a city betting on discovering places by foot and knowing the area in which they are located.

1.2.1 Memory structure

The memory is structured as follows:

- **Introduction**

This section briefly explains what the project is about, summarizes and establishes the basic structure of how the project is organized.

- **Specification**

This section deals with the specification of each of the functionalities of our two applications. Both the web part and the mobile device.

- **Technology used**

In this point of the memory the tools and technologies used in the development of the project are described.

- **Data architecture**

This section deals with how the application is organized at the architecture level, and the elements that interact with each other.

- **Data model**

This section describes the complete structure of the database, the tables that compose it and the relationships between them.

- **Design and Implementation**

This section deals with all the development at the design level, the patterns used, sketches, tools used for both applications.

In addition, it explains in detail the implementation of the project, how it has been built internally, the development at the code level, the relationships between the applications, the algorithms used and the technologies used.

- **Conclusions and future work**

In this section of the report, conclusions are drawn after the completion of the project and future lines of work that exist in the project.

- **Contributions**

In this section the contributions of the creator of the project are explained in detail. What have been their decisions when developing it.

- **Bibliography**

In this section, the references used throughout the development of the project are established.

- **Appendices**

The user manual and the installation manual are attached in the last section of the report. In addition to referring to the code hosted in a repository of both applications.

1.3.1 Purpose and objectives

The main objective of this Final Degree Project is the development of a system consisting of a web application and a mobile application that allows users to discover the monuments and places that are close to them. To do this, it takes advantage of the possibilities offered by geolocation to achieve it.

This objective can be refined in the following more specific objectives:

- Create a web application that allows an administrator user to keep all the information that the system manages. Adding, updating or deleting this information.
- Develop a mobile application that exploits all the information in the database and shows users the different places and monuments that exist in the application.
- Can see registered users, valuations and visits of monuments in real time. You can manage all the data from this application to be a complete tool and thus avoid the administrator having to touch the database or other information.
- The mobile application will be the tool with which the user will interact, and can perform all the functions described above. In addition, it should be an intuitive and easy to use application so that the user feels comfortable and can use it without problems.

1.4.1 State of the art

Regarding the applications that there are in the market, there are many of them that are simply guides of cities in which different places and monuments are indicated, but there is no application that allows to discover specific places or monuments by geolocation giving information and utilities on them .

The geolocation functionality if it is found in other applications in which the user moves in the real world and sees on the map what the application offers. There are also numerous applications on the market. In the research that has been carried out, several apps have been found that use geolocation and show monument information.

- **Pokemon Go** [10]:

Application that uses geolocation to capture creatures through augmented reality. The city is explored in search of new creatures to collect.

- **Google Goggles** [11]:

Application that shows information about products and monuments through the mobile camera.

- **Myooni Global Tourism Guide** [12]:

Tourist guide to explore the monuments around the world. Being able to know the ones that exist in other continents.

- **Visit A City** [13]:

Application to create itineraries and establish the monuments that you want to see in a city.

- **Guida Turística Monument** [14]:

Application that shows an extensive list of monuments with information and images.

It has not been found any application that does the same as the one developed in this project.

2. Especificación de la aplicación

En esta sección se van a presentar las especificaciones de las dos aplicaciones implementadas. Para cada una de ellas, se mostrarán los diagramas de casos con los diferentes casos de uso, y a continuación se desarrolla cada caso de uso.

2.1 Funcionalidades de la Aplicación Web

En este apartado se detallan las funcionalidades referentes a la aplicación web mostrando los casos de uso y las funcionalidades desarrolladas.

2.1.1 Casos de uso de la Aplicación Web

La aplicación web está enfocada al usuario administrador, el cual podrá incluir nuevos lugares y monumentos, modificar los existentes (actualizando su información multimedia o eliminándola). También podrá ver los comentarios y estadísticas que han hecho los usuarios de la aplicación móvil.

A continuación, se explican las siguientes funcionalidades que ofrece la aplicación web.

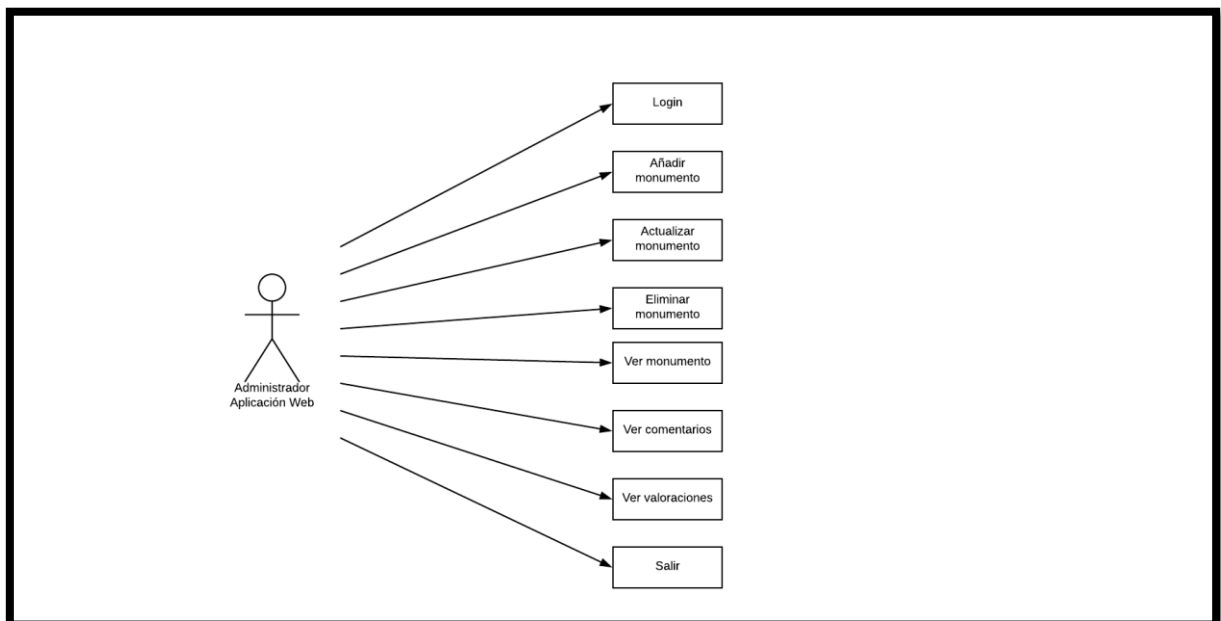


Figura 1 - Casos de Uso Aplicación Web

- **Login:** En la aplicación web habrá un único usuario, el administrador, que deberá logarse mediante un formulario (ver Figura 2) introduciendo usuario y contraseña para poder acceder a la aplicación. En el caso de que los campos estén vacíos o sean incorrectos, se informará al usuario del error.

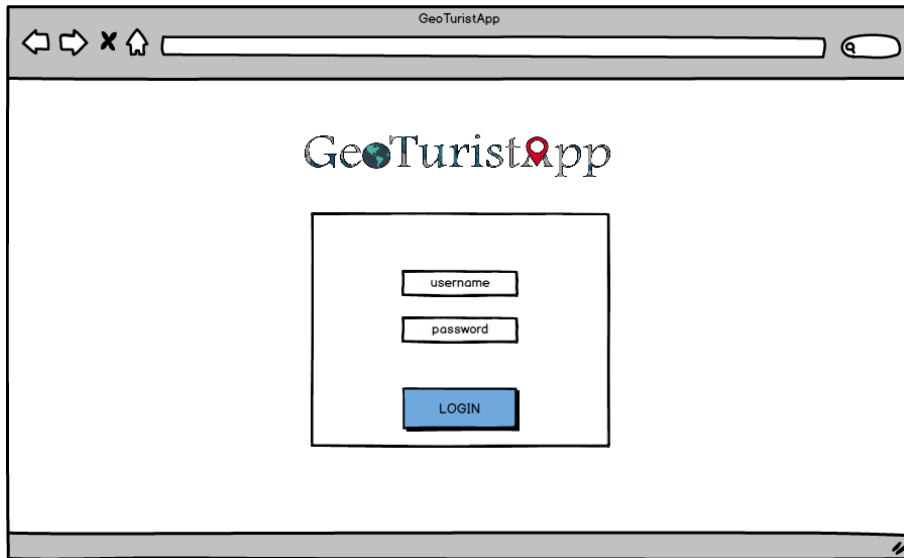


Figura 2 - Login de la aplicación web

- Añadir monumento/lugar:** Esta funcionalidad es una de las más relevantes que realizará el administrador de la aplicación web. Mediante un cuadro de búsqueda se buscará el monumento que se desea introducir en la aplicación, basándonos en la latitud y longitud. El nombre, características, descripción, horario de apertura y cierre, etc. Además, el administrador podrá subir elementos multimedia (imágenes, audios y documentos) que serán mostrados una vez el monumento sea seleccionado en la aplicación móvil por el usuario turista. Estos elementos multimedia se subirán mediante un formulario básico.

En el formulario (ver Figura 3), en el caso de no introducir algún campo, se informará al usuario del error que ha sucedido. No serán obligatorios los campos multimedia, ya que los monumentos pueden tener o no elementos multimedia.

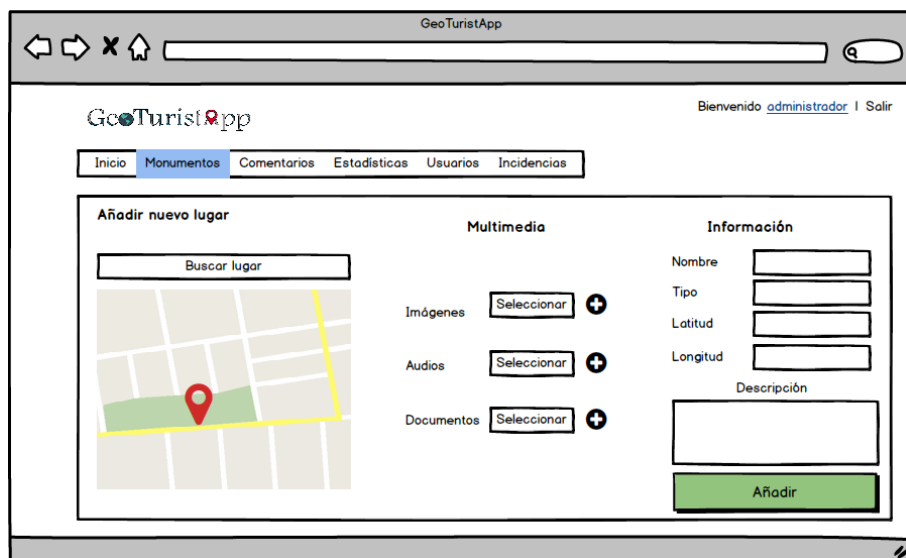


Figura 3 - Alta de un lugar

- Actualizar monumento/lugar: El administrador podrá editar la información almacenada de cada monumento o lugar (Ver Figura 4). Así mismo, podrá subir nuevos elementos multimedia o eliminarlos, actualizar la posición de un monumento o lugar, nombre, descripción o cualquier otro aspecto anteriormente almacenado.

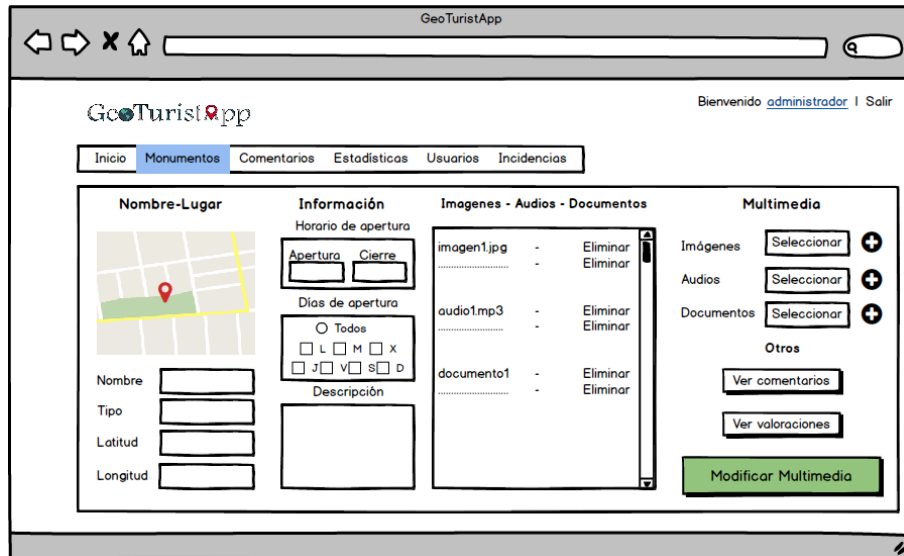


Figura 4 - Alta de un lugar avanzado

- Eliminar monumento/lugar: Se podrán eliminar los monumentos o lugares almacenados en la base de datos. Por ejemplo, si un museo cierra y ya no admite visitas, o en el caso de que toda la información no sea correcta y sea más sencillo eliminar el monumento completamente.
- Ver monumento: El administrador podrá ver todos los monumentos almacenados en la base de datos. Además, podrá ver la información general y específica de cada uno de los monumentos o lugares guardados.
- Ver comentarios: Una de las pestañas de la aplicación web corresponde al apartado de comentarios que los usuarios dejan en la aplicación de cada monumento o lugar (ver Figura 5). Además, el administrador también podrá eliminar comentarios si fuera necesario.

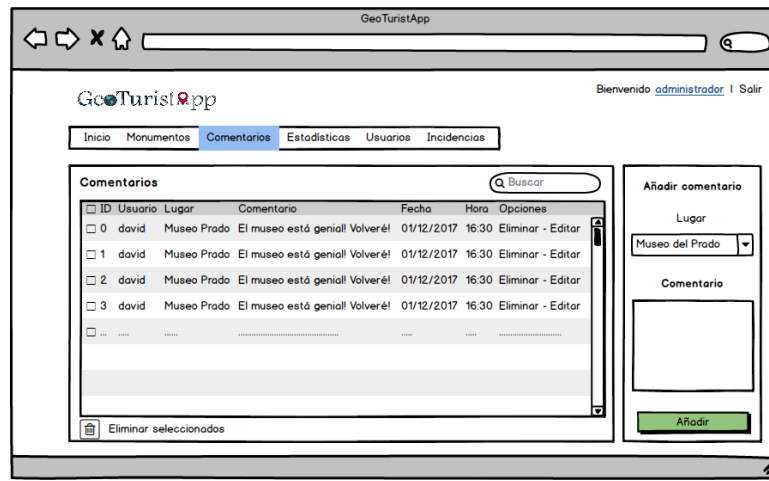


Figura 5 - Gestión de comentarios

- **Ver valoraciones:** En esta vista el administrador podrá ver qué monumentos tienen más visitas, más valoraciones y comentarios. Se podrá generar un cuadro de estadísticas para ver que monumento/lugar es el que más afluencia de visitas tiene. Se realiza una media de las valoraciones dependiendo de cuántos usuarios la valoren.
- **Salir:** Mediante ella se cierra la sesión de la aplicación web.

2.2 Funcionalidades de la Aplicación Móvil

En este apartado se detallan las funcionalidades referentes a la parte móvil de la aplicación, mostrando los casos de uso y las funcionalidades desarrolladas explicando cada una de ellas.

2.2.1 Casos de uso de la Aplicación Móvil

Mediante la aplicación móvil el usuario “turista” podrá ver los diferentes monumentos que hay a su alrededor y corresponden con los que están registrados en la base de datos. Se mostrará el enlace al monumento cercano y al pulsar en él se podrá ver la información de este además de sus elementos multimedia (si los tiene), podrá tener acceso a los monumentos que ha visitado y seguir sus estadísticas personales, y podrá realizar comentarios en los monumentos/lugares visitados.

A continuación, detallaré las funcionalidades anteriormente descritas.

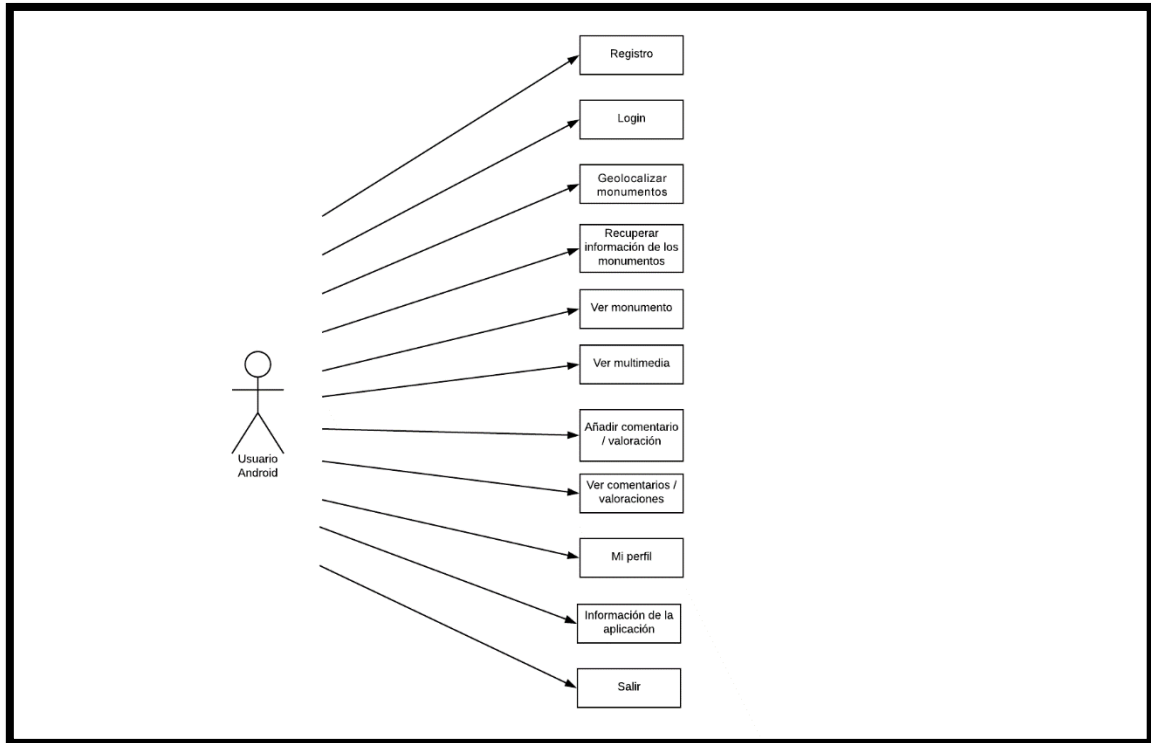


Figura 6 - Casos de Uso Aplicación Móvil

- **Registro en la APP:** El usuario dispondrá de un formulario (Ver Figura 6) en el que introducirá datos de carácter personal. Un nick de usuario único, que será la clave principal por la que se identificará a los usuarios, un password, el correo electrónico, el nombre y apellidos. Una vez realizado el registro los datos serán almacenados en la base de datos. Si alguno de estos datos no se rellena o es incorrecto se mostrará al usuario un mensaje de error.

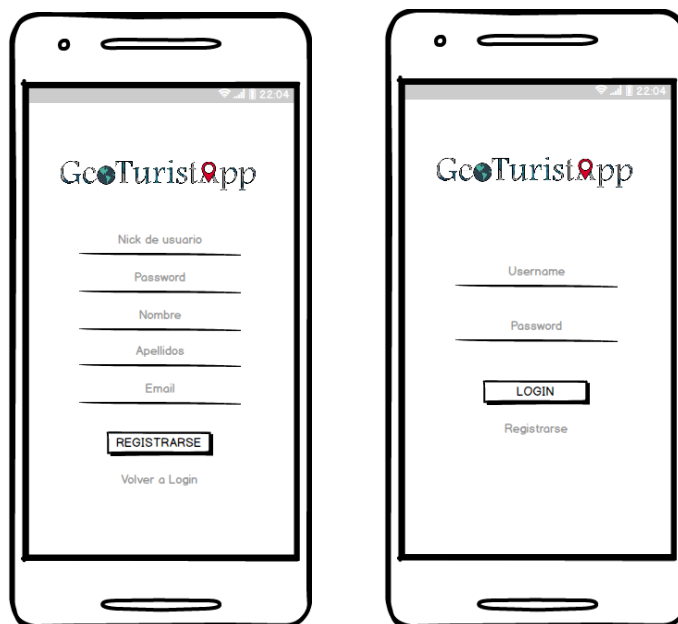


Figura 7 - Registro y Login de la App

2. Especificación de la aplicación

- Login en la APP: El usuario se autenticará en la aplicación con el nick y el password que utilizó en el registro (Ver Figura 7). Si el login es correcto se redirige al usuario a la pantalla principal de la APP. Si es incorrecto se mostrará un mensaje de error al usuario.
- Geolocalizar monumentos: El usuario podrá comprobar el estado del GPS obteniendo la latitud y longitud actual. De este modo podrá comprobar si se reciben datos de la geolocalización y automáticamente se mostrarán en la lista los monumentos que están cerca de la posición del usuario. La distancia a partir de la que se encontrarán los monumentos será máxima de 1km desde la posición del usuario. Si no hay ningún monumento cerca, no se mostrará nada en la lista. (Ver Figura 8)
- Recuperar información de los monumentos: A través del botón de “Descubrir Monumentos” el usuario podrá ver los monumentos cercanos a él gracias a la geolocalización. Se mostrará en forma de lista y al pulsar sobre alguno de ellos se redirigirá al monumento para ver su información, comentarios, valoraciones y elementos multimedia.

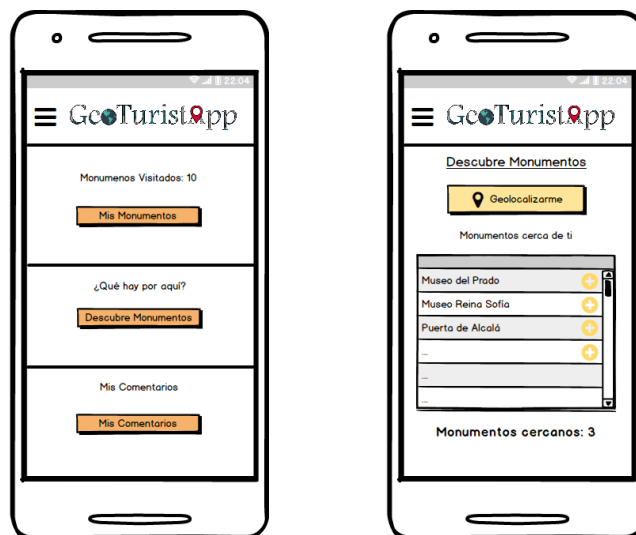


Figura 8 - Menú principal y Descubre

- Ver monumento: El usuario podrá ver toda la información almacenada en la BBDD del monumento seleccionado. Se mostrará la descripción del monumento, la información básica de apertura y cierre, cuántas veces se ha visitado, los comentarios que posea, si se desea añadirlo a los monumentos visitados y dos botones para dirigirse al apartado de multimedia y otro que redirigirá al de valoraciones del monumento actual.
- Ver Multimedia: El usuario podrá ver los archivos multimedia que posee el monumento que se está visitando, imágenes, audios y documentos. Podrá descargarlos y verlos si así lo desea. (Ver Figura 9)

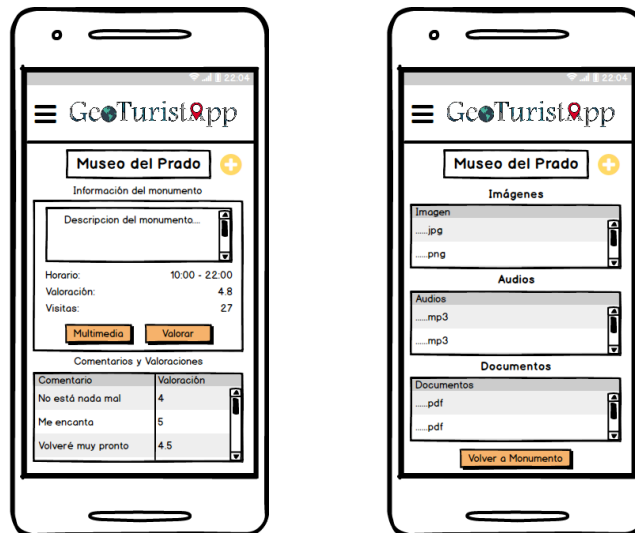


Figura 9 - Ver lugar y ver multimedia

- **Añadir comentario/valoración:** Dentro de la vista del monumento seleccionado se podrá incluir una valoración y un comentario si el usuario así lo desea (Ver Figura 10). Esta información se guardará en la BBDD, en la tabla de valoraciones y de comentarios. En esta vista también se mostrará la valoración actual del monumento.
- **Ver comentarios:** Estas acciones se podrán realizar desde la vista de Inicio. A través de un botón el usuario podrá acceder a los comentarios que ha realizado y en la vista de "Ver Monumento" podrá ver los comentarios de todos los usuarios que hayan comentado el monumento seleccionado. (Ver Figura 10).
- **Mi perfil:** Se podrá ver toda la información personal del usuario, el nick, el nombre y apellidos, el email, etc. Además, se mostrará la información referente a las valoraciones y comentarios realizados por el usuario. (Ver Figura 10).

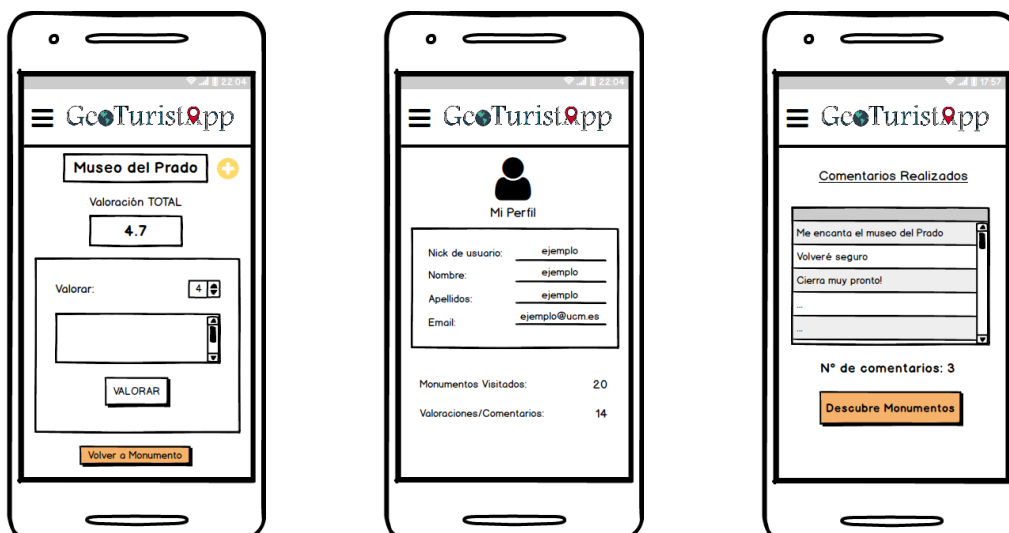


Figura 10 - Valorar, ver perfil y ver comentarios

2. Especificación de la aplicación

- Información de la aplicación: Aquí se mostrará la información referente a la aplicación, el autor, la versión actual y los futuros cambios que se implementarán.
- Salir de la APP: Mediante el botón del menú lateral de cerrar sesión se saldrá saldremos de la aplicación y se redirigirá a la pantalla de login.

3. Tecnología empleada

En esta sección se hablará de las tecnologías utilizadas en todo el proceso de desarrollo de la aplicación. En primer lugar, se explicarán las tecnologías utilizadas en el entorno web.

3.1 Tecnologías de la Aplicación Web

1) **XAMPP**

En la aplicación web y lanzarla utilizamos XAMPP [2], plataforma que incorpora un servidor Apache, MySQL, PHP y Pearl. Utilizaremos el gestor de PHP My Admin para ver las tablas que tenemos en nuestra base de datos MySQL. Para la conexión con la BBDD se ha utilizado PHP.

2) **PHP**

Lenguaje de código abierto [3] enfocado para el desarrollo web. El código generado se ejecuta en el servidor y es muy sencillo de utilizar. Se basa en la programación de scripts y es muy utilizado para la creación de páginas y aplicaciones web.

3) **Apache**

Servidor web HTTP que se utiliza para ejecutar aplicaciones desarrolladas en diversos lenguajes de programación [2].

Las versiones de las tecnologías mencionadas son las que se muestran en el siguiente cuadro:

1. La versión de XAMPP utilizada ha sido:	7.1.10
2. La versión de Apache utilizada ha sido:	2.4.28
3. La versión de PHP utilizada ha sido:	7.1.10

Para la parte Front-end de la aplicación se han utilizado las tecnologías HTML5, para la maquetación, creación de tablas, formularios y diseño web. CSS3 para dar estilo a todo lo generado con HTML5 y JavaScript, lenguaje de scripting para dar funcionalidad a los componentes HTML5 que son estáticos. Para apostar por un diseño genérico se han utilizado librerías Bootstrap y se ha apostado por un diseño simple e intuitivo apostando por una buena accesibilidad.

3.2 Tecnologías de la Aplicación Móvil

1) **Android Studio**

La aplicación móvil es nativa y desarrollada para la plataforma de Android [1]. El lenguaje de programación en Java y se ha utilizado el SDK de Android. El IDE utilizado para su desarrollo ha sido Android Studio en su versión 3.0.1. Al ser software open source y tener una gran comunidad detrás [8], se ha optado por esta tecnología ya que el aprendizaje ha sido bastante rápido.

3.3 Otras tecnologías

1) **GitHub**

Para el control de versiones tanto para la aplicación web como para la móvil se ha utilizado el servidor de GitHub [16] para almacenar el código y mantener la aplicación actualizada.

2) **API de Google Maps**

Para el uso de mapas en la aplicación web se ha optado por utilizar la API de Google MAPS [15] ya que es gratuita y ofrece un montón de ventajas como, por ejemplo, la búsqueda instantánea de lugares mediante un buscador insertado en el mapa.

4. Arquitectura de la aplicación

En este apartado se analiza cómo están construidas las aplicaciones a nivel arquitectónico (ver Figura 11), cómo se relacionan entre sí y el papel de cada uno de los elementos de la aplicación.

Se trata de dos aplicaciones independientes que se comunican mediante un almacén de datos común. Los elementos que forman el sistema son:

- Aplicación móvil
- Aplicación Web
- Servidor
- Base de datos

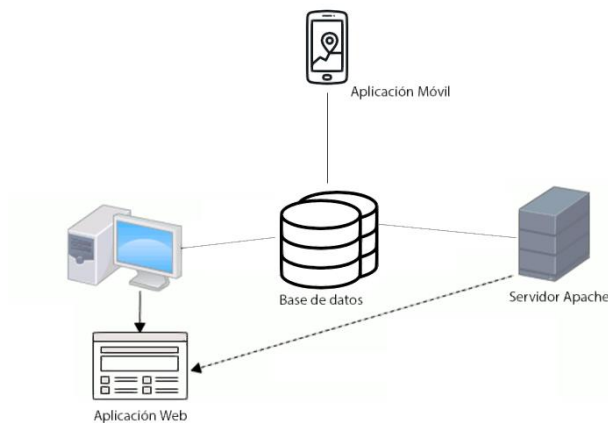


Figura 11 - Arquitectura

Mediante la aplicación web, el administrador rellenará la base de datos de información sobre los lugares y monumentos que desee. Incluyendo elementos multimedia como imágenes, documentos o audios que se alojarán en la base de datos.

La base de datos relacional es la encargada de almacenar los datos que el usuario administrador de la aplicación web inserta en ella. Se almacenarán los usuarios, lugares, comentarios, valoraciones, etc. Estos datos pueden ser modificados o eliminados en cualquier momento. Observar que la base de datos es común para las dos aplicaciones.

La aplicación móvil está orientada a dispositivos con el sistema operativo Android. La aplicación móvil es la que obtiene las coordenadas del usuario a través de la geolocalización y mediante estos datos es capaz de explotar la base de datos y mostrar los datos que en ella se encuentran.

Se utilizará un servidor para ejecutar la aplicación web y para acceder a la base de datos MySQL.

5. Modelo de datos

En esta sección se tratará la relación que existe entre los componentes de la base de datos y cómo esta se comunica con la aplicación web y móvil. Una vez definidas las tablas y las relaciones entre ellas se genera un gráfico visual de las relaciones existentes con los datos de cada tabla (ver Figura 12).

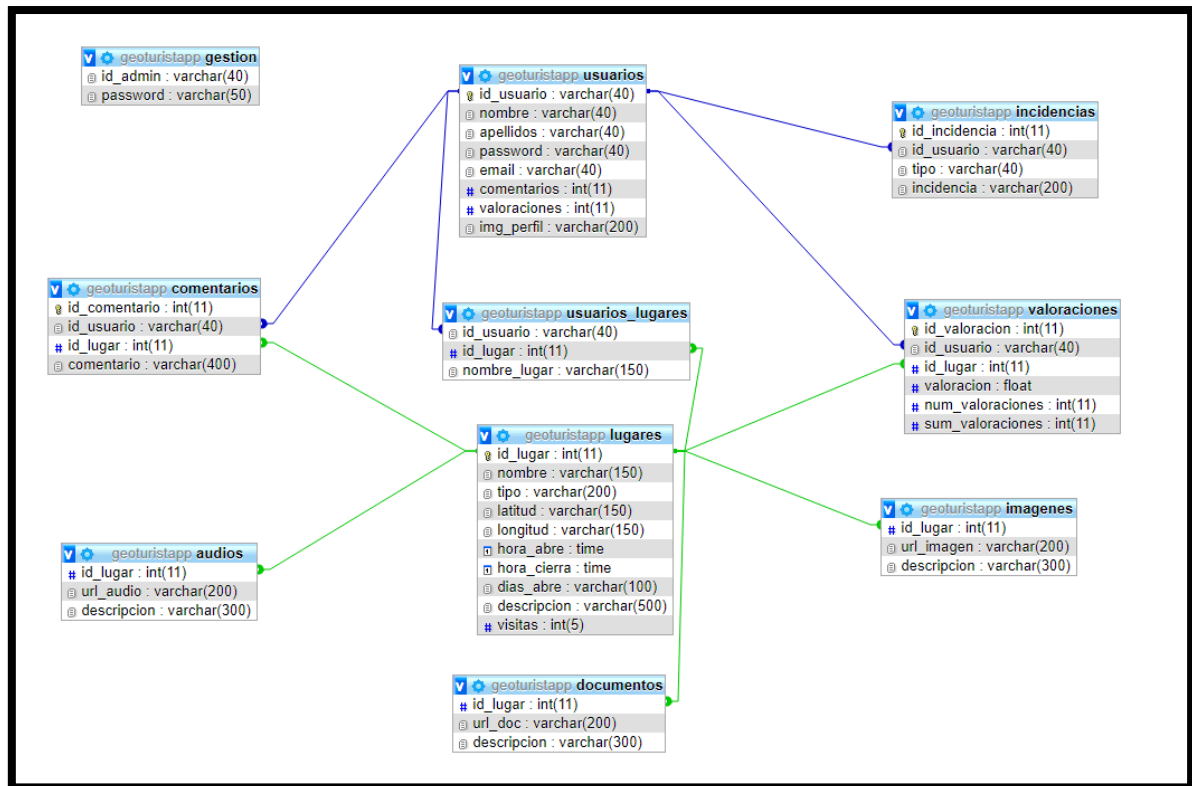


Figura 12 - Base de datos de la aplicación

Como se puede observar hay un total de 10 tablas que componen la estructura completa de la base de datos de la aplicación.

A continuación, se exponen los atributos de cada tabla:

Tabla usuarios:

#	Nombre	Tipo
1	id_usuario 	varchar(40)
2	nombre	varchar(40)
3	apellidos	varchar(40)
4	password	varchar(40)
5	email	varchar(40)
6	comentarios	int(11)
7	valoraciones	int(11)
8	img_perfil	varchar(200)

Figura 13 - Tabla de usuarios

En esta tabla (ver Figura 13), el **id_usuario** es la clave primaria. Equivale al nick de usuario cuándo se registra. Es única y es la que se utilizará para logarse en la aplicación y para realizar las consultas pertinentes a la base de datos, como, por ejemplo, obtener los comentarios o los monumentos que el usuario ha visitado.

- **nombre:** Cadena de caracteres que almacena el nombre del usuario.
- **apellidos:** Cadena de caracteres que almacena los apellidos del usuario.
- **password:** Cadena de caracteres que almacena la contraseña del usuario. Está cifrado con un algoritmo md5 para evitar que el usuario administrador pueda verla a simple vista.
- **email:** Cadena de caracteres que almacena el correo electrónico del usuario.
- **comentarios:** Entero que guarda el número de comentarios que ha realizado el usuario.
- **valoraciones:** Entero que guarda el número de valoraciones que ha realizado el usuario.
- **img_perfil:** Cadena de caracteres que almacena la ruta de imagen de perfil. (No se utiliza por ahora)

Tabla lugares (ver Figura 14):

#	Nombre	Tipo
1	id_lugar 	int(11)
2	nombre	varchar(150)
3	tipo	varchar(200)
4	latitud	varchar(150)
5	longitud	varchar(150)
6	hora_abre	time
7	hora_cierra	time
8	dias_abre	varchar(100)
9	descripcion	varchar(500)
10	visitas	int(5)

Figura 14 - Tabla de lugares

- **id_lugar:** Clave principal auto incrementable que identifica de forma única a un lugar. Se utiliza para hacer las peticiones pertinentes a la base de datos y obtener la información del lugar en cuestión. Como por ejemplo su descripción, la hora de apertura y cierre o los elementos multimedia que posee.
- **nombre:** Cadena de caracteres que almacena el nombre del monumento/lugar.
- **tipo:** Cadena de caracteres que almacena el tipo de lugar del que se trata. Una iglesia, un museo, un ayuntamiento...
- **latitud:** Cadena de caracteres que almacena la latitud de las coordenadas del monumento.
- **longitud:** Cadena de caracteres que almacena la longitud de las coordenadas del monumento.
- **hora_abre:** Dato de tipo time que almacena la hora de apertura del lugar.
- **hora_cierra:** Dato de tipo time que almacena la hora de cierre del lugar.
- **días_abre:** Cadena de caracteres que almacena los días que el monumento abre. Esta información se guarda como "L" "M" "X" ... para indicar los días de apertura.
- **descripción:** Cadena de caracteres que almacena la descripción del lugar.
- **visitas:** Entero que almacena el número de visitas que tiene el lugar. Se incrementa cada vez que se visita un monumento/lugar.

Tabla usuarios – lugares (ver Figura 15):

Nombre	Tipo
id_usuario 	varchar(40)
id_lugar 	int(11)
nombre_lugar	varchar(150)

Figura 15 - Tabla de usuarios - lugares

- **id_usuario:** Clave extranjera que hace referencia a la clave principal de la tabla usuarios.
- **id_lugar:** Clave extranjera que hace referencia a la clave principal de la tabla lugares.
- **nombre_lugar:** Cadena de caracteres que almacena el nombre del monumento.

Tabla valoraciones (ver Figura 16):

#	Nombre	Tipo
1	id_valoracion  	int(11)
2	id_usuario 	varchar(40)
3	id_lugar 	int(11)
4	valoracion	float
5	num_valoraciones	int(11)
6	sum_valoraciones	int(11)

Figura 16 - Tabla de valoraciones

- **id_valoracion:** Clave principal auto incrementable que identifica de forma única a una valoración realizada por un usuario de la aplicación
- **id_usuario:** Clave extranjera que hace referencia a la clave principal de la tabla usuarios.
- **id_lugar:** Clave extranjera que hace referencia a la clave principal de la tabla lugares.

- **valoración:** Dato de tipo float que almacena la valoración total del lugar/monumento. Se realiza una media entre todas las valoraciones y el número de usuarios.
- **num valoraciones:** Dato de tipo entero que almacena el número total de valoraciones que tiene el lugar/monumento. Necesarias para realizar la media.
- **sum valoraciones:** Dato de tipo entero que almacena la suma de todas las valoraciones que se han realizado al movimiento. Necesarias para realizar la media.

Tabla imágenes (ver Figura 17):

Nombre	Tipo
id_lugar 	int(11)
url_imagen	varchar(200)
descripcion	varchar(300)

Figura 17 - Tabla de imágenes

- **id lugar:** Clave extranjera que hace referencia al id_lugar de la tabla lugares.
- **url imagen:** Cadena de caracteres que almacena la url de la imagen del monumento.
- **descripción:** Descripción de la imagen. Por ahora no se utiliza.

Tabla documentos (ver Figura 18):

Nombre	Tipo
id_lugar 	int(11)
url_doc	varchar(200)
descripcion	varchar(300)

Figura 18 - Tabla de documentos

- **id lugar:** Clave extranjera que hace referencia al id_lugar de la tabla lugares.
- **url doc:** Cadena de caracteres que almacena la url del documento del monumento.
- **descripción:** Descripción del documento. Por ahora no se utiliza.

Tabla audios (ver Figura 19):

Nombre	Tipo
id_lugar 	int(11)
url_audio	varchar(200)
descripcion	varchar(300)

Figura 19 - Tabla de audios

- **id lugar:** Clave extranjera que hace referencia al id_lugar de la tabla lugares.
- **url audio:** Cadena de caracteres que almacena la url del audio del monumento.
- **descripción:** Descripción del audio. Por ahora no se utiliza.

Tabla comentarios (ver Figura 20):


Nombre	Tipo
id_comentario 	int(11)
id_usuario 	varchar(40)
id_lugar 	int(11)
comentario	varchar(400)

Figura 20 - Tabla de comentarios

- **id_comentario:** Clave principal auto incrementable que identifica de forma única un comentario realizado por un usuario de la aplicación.
- **id_usuario:** Clave extranjera que hace referencia al id_usuario de la tabla usuarios.
- **id_lugar:** Clave extranjera que hace referencia al id_lugar de la tabla lugares.
- **comentario:** Cadena de caracteres que almacena el comentario realizado por el usuario de un lugar/monumento.

Tabla gestión (ver Figura 21):

Nombre	Tipo
id_admin	varchar(40)
password	varchar(50)

Figura 21 - Tabla de gestión

Esta tabla corresponde al usuario que administra la página web. Son las credenciales necesarias para poder administrarla y no está relacionada con ninguna otra tabla de la aplicación para mayor seguridad.

- **id_admin:** Cadena de caracteres que almacena el id del usuario administrador de la página web.
- **password:** Cadena de caracteres que almacena la contraseña del administrador de la página web.

Tabla incidencias:



Nombre	Tipo
id_incidencia 	int(11)
id_usuario 	varchar(40)
tipo	varchar(40)
incidencia	varchar(200)

Figura 22 - Tabla de incidencias

Esta última tabla no se utiliza en la aplicación actualmente, ya que desde la aplicación móvil no se puede enviar ningún tipo de reporte por ahora. Pero se ha implementado para mejoras futuras del proyecto.

- **id incidencia:** Clave principal auto incrementable que identifica de forma única a una incidencia que un usuario ha realizado en la aplicación sobre un lugar/monumento.

- **id_usuario**: Clave extranjera que hace referencia al id_usuario de la tabla usuarios.
- **tipo**: Cadena de caracteres que almacena el tipo de incidencia, puede ser de tipo “Problema” o de “Información”
- **incidencia**: Cadena de caracteres que almacena la incidencia que reporta el usuario. Indicando el problema que ha encontrado o bien preguntando por algo en específico.

6. Diseño de la aplicación

En este apartado se van a mostrar los detalles sobre diseño que se han tenido en cuenta para realizar el sistema implementado. El diseño es una pieza clave en el desarrollo de una aplicación, ya que, si no se tiene una buena interfaz, no se podrá usar de una manera eficiente.

En este sentido, se ha tenido en cuenta la usabilidad, la seguridad y el tipo de usuario al que va dirigida la aplicación. Así mismo, se ha llevado a cabo un análisis de la interfaz, para adecuarla al usuario final y que este pueda usarla de la mejor manera posible.

6.1 Diseño de la Aplicación Web

Para la creación de la interfaz de la aplicación web se ha optado por un diseño:

- Sencillo
- Minimalista
- Intuitivo
- Simple
- Usable

La interfaz de la aplicación es muy sencilla por lo que cualquier usuario sin mucha experiencia podrá usarla sin problemas. El uso de tecnologías como HTML5 y CSS3 acompañadas de librerías como Bootstrap hacen que el diseño final de la aplicación web sea muy cómodo y usable.

Se ha optado por seguir un patrón genérico a la hora de desarrollar cada vista. De esta manera cuándo el usuario vaya a realizar una acción se dará cuenta de que el proceso es muy parecido para cada una de las funcionalidades disponibles.

A continuación, se va a describir cada una de las vistas que componen la aplicación web.

a) Login

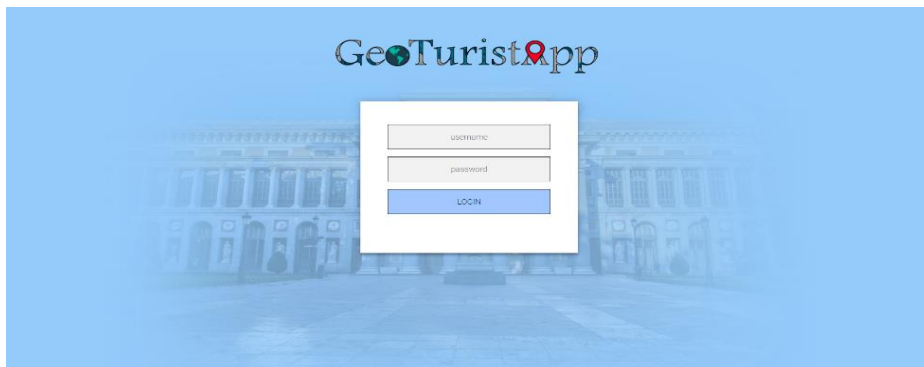


Figura 23 - Vista de login de la aplicación web

La vista de login es muy simple. Tiene dos campos para introducir el usuario y la contraseña del usuario que administra la web (ver Figura 23).

Estos datos validan directamente contra la base de datos, por lo que si los datos introducidos son incorrectos se informará al usuario del error.

b) Inicio

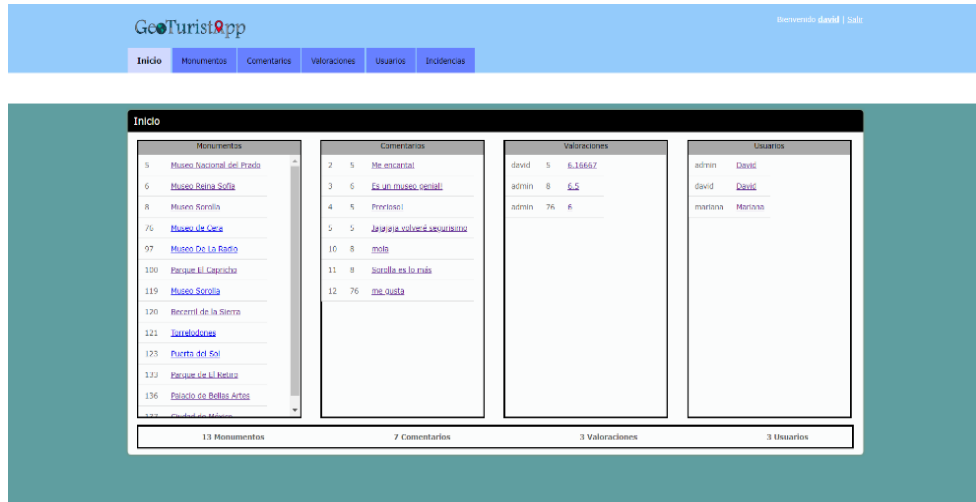


Figura 24 - Vista de inicio de la aplicación web

Una vez autenticado en la aplicación, el usuario verá la vista de Inicio. Esta vista contiene la información resumida que hay almacenada en la base de datos: monumentos, comentarios, valoraciones y usuarios. En la parte inferior hay un recuento de todos estos datos. Cada uno de estos datos es un enlace a la vista principal de cada tipo, es decir, los monumentos poseen enlaces que redirigen a la vista del monumento seleccionado para ver su información (ver Figura 24).

Además, mediante el uso de AJAX, esta información se actualiza cada 5 segundos para saber en tiempo real los datos que hay y sus cambios.

c) Monumentos

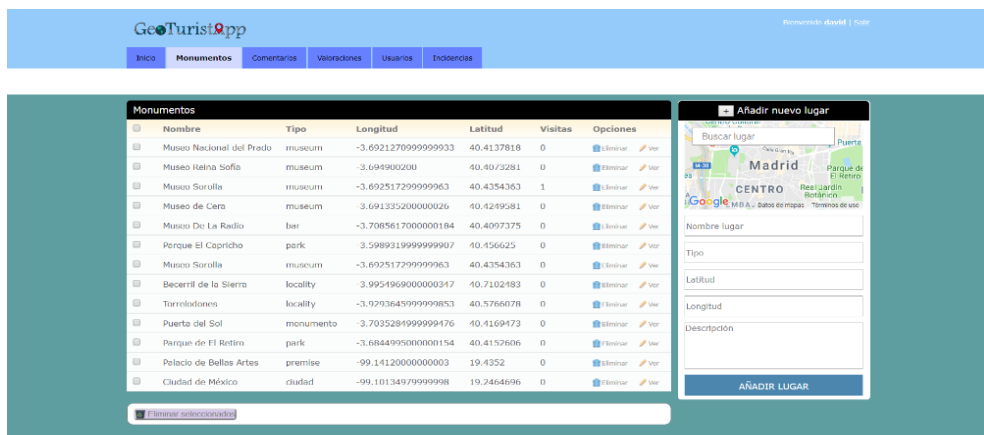


Figura 25 - Vista de monumentos de la aplicación web

En esta vista, el usuario administrador podrá ver la información de los lugares dados de alta: su nombre, el tipo, latitud, longitud y el número de visitas que lleva cada uno de ellos (ver Figura 25).

En la columna de opciones se pueden ver dos iconos. “Eliminar”, que eliminará de forma individual ese monumento y “Ver” que, al pulsar en él, redirigirá al usuario a la vista del monumento para ver todos sus datos.

Figura 26 - Añadir lugar

En la parte superior derecha se puede ver un formulario que contiene un mapa con un buscador y cinco campos de texto (ver Figura 26). Al escribir en el campo de búsqueda del mapa aparecen varias opciones que facilita la API de Google MAPS.

Al seleccionar la posición que el administrador desea, se incluirá la información del lugar en los campos de texto de la zona inferior. El nombre, el tipo y las coordenadas (latitud y longitud) se incluyen de forma automática. Y el último campo “Descripción” se podrá rellenar si el usuario administrador quiere indicar más detalles sobre el lugar.

Todos los campos se pueden modificar por si el usuario administrador no está de acuerdo con alguno de los datos generados automáticamente.

Al pulsar en el botón de añadir lugar, se guardará en la base de datos.

d) Añadir lugar

Figura 27 - Añadir lugar completo

Si el usuario administrador quiere dar más detalles sobre el lugar que va a agregar puede pulsar sobre el botón de “+” del formulario de la vista anterior. Al pulsar en él, se le redirigirá a la vista de “Añadir Lugar”. Esta vista aporta más campos que la anterior, añadiendo la hora de apertura y cierre del lugar, los días que abre y los elementos multimedia (ver Figura 27). El administrador podrá añadir imágenes del lugar, audios, como por ejemplo audio guías del lugar, y documentos, como planos, pdfs, excels... más elaborados.

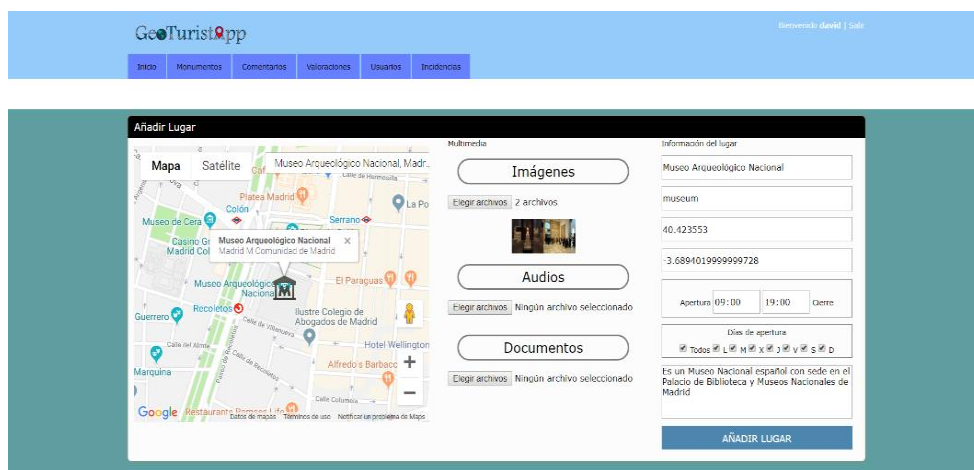


Figura 28 - Añadir monumento con campos completos

La vista con los campos completos se visualizará como se puede observar en la imagen. Si el usuario administrador pulsa en el botón de Añadir lugar, se guardará toda la información del monumento en la base de datos (ver Figura 28).

e) Actualizar elementos multimedia

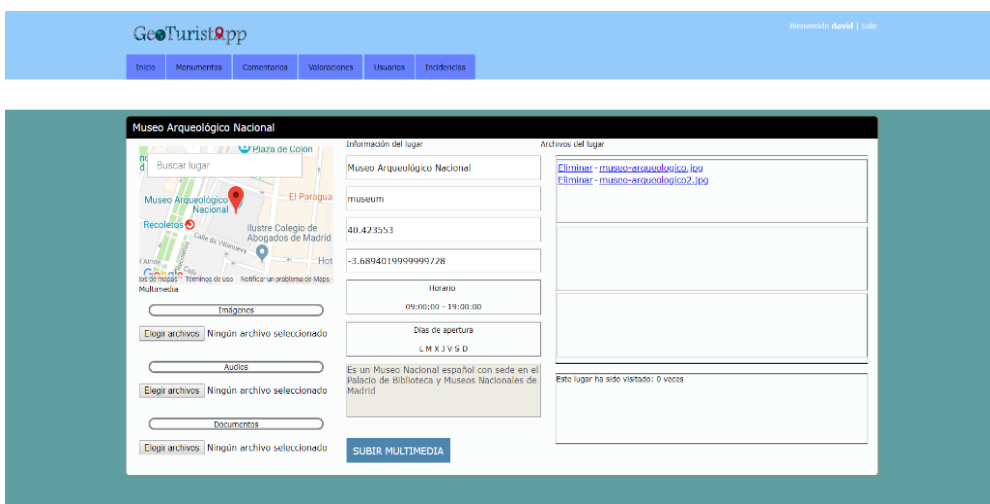


Figura 29 - Ver monumento y actualizar elementos multimedia

Si el usuario lo desea, podrá acceder a la información del monumento pulsando en el botón de ver en la lista de todos los monumentos. Los campos son editables por si el usuario desea copiar información que le resulte útil.

Se puede ver en la parte derecha los archivos multimedia que incluye el lugar, pudiendo además de verlos, eliminarlos. En la parte inferior derecha también se puede ver el número de visitas que tiene el lugar (ver Figura 29).

Además de poder ver su información, el administrador podrá agregar nuevos elementos multimedia. De esta manera no es necesario crear un nuevo lugar con más archivos multimedia, si no que se agregan al lugar actual.

f) Eliminar monumentos

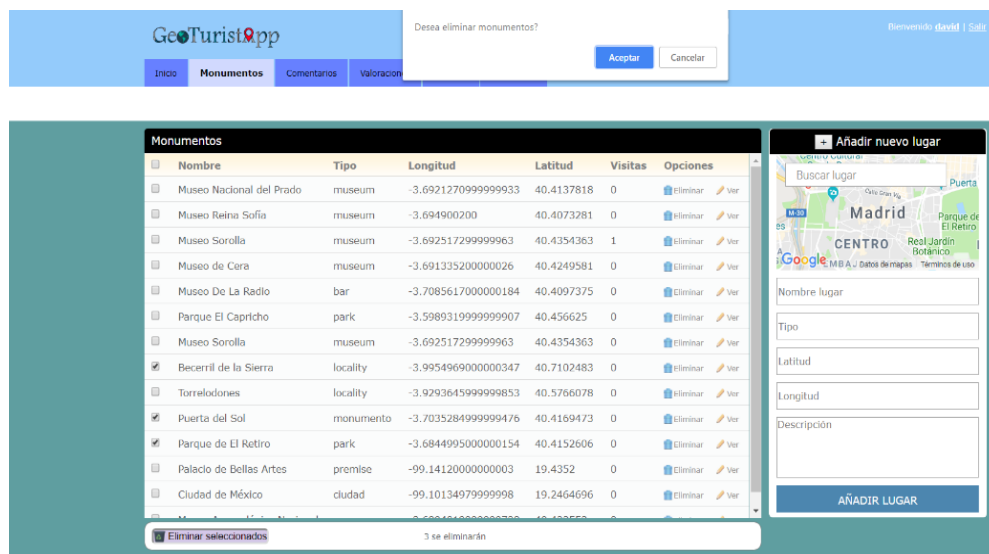


Figura 30 - Eliminar monumento/s

En la vista de monumentos (ver Figura 30) el usuario administrador podrá eliminar los lugares de uno en uno o si lo prefiere seleccionar varios y eliminarlos de una sola vez. Para mayor seguridad, cada vez que se decida eliminar un lugar, se notificará con una alerta para asegurarse de que eso es lo que se quiere hacer.

g) Comentarios

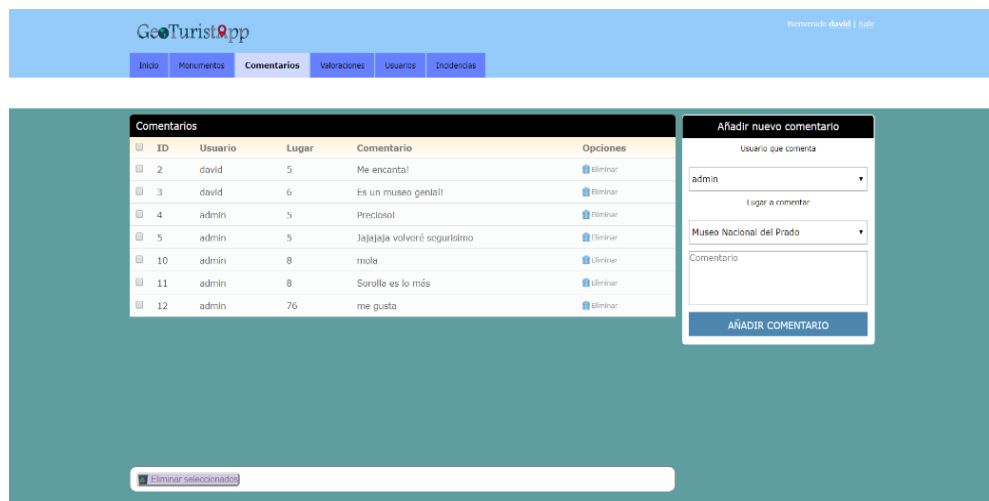


Figura 31 - Comentarios de los monumentos

La vista de comentarios muestra los usuarios que han realizado comentarios sobre lugares desde la app. En esta vista se muestra la información básica sobre los comentarios, como el usuario que lo ha realizado, el id del lugar y el propio comentario. El administrador además podrá eliminar los que considere inapropiados (ver Figura 31).

h) Valoraciones

The screenshot shows the 'Valoraciones' page with a navigation bar at the top containing 'Inicio', 'Monumentos', 'Comentarios', 'Valoraciones', 'Usuarios', and 'Incidencias'. The main content area is divided into two sections:

- Valoraciones:** A table listing reviews by user, location, rating, and number of reviews.

Usuario	Lugar	Valoración	Num.Val	Opciones
david	5	6.16667	6	Eliminar
admin	8	6.5	2	Eliminar
admin	76	6	1	Eliminar
- Valoraciones actuales:** A form to add a new review. It includes a dropdown for 'Usuario que valora' (currently 'david'), a dropdown for 'Lugar a valorar' (currently 'Museo Arqueológico Nacional'), a 'Valoración' input field, and an 'AÑADIR VALORACIÓN' button.

At the bottom of the page, there is a search bar with the text 'Eliminar seleccionados'.

Figura 32 - Valoraciones de los monumentos

La vista de valoraciones (ver Figura 32) guarda relación con la vista de Comentarios. En ella se pueden ver las valoraciones realizadas por los usuarios de la app. El id del lugar que se valora, la valoración total del lugar y el número total de ellas.

Se pueden incorporar nuevas valoraciones. Las valoraciones también se pueden eliminar y sólo es posible valorar del 1 al 10.

i) Usuarios

The screenshot shows the 'Usuarios' page with a navigation bar at the top containing 'Inicio', 'Monumentos', 'Comentarios', 'Valoraciones', 'Usuarios', and 'Incidencias'. The main content area is divided into two sections:

- Usuarios:** A table listing users with their ID, name, surnames, email, and counts of comments and reviews.

ID	Nombre	Apellidos	E-Mail	C	V	Opciones
admin	David	Admin	david_admin@ucm.es	9	5	Eliminar
david	David	Rodriguez	deviro05@ucm.es	2	0	Eliminar
mariana	Mariana	Vega	mariana@gmail.com	0	0	Eliminar
- Añadir nuevo usuario:** A form to add a new user. It includes input fields for 'ID Usuario', 'Password', 'Nombre', 'Apellidos', and 'E-Mail', and an 'AÑADIR USUARIO' button.

At the bottom of the page, there is a search bar with the text 'Eliminar seleccionados'.

Figura 33 - Usuarios de la aplicación

La vista de usuarios (ver Figura 33) muestra al administrador de la aplicación cuántos usuarios hay registrados en la app. Muestra toda su información personal además de los comentarios y valoraciones que han realizado.

Podrán eliminarse los usuarios si por ejemplo uno de ellos no tiene un buen comportamiento por los comentarios que realice.

j) Incidencias

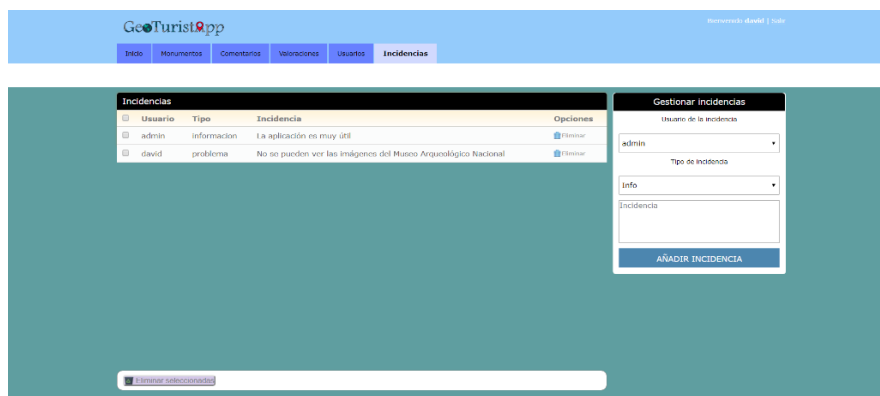


Figura 34 - Incidencias de la aplicación

La vista de incidencias (ver Figura 34) muestra los problemas reportados por los usuarios de la app. Si encuentran algún problema se verán en esta vista.

Se muestra el usuario que ha reportado el error o ha solicitado información, el tipo de reporte y el texto de la incidencia.

Como en todas las vistas se pueden agregar incidencias y también eliminar las existentes.

6.2 Diseño de la Aplicación Móvil

La aplicación móvil posee un diseño:

- Intuitivo
- Usable
- Simple
- Navegable

A lo largo del desarrollo se ha pensado en los tipos de usuario que iban a usar la aplicación y está pensada para que cualquier usuario sea de la edad que sea pueda utilizarla. Desde el usuario experto que está acostumbrado al uso de apps, al usuario inexperto que no tiene mucha relación con la tecnología.

a) Login



Figura 35 - Vista de Login de la app

La vista de login (ver Figura 35) es muy sencilla, tiene dos campos de texto, uno para el username y el password, que sirven para que el usuario registrado valide contra la base de datos. En el caso de que el usuario no tenga una cuenta, podrá pinchar en el enlace de “Registrarse” que le redirigirá a la vista de Registro que se describirá a continuación.

Se ha optado por una interfaz sencilla, eligiendo una gama de colores que sea cómoda para el usuario y distintiva.

b) Registro

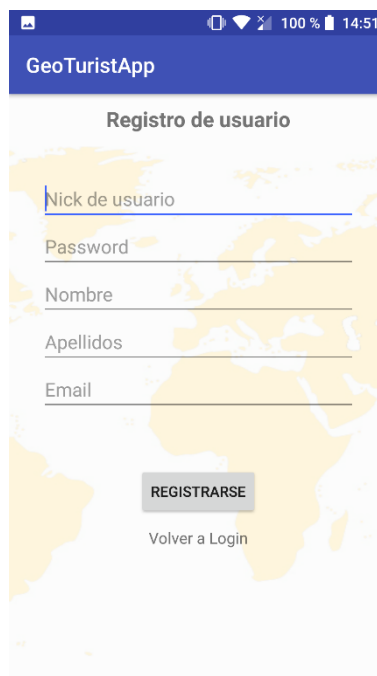


Figura 36 - Vista de Registro de la app

En la vista de Registro (ver Figura 36) tenemos los campos principales que corresponden a los datos personales del usuario. El nick de usuario que será único, el nombre y los apellidos y el email.

Se ha decidido que todos los campos sean obligatorios

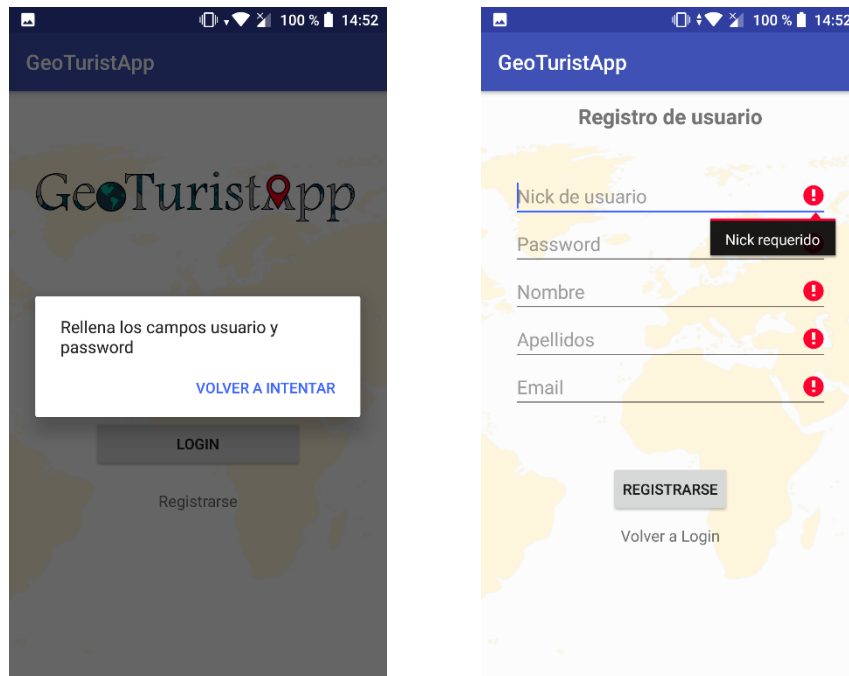


Figura 37 - Validación de los formularios de Login y Registro

La validación está presente en ambas vistas (ver Figura 37). Si no se rellenan los campos la interfaz mostrará los mensajes de error correspondientes. Además, en el registro se indicará si el nick ya existe o si algún campo del formulario no es correcto.

Una vez realizado el registro se redirigirá al usuario a la pantalla principal de login. En ella pondrá sus datos y entrará en la página principal de la aplicación.

c) Pantalla principal



Figura 38 - Pantalla Principal de la app

En la vista principal (ver Figura 38) de la aplicación tendremos tres posibles opciones. La primera de ellas “Mis monumentos” mostrará una vista nueva en la que se incluyen los monumentos que ha visitado el usuario actual, es decir, los monumentos que ha añadido a su lista una vez que los ha podido localizar.

El segundo botón “Descubre monumentos” llevará al usuario a una vista para geolocalizar su posición y comprobar si existen lugares o monumentos cerca de él. Si no se encuentra ningún monumento o lugar en un radio de máximo 1 km no se mostrará nada en la lista.

El último botón “Mis comentarios” llevará al usuario a una vista dónde se encuentran todos los comentarios que ha realizado. Se realizará un recuento de ellos y se mostrarán en forma de lista añadiendo un scroll para mayor comodidad.

Además de estos tres botones la interfaz contará siempre con un menú lateral accesible desde cualquier vista de la app.

d) Menú lateral

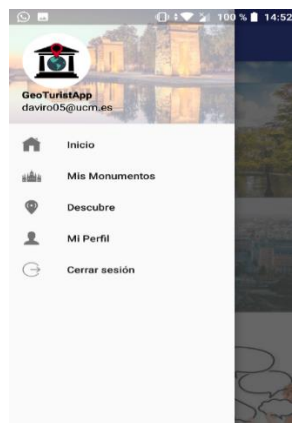


Figura 39 - Menú lateral de la app

Al desplegar el menú aparecerán las distintas opciones disponibles en la aplicación (ver Figura 39). Además de incluir las de la vista principal, también incluirá la opción de “Mi Perfil”.

e) Mis monumentos



Figura 40 - Vista de Mis Monumentos en la app

Como se ha comentado anteriormente, la vista de “Mis monumentos” muestra los monumentos que han sido visitados por el usuario. Sólo aparecerán los lugares o monumentos que el usuario haya añadido a su lista (ver Figura 40).

Cada uno de los monumentos que se muestran en la lista es un enlace a la vista de ese monumento en específico, que mostrará la información completa de este.

f) Ubicación y Descubre Monumentos

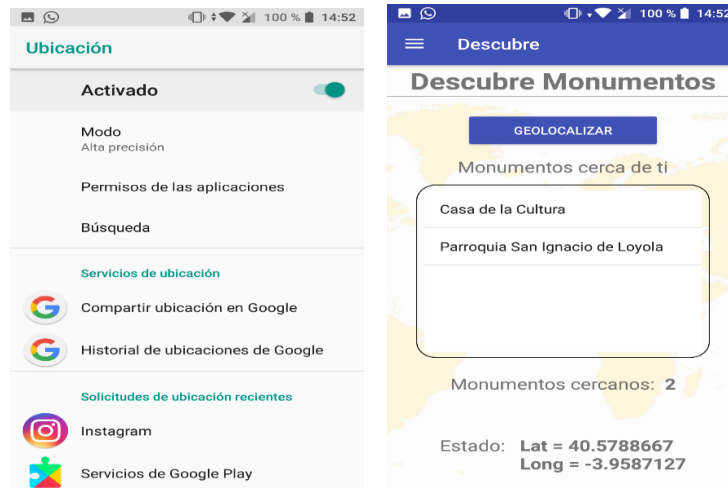


Figura 41 - Vista de Uso de Ubicación y Descubre Monumentos en la app

El tercer elemento del menú es el de “Descubre monumentos” que da la posibilidad al usuario de obtener su posición geográfica mediante un botón “Geolocalizar” y calcular la latitud y longitud del lugar en el que se encuentra para que pueda comprobar si hay lugares o monumentos cercanos a él.

Al pulsar sobre el botón de “Geolocalizar” se pedirá al usuario que active su ubicación (en el caso de que no la tenga activada”). (ver Figura 41)

Al activarse el gps del dispositivo móvil obtendrá la ubicación y comprobará si alguno de los lugares guardados en la base de datos está dentro de la latitud y longitud correspondientes.

g) Vista Monumento



Figura 42 - Vista de un monumento en la app

Al pulsar sobre un lugar en la lista de monumentos se redirigirá al usuario a la vista del monumento seleccionado. En ella aparecerá la información básica. Una descripción (si posee una), los días de apertura, las vistas totales y la valoración acumulada que lleva.

En la parte inferior se encuentran los comentarios que se han realizado de ese lugar mediante una lista con scroll para facilitar su visualización.

Además, se incluyen tres botones. El primero de ellos se encuentra en la parte superior y es el de “Añadir monumento” que solamente estará activo si el usuario no lo ha añadido a su lista de lugares. Al pulsar en el botón, se agregará el lugar a la lista y se deshabilitará mostrando el mensaje de “Añadido”.

Los otros dos botones son “Multimedia” y “Valorar”. El primero de ellos llevará al usuario a la vista de elementos multimedia de ese monumento. El otro, redirigirá a la vista de valorar el lugar o monumento y añadir un comentario si lo desea el usuario.

h) Vista Valorar

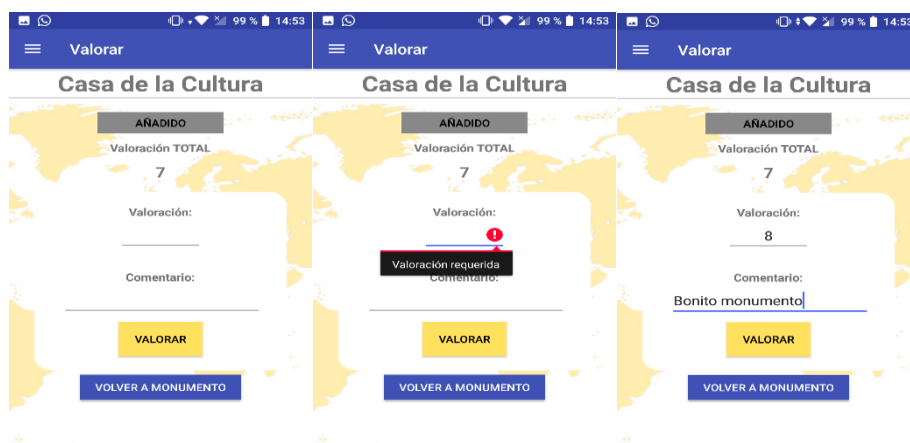


Figura 43 - Vistas de Valorar y Comentar un monumento

Las tres vistas anteriores muestran la forma que tiene el usuario de valorar un lugar (ver Figura 43). La valoración será siempre obligatoria y si no se introduce una al darle al botón de valorar, no se permitirá realizar la valoración. En cambio, los comentarios son opcionales.

i) Vista de elementos multimedia



Figura 44 - Vista de elementos multimedia de un monumento

La vista de elementos multimedia (ver Figura 44) mostrará al usuario los archivos de imagen, audio y documentos que posee el lugar seleccionado.



Figura 45 - Vista de Mi Perfil de la app

La vista de perfil (ver Figura 45) muestra los datos personales del usuario además de incluir los comentarios que ha realizado y las valoraciones que ha hecho a cada lugar.

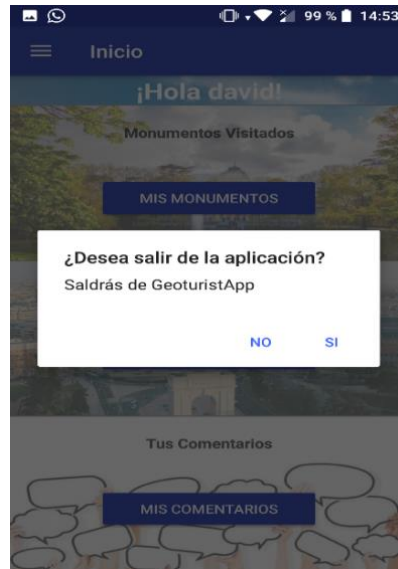


Figura 46 - Vista de Cerrar sesión de la app

Para cerrar sesión (ver Figura 46) en la app y que no se recuerde el usuario autenticado, se podrá hacer desde el menú lateral, en la opción de "Cerrar sesión".

7. Implementación de la aplicación

En este capítulo se muestra y detalla el proceso de implementación de la aplicación web y de la aplicación móvil. Además de mostrar cómo se construye la base de datos y los algoritmos más importantes de la aplicación.

7.1 Implementación de la Aplicación Web

La parte web de la aplicación posee varias funcionalidades. A continuación, se van a explicar las más relevantes.

a) Login:

```

if($_POST['usuario']!=" ")
{
    $usuario=$_POST['usuario'];

    if (!isset($_SESSION['usuario'])) {
        $_SESSION['usuario']=$usuario;
    }
    $clave=$_POST['password'];

    if(validar_clave($clave))
    {
        $usuario=mysqli_real_escape_string($conexion,$usuario);
        $clave=md5(mysqli_real_escape_string($conexion,$clave));

        $sentencia=mysqli_query($conexion,"SELECT * FROM gestion WHERE id_admin ='$usuario' AND password ='$clave'");

        if(mysqli_num_rows($sentencia)==1)
        {
            $fila=mysqli_fetch_array($sentencia);
            $valor=$fila['id_admin'];
            $_SESSION['usuario']=$valor;

            header("Location:inicio.php");
        }
        else
        {
            header("Location: index.php?error=error_nickopass");
        }
    }
    mysqli_close($conexion);
}

```

Figura 47 - Código de Login

El proceso de autenticarse es sencillo (ver Figura 47). El código utiliza el usuario y password del formulario principal para validar esa información con la tabla **gestión** de la base de datos.

b) Inicio:

Tras el login, se muestra la página principal de la aplicación web, que contiene los datos principales de la base de datos. Los monumentos, valoraciones, comentarios y usuarios.

Cada uno de ellos está separado en un fichero individual, para mantener un encapsulamiento y hacerlo genérico (ver Figura 48).

```

<?php
if(isset($_GET['id'])){
    if($_GET['id'] == 'monumentos')
        include 'monumentos.php';
    if($_GET['id'] == 'comentarios')
        include 'comentarios.php';
    if($_GET['id'] == 'valoraciones')
        include 'valoraciones.php';
    if($_GET['id'] == 'usuarios')
        include 'usuarios.php';
    if($_GET['id'] == 'incidencias')
        include 'incidencias.php';
    if($_GET['id'] == 'add_monumento')
        include 'add_monumento.php';
    if($_GET['id'] == 'ver_monumento')
        include 'ver_monumento.php';
    if($_GET['id'] == 'configuracion')
        include 'configuracion.php';
}
else{
    include 'principal.php';
}
}
?>

```

Figura 48 - Código de Inicio secciones

Dependiendo de la página a la que el usuario navegue, se mostrará una vista u otra. Según el parámetro id de la url.

La vista principal tiene una característica especial, y es que hace uso de AJAX para que los datos que se muestran estén actualizados en tiempo real (ver Figura 49).

```

$(document).ready(function() {
    // Llamamos a las funciones la primera vez para que no tarde en cargar
    actualizar_lugares();
    actualizar_usuarios();
    actualizar_comentarios();
    actualizar_valoraciones();

    function actualizar_lugares(){
        var tipo="lugares";

        $.ajax({
            type: 'POST',
            url: "actualizar.php?tipo="+tipo,
            success: function(respuesta) {
                $('#up_lugares').html(respuesta);
            }
        });
    }

    function actualizar_usuarios(){
        var tipo="usuarios";

        $.ajax({
            type: 'POST',
            url: "actualizar.php?tipo="+tipo,
            beforeSend: function () {
                $('#up_usuarios').html("Procesando, espere por favor...");
            },
            success: function(respuesta) {
                $('#up_usuarios').html(respuesta);
            }
        });
    }
}

```

Figura 49 - Código de actualizar datos

c) Obtener monumentos:

En la vista de monumentos se pueden ver todos los lugares que hay dados de alta en la base de datos.

```

function obtener_lugares($conexion){
    $lugares = mysqli_query($conexion,"SELECT * FROM lugares");

    if($lugares){
        while($fila = mysqli_fetch_array($lugares))
        {
            echo "<tr>
                <td><input type='checkbox' name='lug_sel[]' class='lugares_sel' value='{$fila[id_lugar]}'/></td>
                <td>{$fila[nombre]}</td>
                <td>{$fila[tipo]}</td>
                <td class='long'>{$fila[longitud]}</td>
                <td class='lat'>{$fila[latitud]}</td>
                <td>{$fila[visitas]}</td>
                <td><a href='inicio.php?id=monumentos&eliminar={$fila[id_lugar]}' class='ico del' onclick='return confirmar()'>Eliminar</a>
                <a href='inicio.php?id=ver_monumento&id_lugar={$fila[id_lugar]}'
                &nombre_lugar={$fila[nombre]}&lat={$fila[latitud]}&lon={$fila[longitud]}'
                &h_abre={$fila[hora_abre]}&h_cierra={$fila[hora_cierra]}&dias_abre={$fila[dias_abre]}
                &descripcion={$fila[descripcion]}&visitas={$fila[visitas]}' class='ico edit'>Ver</a></td>
            </tr>";
        }
    }
}

```

Figura 50 - Código de obtener monumentos

La función implementada obtener_lugares recoge todos los lugares guardados y los muestra directamente a la vista (ver Figura 50).

En esta vista también se permite eliminar un lugar (ver Figura 51).

d) Eliminar lugares:

```

function eliminar_lugar($id_lugar,$conexion){
    $eliminar = mysqli_query($conexion,"DELETE FROM lugares WHERE id_lugar='{$id_lugar}'");
    echo "<script>>window.location = './inicio.php?id=monumentos'</script>";
}

```

Figura 51 - Código de eliminar lugares

e) Ver monumento:

El usuario puede elegir ver un monumento en específico. De esta manera podrá ver todos los datos del lugar además de los elementos multimedia (si posee alguno de ellos) (ver Figura 52).

```

<?php
    $ver_lugares = mysqli_query($conexion,"SELECT * FROM lugares
    WHERE id_lugar='{$_GET[id_lugar]}");

    if($ver_lugares){
        if($file=mysqli_fetch_array($ver_lugares))
        {
            $nombre_lugar=$file['nombre'];
            $tipo=$file['tipo'];
            $lat=$file['latitud'];
            $lon=$file['longitud'];
            $h_abre=$file['hora_abre'];
            $h_cierra=$file['hora_cierra'];
            $dias_abre=$file['dias_abre'];
            $descripcion=$file['descripcion'];
            $visitas=$file['visitas'];
        }
    }
}

```

Figura 52 - Código de ver monumento

Y los elementos multimedia se visualizan gracias a la función obtener_archivos que según el parámetro que se le pase mostrará un tipo de archivo u otro (imágenes, audios o documentos)

```
function obtener_archivos($id_lugar,$tipo,$conexion){
    $url="";

    if($tipo == "img_lugares"){
        $sql = "SELECT * FROM imagenes WHERE id_lugar='$id_lugar'";
        $url="url_imagen";
    }
    else if($tipo == "audio_lugares"){
        $sql = "SELECT * FROM audios WHERE id_lugar='$id_lugar'";
        $url="url_audio";
    }
    else if($tipo == "doc_lugares"){
        $sql = "SELECT * FROM documentos WHERE id_lugar='$id_lugar'";
        $url="url_doc";
    }

    $resultado = mysqli_query($conexion,$sql);

    if($resultado){
        while($fila = mysqli_fetch_array($resultado))
        {
            echo "<tr>
                <td><a href='inicio.php?id=ver_monumento&id_lugar=$fila[id_lugar]
                &nombre_lugar=$_GET[nombre_lugar]&lat=$_GET[lat]&lon=$_GET[lon]
                &tipo_archivo=$tipo&url=$fila[url]&eliminar=true'>Eliminar</a></td>
                <td> - </td>
                <td><a href='./multimedia/$tipo/$fila[url]'>$fila[url]</a></td>
            </tr>";
        }
    }
}
```

Figura 53 - Código de obtener multimedia

Para mostrar los usuarios, comentarios y valoraciones la implementación que se utiliza es genérica. Sólo cambia la tabla de la base de datos a la que se hace la consulta (ver Figura 53).

Estas funciones son:

- **obtener_usuarios, obtener_comentarios y obtener_valoraciones**

```
function obtener_valoraciones($conexion){
    $valoraciones = mysqli_query($conexion,"SELECT * FROM valoraciones");

    if($valoraciones){
        while($fila = mysqli_fetch_array($valoraciones))
        {
            echo "<tr>
                <td><input type='checkbox' name='valor_sel[]' class='lugares_sel' value='$fila[id_valoracion]'></td>
                <td>$fila[id_usuario]</td>
                <td>$fila[id_lugar]</td>
                <td>$fila[valoracion]</td>
                <td>$fila[num_valoraciones]</td>
                <td><a href='inicio.php?id=valoraciones&eliminar=$fila[id_valoracion]' class='ico del' onclick='return confirmar()'>Eliminar</a>
            </tr>";
        }
    }
}
```

Figura 54 - Código de obtener valoraciones

Además, cada una de ellas también posee la función de eliminar un elemento de la tabla (ver Figura 54).

La función más interesante en la parte de la aplicación web es la de dar de alta lugares, ya que serán necesarios para que la aplicación móvil pueda encontrarlos a la hora de realizar una consulta con las coordenadas de su posición.

- **Alta lugar:**

```
function alta_lugar_completo($nombre,$tipo,$latitud,$longitud,$hora_abre,$hora_cierra,$dias_abre,$desc,$visitas,$conexion)
{
    $alta = mysqli_query($conexion,"INSERT INTO lugares (nombre,tipo,latitud,longitud,hora_abre,hora_cierra,
    dias_abre,descripcion,visitas)
    VALUES ('$nombre','$tipo','$latitud','$longitud','$hora_abre','$hora_cierra','$dias_abre','$desc','$visitas')");

    if($alta){
        $query=mysqli_query($conexion,"SELECT MAX(id_lugar) as id FROM lugares");
        $id=mysqli_fetch_array($query); //$id["id"] = el ultimo id (mayor)
        //echo "id: ".$id['id'];
        //echo($FILES['file_img']['size'][0]);

        if($FILES['file_img']['size'][0] != 0){
            subir_imagenes($id['id'],$conexion);
        }
        if($FILES['file_audio']['size'][0] != 0){
            subir_audios($id['id'],$conexion);
        }
        if($FILES['file_doc']['size'][0] != 0){
            subir_docs($id['id'],$conexion);
        }
    }
    else
        echo "<p>Error</p>";

    echo "<script>>window.location = './inicio.php?id=monumentos'</script>";
}
}
```

Figura 55 - Código de alta de lugar

Tras rellenar el formulario de alta, se pasa toda la información a la función que se muestra en la imagen anterior: **alta_lugar_completo**

Esta función genera un id único para el nuevo lugar que se da de alta en la base de datos (ver Figura 55). Además, si incluye elementos multimedia, se añaden llamando a cada una de las funciones anteriormente mencionadas: **subir_imagenes, subir_audios, subir_docs.**

7.2 Implementación de la Aplicación Móvil

La aplicación móvil se compone de tres partes (ver Figura 56). Cada una de ellas se comunica con la otra y crean una conexión que hace que la aplicación funcione.

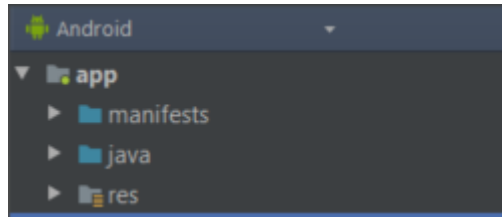


Figura 56 - Estructura app

La primera de ellas es manifest que incluye el fichero de configuración de la aplicación llamado AndroidManifest.xml. A continuación, se pueden ver las actividades de las que se componen y los permisos necesarios (ver Figura 57) para el buen funcionamiento de la aplicación.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Figura 57 - Permisos app

Los permisos son:

- Permiso de acceso a internet
- Permisos para acceder a la localización del dispositivo móvil

El siguiente componente de la aplicación Android es el paquete java. En él se encuentra el código de nuestra aplicación, las clases, paquetes y componentes necesarios para construir la app.

La siguiente imagen muestra la estructura de la carpeta java (ver Figura 58):

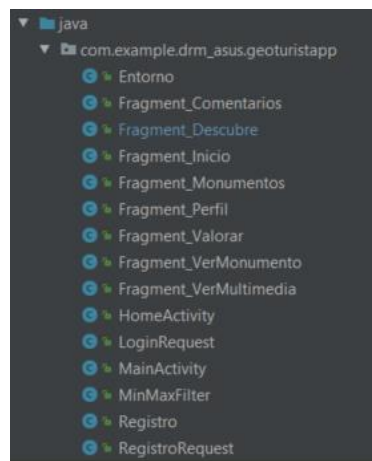


Figura 58 - Estructura clases java

La última parte, es la carpeta `res`, que incluye las vistas, los elementos visuales, como por ejemplo imágenes que utiliza la aplicación y se guardan en la carpeta `drawable` y `mipmap`. Las vistas son ficheros `xml` que se guardan en la carpeta `layout` (ver Figura 59).

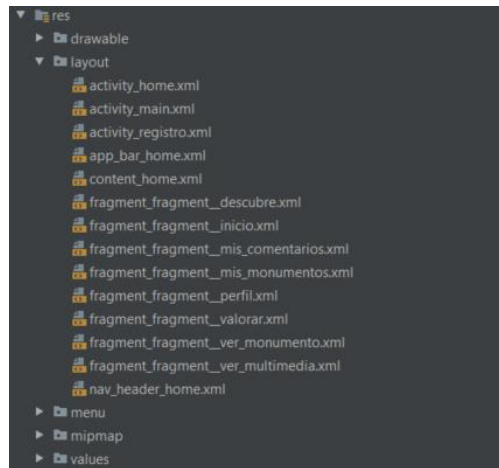


Figura 59 - Estructura `res`

Después de definir la estructura básica de nuestra aplicación móvil se va a explicar cada una de las funcionalidades implementadas en ella.

a) Login:

Para el login se hace uso del objeto **SharedPreferences** (ver Figura 60) que sirve para almacenar el usuario y la contraseña y mantener la aplicación con la sesión iniciada.

```
SharedPreferences.Editor editor = getSharedPreferences(USER_PASS, MODE_PRIVATE).edit();
editor.putString(R.id.username, jsonResponse.getString("id_usuario"));
editor.putString(R.id.password, et_password.getText().toString());
editor.apply();
```

Figura 60 - `SharedPreferences`

Además, el login se hace mediante Volley, una librería de Android para poder hacer peticiones `http`. Se hace una petición a la función `php login_usuario` pasándole los datos en un `HashMap` y obteniendo una respuesta. En caso de ser correcta, se entrará en la aplicación (ver Figura 61).

```
private Map<String, String> params;
public LoginRequest(String id_usuario, String password, Response.Listener<String> listener) {
    super(Request.Method.POST, LOGIN_REQUEST_URL, listener, errorListener: null);
    params = new HashMap<>();
    params.put(R.id.id_usuario, id_usuario);
    params.put(R.id.password, password);
}
```

Figura 61 - Realizar login

Si hubiera algún error se mostraría un mensaje de usuario o `password` incorrectos.

b) Registro:

El registro (ver Figura 62) está implementado igual que el login. Los datos se pasan mediante Volley, realizando una petición y llamando a la función del servidor registro_usuario

```
private Map<String,String> params;

public RegistroRequest(String id_usuario, String nombre, String apellidos, String password, String email, Response.Listener<String> listener){
    super(Method.POST, REGISTER_REQUEST_URL,listener, errorListener: null);
    params= new HashMap<>();
    params.put( R.id.usuario, id_usuario);
    params.put( R.id.nombre, nombre);
    params.put( R.id.apellidos, apellidos);
    params.put( R.id.password, password);
    params.put( R.id.email, email);
}
}
```

Figura 62 - Realizar registro

Si el registro es correcto, cambiará la vista a la de login para que el usuario pueda introducir los datos con los que se acaba de registrar.

Si hubiera algún error en el registro se mostrará en la vista indicando de que error se trata.

c) Inicio:

Una vez realizado el login correctamente se redirigirá al usuario a la pantalla principal de la aplicación móvil (ver Figura 63). En ella podrá acceder tanto al menú lateral y cada uno de sus componentes, implementado con la función onOptionsItemSelected

```
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle navigation view item clicks here.
    int id = item.getItemId();

    if (id == R.id.nav_inicio) {
        // Creamos un bundle para poder pasarle los datos del usuario logeado y poder pasarselos a todos los fragments
        Fragment_inicio fragment_inicio = new Fragment_inicio();
        fragmentManager.beginTransaction().replace(R.id.contenedor, fragment_inicio).commit();
    } else if (id == R.id.nav_monumentos) {
        Fragment_Monumentos fragment_monumentos = new Fragment_Monumentos();
        fragmentManager.beginTransaction().replace(R.id.contenedor, fragment_monumentos).commit();
    } else if (id == R.id.nav_descubre) {
        Fragment_Descubre fragment_descubre = new Fragment_Descubre();
        fragmentManager.beginTransaction().replace(R.id.contenedor, fragment_descubre).commit();
    }
}
```

Figura 63 - Menú app

También existen tres botones que redirigen a Mis monumentos, Descubre y Mis comentarios (ver Figura 64).

```

bienvenida.setText("Hola "+username+"");

btn_mis_monumentos.setOnClickListener((view) -> {
    Fragment_Monumentos fragment_monumentos = new Fragment_Monumentos();

    getActivity().getSupportFragmentManager().beginTransaction()
        .replace(R.id.contenedor, fragment_monumentos)
        .addToBackStack(null)
        .commit();
});

btn_descubre_monumentos.setOnClickListener((view) -> {
    Fragment_Descubre fragment_descubre = new Fragment_Descubre();

    getActivity().getSupportFragmentManager().beginTransaction()
        .replace(R.id.contenedor, fragment_descubre)
        .addToBackStack(null)
        .commit();
});

btn_mis_comentarios.setOnClickListener((view) -> {
    Fragment_Comentarios fragment_comentarios = new Fragment_Comentarios();

    getActivity().getSupportFragmentManager().beginTransaction()
        .replace(R.id.contenedor, fragment_comentarios)
        .addToBackStack(null)
        .commit();
});

```

Figura 64 - Botones inicio

d) Mis monumentos:

La implementación de esta funcionalidad se divide en tres partes. La primera se basa en hacer la petición a la función del servidor pasándole como parámetro en la url el id del usuario (ver Figura 65):

```

http://192.168.1.44/geoturistapp/monumentos_usuario.php?id_usuario="

```

Figura 65 - Url obtener monumentos de usuario

Esto se hace mediante la función getJSON explicada en la siguiente imagen:

```
private void getJSON(final String urlWebService) {

    class GetJSON extends AsyncTask<Void, Void, String> {

        @Override
        protected void onPreExecute() { super.onPreExecute(); }

        @Override
        protected void onPostExecute(String s) {
            super.onPostExecute(s);
            //Toast.makeText(getActivity().getApplicationContext(), s, Toast.LENGTH_SHORT).show();
            try {
                loadIntoListView(s);
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }

        @Override
        protected String doInBackground(Void... voids) {

            try {
                //creamos la URL
                URL url = new URL(urlWebService);

                //Se abre la URL usando HttpURLConnection
                HttpURLConnection con = (HttpURLConnection) url.openConnection();

                //StringBuilder para leer la información
                StringBuilder sb = new StringBuilder();

                BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(con.getInputStream()));

                String json;

                while ((json = bufferedReader.readLine()) != null) {
                    sb.append(json + "\n");
                }

                //Retornamos los datos leídos de la petición realizada. En este caso los datos del json
                return sb.toString().trim();
            } catch (Exception e) {
                return null;
            }
        }
    }
}
```

Figura 66 - Método getJSON

Una vez realizada la petición y con los datos recogidos, ahora se vuelcan en el ListView de la vista (ver Figura 66).

Para volcarlo se crea un array de strings para agregar al ListView. Después se recorre el array de Json obtenido en el proceso anterior y dentro del bucle for se obtiene el objeto Json del array anterior. Por último, se obtienen los parámetros para pintarlos en el ListView.

```

private void loadIntoListView(String json) throws JSONException {
    JSONArray jsonArray = new JSONArray(json);

    //Creamos una array de strings para agregarlo al ListView
    final String[] lugares = new String[jsonArray.length()];
    final String[] id_lugares = new String[jsonArray.length()];

    //Recorremos los datos obtenidos en el json
    for (int i = 0; i < jsonArray.length(); i++) {

        //Obtenemos el objeto json del array de jsons
        JSONObject obj = jsonArray.getJSONObject(i);

        //obtenemos los parametros que queremos del objeto json
        lugares[i] = obj.getString("nombre_lugar");
        id_lugares[i] = obj.getString("id_lugar");

    }

    //Creamos un array adapter para almacenar los datos anteriores
    ArrayAdapter<String> arrayAdapter = new ArrayAdapter<>(getActivity(), android.R.layout.simple_list_item_1, lugares);

    if(lugares != null) {
        lista_monumentos.setAdapter(arrayAdapter);
        tv_visitados.setText(String.valueOf(lugares.length));

        lista_monumentos.setOnItemClickListener((parent, view, position, id) -> {
            Log.d("tag", "click", String.valueOf(position));

            Fragment_VerMonumento fragment_monumento = new Fragment_VerMonumento();

            Bundle bundl = new Bundle();

            bundl.putString("nombre_lugar", lugares[position]);
            bundl.putString("id_lugar", id_lugares[position]);
            bundl.putString("id_usuario", username);

            fragment_monumento.setArguments(bundl);

            FragmentTransaction transaction = getFragmentManager().beginTransaction();

            transaction.replace(R.id.contenedor, fragment_monumento);
            //transaction.addToBackStack(null);

            transaction.commit();

        });
    }
}

```

Figura 67 - Volcado en ListView

De este modo, se muestran los monumentos que el usuario tiene añadidos en la lista que se muestra en la vista (ver Figura 67).

Este proceso es genérico para las demás funcionalidades, es decir, se realizan las mismas funciones y se vuelcan los datos dependiendo de la petición que se haga.

Éstas funciones son: Mis comentarios, Ver multimedia y Descubre monumentos.

e) Descubre monumentos:

Esta función es la base principal de la aplicación, gracias a ella es posible detectar los lugares cercanos disponibles que están guardados en la base de datos.

Se explicará más adelante en el apartado de Algoritmos.

f) Realizar valoración y comentarios:

En estas dos funcionalidades también se utiliza la función getJSON (ver Figura 68)

```

// Funcionalidad del boton de valorar
btn_valorar.setOnClickListener((v) -> {

    comentario = et_comentario.getText().toString();
    valoracion = et_valorar.getText().toString();

    url_valorar = entorno + "id_usuario=" + id_usuario + "id_lugar=" + id_lugar + "valoracion=" + valoracion
    + "comentario=" + comentario;
    // Al hacer click en el botón de Valorar comprobamos que los campos valorar y comentario (no es necesario) están completos.
    // Necesitamos el id usuario, id lugar, valoracion y comentario

    if(!TextUtils.isEmpty(valoracion)) {
        getJSON(url_valorar, "valorar");
        //Toast.makeText(getActivity().getApplicationContext(), url_valorar, Toast.LENGTH_SHORT).show();

        Fragment_VerMonumento fragment_ver_monumento = new Fragment_VerMonumento();

        Bundle bundl = new Bundle();

        bundl.putString("nombre_lugar", nombre_lugar);
        bundl.putString("id_lugar", id_lugar);
        bundl.putString("id_usuario", id_usuario);

        fragment_ver_monumento.setArguments(bundl);

        FragmentTransaction transaction = getFragmentManager().beginTransaction();

        transaction.replace(R.id.contenedor, fragment_ver_monumento);
        transaction.addToBackStack(null);

        transaction.commit();
    }
    else{
        et_valorar.setError("Valoración requerida");
    }
})

```

Figura 68 - Código realizar valoración y comentarios

g) Ver monumento:

Esta funcionalidad es la que más peticiones tiene.

```

private static String entorno = "http://192.168.1.44/geoturistapp/ver_monumento_usuario.php?";
//private static String entorno = "http://socmica.000webhostapp.com/proyectos/geoturistapp/ver_monumento_usuario.php?";
private static String entorno_com = "http://192.168.1.44/geoturistapp/lista_com_usuario.php?";
//private static String entorno_com = "http://socmica.000webhostapp.com/proyectos/geoturistapp/lista_com_usuario.php?";
private static String entorno_add_lugar = "http://192.168.1.44/geoturistapp/add_lugar_usuario.php?";
//private static String entorno_add_lugar = "http://socmica.000webhostapp.com/proyectos/geoturistapp/add_lugar_usuario.php?";

```

Figura 69 - Código ver monumento

Cada una de ellas se pasa a la función getJSON para obtener los datos solicitados por el usuario.

Se pueden visualizar los comentarios que posee el lugar visitado.

Se puede agregar el monumento que se está visitando a la lista personal del usuario.

Además, posee dos botones, Multimedia y Valorar (ver Figura 69) que redirigen a las respectivas vistas. Se les pasan todos los datos para poder hacer las peticiones al servidor y guardar la valoración y comentarios. Y ver los distintos archivos multimedia.

```

btn_multimedia.setOnClickListener((v) -> {
    // Create new fragment and transaction
    Fragment_VerMultimedia fragment_multimedia = new Fragment_VerMultimedia();
    Bundle bundl = new Bundle();

    bundl.putString("id_lugar", id_lugar);
    bundl.putString("nombre_lugar", nombre_lugar);
    bundl.putString("id_usuario", id_usuario);
    bundl.putBoolean("agregado", agregado);

    fragment_multimedia.setArguments(bundl);

    FragmentTransaction transaction = getFragmentManager().beginTransaction();

    transaction.replace(R.id.contenedor, fragment_multimedia);
    transaction.addToBackStack(null);

    transaction.commit();

});

btn_valorar.setOnClickListener((v) -> {
    // Create new fragment and transaction
    Fragment_Valorar fragment_valorar = new Fragment_Valorar();

    Bundle bundl = new Bundle();

    bundl.putString("nombre_lugar", nombre_lugar);
    bundl.putString("id_lugar", id_lugar);
    bundl.putString("id_usuario", id_usuario);

    bundl.putString("val_total", val_total);
    bundl.putString("num_val", num_val);

    bundl.putBoolean("agregado", agregado);

    fragment_valorar.setArguments(bundl);

    FragmentTransaction transaction = getFragmentManager().beginTransaction();

    transaction.replace(R.id.contenedor, fragment_valorar);
    transaction.addToBackStack(null);

    transaction.commit();

});

```

Figura 70 - Código botones multimedia y valoración

Se crea un bundle para meter todos los datos necesarios en las respectivas vistas y poder utilizarlos en ellas (ver Figura 70).

h) Ver perfil:

La implementación de esta función es muy sencilla. Simplemente se muestran los datos del usuario en la vista, además de las valoraciones y comentarios que ha realizado (ver Figura 71).

```

public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {

    ((HomeActivity) getActivity())
        .setActionBarTitle("Mi Perfil");

    View v = inflater.inflate(R.layout.fragment_fragment_perfil, container, attachToRoot: false);

    tv_nick = v.findViewById(R.id.tv_nick);
    tv_nombre = v.findViewById(R.id.tv_nombre);
    tv_apellidos = v.findViewById(R.id.tv_apellidos);
    tv_email = v.findViewById(R.id.tv_email);
    tv_comentarios = v.findViewById(R.id.tv_comentarios);
    tv_valoraciones = v.findViewById(R.id.tv_valoraciones);

    // Recibimos los datos del activity HomeActivity para poder mostrarlos en el Fragment de Perfil.
    Bundle bundl = getArguments();

    nick = bundl.getString( key: "nick");
    nombre = bundl.getString( key: "nombre");
    apellidos = bundl.getString( key: "apellidos");
    email = bundl.getString( key: "email");
    comentarios = bundl.getString( key: "comentarios");
    valoraciones = bundl.getString( key: "valoraciones");

    // Los escribimos en los textview de nuestro layout
    tv_nick.setText(nick);
    tv_nombre.setText(nombre);
    tv_apellidos.setText(apellidos);
    tv_email.setText(email);
    tv_comentarios.setText(comentarios);
    tv_valoraciones.setText(valoraciones);

    return v;
}

```

Figura 71 - Código ver perfil

i) Cerrar sesión:

Esta funcionalidad está implementada en el menú lateral de la aplicación.

Al pulsar sobre el botón del menú, se borrará la información del objeto SharedPreferences. Se redirigirá al usuario a la pantalla de login por si desea autenticarse con otro tipo de usuario o simplemente cerrar la aplicación (ver Figura 72).

```

else if (id == R.id.nav_logout) {

    // Elemento del menu lateral para cerrar la sesión.

    final AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder( context: this);

    alertDialogBuilder.setTitle("¿Desea salir de la aplicación?");
    //alertDialogBuilder.setIcon(R.drawable.question);
    alertDialogBuilder.setMessage("Saldrás de GeoturistApp");
    alertDialogBuilder.setCancelable(false);

    alertDialogBuilder.setPositiveButton( text: "Si", (arg0, arg1) -> {
        // BORRAMOS EL USERNAME Y PASSWORD en SharedPreferences
        SharedPreferences.Editor editor = getSharedPreferences( USER_PASS, MODE_PRIVATE).edit();
        editor.putString( key: "username", s: null);
        editor.putString( key: "password", s: null);
        editor.apply();
        Intent intentLogout = new Intent( packageContext: HomeActivity.this, MainActivity.class);
        HomeActivity.this.startActivity(intentLogout);
    });

    alertDialogBuilder.setNegativeButton( text: "No", (dialog, which) -> {
        dialog.cancel();
    });

    AlertDialog alertDialog = alertDialogBuilder.create();
    alertDialog.show();
}

DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
drawer.closeDrawer( GravityCompat.START);
return true;
}

```

Figura 72 - Código cerrar sesión

7.3 Base de datos

Mediante una función definida en el fichero `config_conexion.php` se establece la conexión con la base de datos. En este caso como se tienen dos entornos (uno en local, y otro en remoto), se ha definido una condición a la hora de realizar la conexión.

```
function conecta()
{
    $local = true;

    if($local == true)
    {
        $host='localhost';
        $usuario='root';
        $clave='';
        $db='geoturistapp';
    }
    else
    {
        $host='localhost';
        $usuario='id1773399_david';
        $clave='geoturistapp';
        $db='id1773399_geoturistapp';
    }

    $conexion=mysqli_connect($host,$usuario,$clave,$db);

    return $conexion;
}
```

Figura 73 - Conexión con BBDD

La función `conecta` es indispensable para realizar la conexión con la base de datos, y es necesaria para realizar cualquier petición que se haga (ver Figura 73).

7.4 Algoritmos

El algoritmo que se utiliza para obtener la geolocalización dentro de la aplicación móvil es la parte vital para que todo funcione.

Se divide en tres partes:

1. Obtener coordenadas mediante el sensor GPS del dispositivo móvil

```

// Si el usuario hace click en el botón de geolocalizar comenzamos a obtener las coordenadas desde el dispositivo móvil.
private void locationStart() {
    LocationManager locationManager = (LocationManager) getActivity().getSystemService(Context.LOCATION_SERVICE);
    Localizacion local = new Localizacion();
    Local.setMainActivity(Local.getMainActivity());
    final boolean gpsEnabled = locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);
    if (!gpsEnabled) {
        Intent settingsIntent = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
        startActivity(settingsIntent);
    }
    if (ActivityCompat.checkSelfPermission(getActivity(), Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
        && ActivityCompat.checkSelfPermission(getActivity(), Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(getActivity(), new String[]{Manifest.permission.ACCESS_FINE_LOCATION}, REQUEST_CODE);
        return;
    }
    locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, minTime 0, minDistance 0, (LocationListener) local);
    locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, minTime 0, minDistance 0, (LocationListener) local);
    estado.setText("Buscando...");
}

```

Figura 74 - Obtener coordenadas

Se comprueba si el GPS del dispositivo está activo (ver Figura 74). En caso de no estarlo, se redirigirá al usuario a las opciones de ubicación de su dispositivo móvil para activarla.

2. Cada vez que las coordenadas cambian debido a un cambio de ubicación

```

/* Aquí empieza la Clase Localizacion */
public class Localizacion implements LocationListener {
    Fragment_Descubre descubre_frag;

    public Fragment_Descubre getMainActivity() { return descubre_frag; }

    public void setMainActivity(Fragment_Descubre descubre_frag) {
        this.descubre_frag = descubre_frag;
    }

    @Override
    public void onLocationChanged(Location loc) {
        // Este metodo se ejecuta cada vez que el GPS recibe nuevas coordenadas
        // debido a la deteccion de un cambio de ubicacion

        loc.getLatitude();
        loc.getLongitude();

        String text = "\n Lat = "
            + loc.getLatitude() + "\n Long = " + loc.getLongitude();
        estado.setText(text);
        obtener_monumentos_coordenadas(String.valueOf(loc.getLongitude()), String.valueOf(loc.getLatitude()));
    }
}

```

Figura 75 - Cambio de localización

Si el usuario se mueve, cambiará su ubicación, por lo que es importante que las coordenadas cambien, en caso contrario obtendría una única vez las coordenadas y no se actualizaría con los cambios de posición del usuario. Impidiendo así que se muestren los lugares cercanos nuevos (ver Figura 75).

3. Según la actividad del sensor GPS se mostrará una información u otra (ver Figura 76)

```

@Override
public void onProviderDisabled(String provider) {
    // Este metodo se ejecuta cuando el GPS es desactivado
    estado.setText("GPS Desactivado");
}

@Override
public void onProviderEnabled(String provider) {
    // Este metodo se ejecuta cuando el GPS es activado
    estado.setText("GPS Activado");
}

@Override
public void onStatusChanged(String provider, int status, Bundle extras) {
    switch (status) {
        case LocationProvider.AVAILABLE:
            Log.d("tag: "debug", "msg: "LocationProvider.AVAILABLE");
            break;
        case LocationProvider.OUT_OF_SERVICE:
            Log.d("tag: "debug", "msg: "LocationProvider.OUT_OF_SERVICE");
            break;
        case LocationProvider.TEMPORARILY_UNAVAILABLE:
            Log.d("tag: "debug", "msg: "LocationProvider.TEMPORARILY_UNAVAILABLE");
            break;
    }
}
}

```

Figura 76 - Actividad sensor GPS

En la parte del servidor, tras mandarse la latitud y la longitud mediante la url (ver Figura 77).

```
private static String entorno = "http://192.168.1.44/geoturistapp/coordenadas_usuario.php?";
```

Figura 77 - Obtener coordenadas usuario

Se comparará con cada una de las coordenadas almacenadas en la base de datos (ver Figura 78).

```

$response = array();
$obtener_lugares = mysqli_query($conexion,"SELECT * FROM lugares");
if($obtener_lugares){
    while($fila = mysqli_fetch_array($obtener_lugares))
    {
        $punto2 = [$fila['longitud'], $fila['latitud']];
        $distancia = distance($punto1[0], $punto1[1], $punto2[0], $punto2[1]);
        //echo $distancia." kilometros <br>";
        if($distancia < 1){ // Si la distancia es menor a 1 kilometro se añade ese monumento al array
            $temp = [
                'id_lugar' => $fila['id_lugar'],
                'nombre_lugar' => $fila['nombre']
            ];
            array_push($response, $temp);
        }
    }
    echo json_encode($response);
}
}

```

Figura 78 - Comparación de coordenadas

Si las coordenadas de los lugares comparados se encuentran dentro del radio no superior a 1km, se añadirá el lugar a un array que será enviado a la app.

8. Conclusiones y trabajo futuro

8.1 Conclusiones

A continuación, se desarrollan las conclusiones llevadas a cabo al finalizar el proyecto:

- Se ha creado un sistema formado por dos aplicaciones de forma independiente que poseen un almacenamiento de datos común. Gracias a ellas se pueden descubrir monumentos y lugares en una ciudad.
- Una de las aplicaciones creadas es una página web que sirve para administrar el contenido de la base de datos. Introduciendo lugares y monumentos por un usuario administrador.
- Otra aplicación es móvil, enfocada para que los usuarios utilizando la geolocalización descubran los monumentos y lugares que hay a su alrededor.
- La base de datos es la encargada de almacenar los lugares y monumentos, su información y los datos multimedia de cada uno de ellos.
- Se pretendió utilizar notificaciones para avisar al usuario cuándo se acercase a un monumento, pero finalmente no se ha implementado

8.1 Conclusions

Next, the conclusions carried out at the end of the project are developed:

- A system has been created consisting of two applications independently that have a common data storage. Thanks to them you can discover monuments and places in a city.
- One of the created applications is a web page that serves to manage the contents of the database. Introducing places and monuments by an administrator user.
- Another application is mobile, focused so that users using geolocation discover the monuments and places around them.
- The database is in charge of storing the places and monuments, their information and the multimedia data of each one of them.
- It was intended to use notifications to warn the user when approaching a monument, but finally it has not been implemented

8.2 Trabajo futuro

Este trabajo deja abiertas una serie de líneas de trabajo futuro que permitirán avanzar, extenderlo y mejorarlo. A continuación, se indican estas líneas de trabajo:

- **Gestión de las incidencias.** En la parte web ya se encuentra implementado, pero en la aplicación móvil se agregará un elemento en el menú lateral que llevará a una vista nueva. Esta contendrá un formulario para reportar errores o pedir información sobre algún lugar.
- **Introducir un mapa.** En la aplicación móvil se añadirá un mapa con los monumentos que hay disponibles, así el usuario podrá saber con certeza que monumentos hay a su alrededor.
- **Notificaciones push.** A la hora de buscar monumentos es una idea interesante añadir notificaciones para avisar al usuario de que se han encontrado monumentos que estén en la base de datos. Esta función podrá ejecutarse en segundo plano y avisar en el momento preciso, para no tener que estar continuamente obteniendo la geolocalización a mano.
- **Edición de perfil.** Se implementará la opción de editar los datos personales del usuario, por si quiere cambiar de dirección de correo electrónico.
- **Subida de imágenes de perfil.** Se dará la posibilidad al usuario de añadir una foto de perfil para identificar mejor a los usuarios.

8.2 Future work

This paper leaves a series of future work lines undone that will allow it to advance, extend and improve. These work lines are:

- **Incident management.** In the website it is already implemented, but in the mobile application an element will be added in the side menu that will lead to a new view. This view will contain a form to report mistakes or request information about some places.
- **Introduce a map.** In the mobile application a map with the available monuments will be added, so the user can know with certainty which monuments are around him.
- **Push notifications.** When looking for monuments it is an interesting idea to add notifications to warn the user that monuments have been found in the database. This function can run in the background and warn at the precise time, so you do not have to continually get the geolocation by hand.
- **Profile editing.** The option of editing the personal data of the user will be implemented, in case he wants to change his email address.
- **Upload of profile images.** The user will be given the possibility of adding a profile picture to better identify the users.

9. Aportaciones

En este apartado se hablará de las aportaciones que se han hecho al trabajo, incluyendo además los problemas y dificultades encontrados a lo largo del desarrollo.

Después de desarrollar la aplicación (tanto la parte web como la parte móvil) y explicando todos los puntos anteriores en la memoria, se han obtenido las siguientes conclusiones. Desde el principio del desarrollo es muy importante tener en cuenta una previsión de tiempos. A medida que se programa es importante planificarse para saber hasta dónde se puede llegar. Muchas veces uno cree que ya ha finalizado una funcionalidad, y en realidad hay todavía algunos fallos que no se habían previsto.

También he podido comprobar que aprender un lenguaje tiene sus dificultades, pero poco a poco te vas sintiendo más y más cómodo con él. El programar con Android Studio te ayuda a tener un entorno dónde se pueden realizar tanto el desarrollo como las pruebas.

Muchos de los problemas encontrados a lo largo del proyecto me han enseñado a buscar diferentes estrategias para solventarlos.

La parte web ha sido más fácil de implementar ya que la tecnología ya la conocía, aunque se han aplicado nuevas técnicas y métodos para mejorar la funcionalidad.

En general el proceso de desarrollo ha sido bueno y he aprendido mucho. Tanto a nivel teórico como práctico, aplicando tecnologías nuevas no vistas antes en la carrera. Y cada día que pasa hay algo nuevo que aprender

Bibliografía

- [1] <<Android Developers>>. [Online] Disponible en: <https://developer.android.com/>
- [2] <<Apache Friends. Xampp>>. [Online] Disponible en:
<https://www.apachefriends.org/es/index.html>
- [3] <<PHP docs>>. [Online] Disponible en: <http://php.net/manual/es/intro-whatism.php>
- [4] <<W3Schools Web Tutorials>> [Online] Disponible en: <https://www.w3schools.com/>
- [5] <<Stack Overflow Español>> [Online] Disponible en: <https://es.stackoverflow.com/>
- [6] <<Stack Overflow Inglés>> [Online] Disponible en: <https://stackoverflow.com/>
- [7] <<MySQL Examples>> [Online] Disponible en:
http://www.ntu.edu.sg/home/ehchua/programming/sql/mysql_beginner.html
- [8] <<Android Studio tutorials>> [Online] Disponible en:
<https://www.androidauthority.com/android-studio-tutorial-beginners-637572/>
- [9] <<Balsamiq Mockups>> [Online] Disponible en: <https://balsamiq.cloud/>
- [10] <<Pokémon Go >> [Online] Disponible en:
<https://play.google.com/store/apps/details?id=com.nianticlabs.pokemongo>
- [11] <<Google Goggles>> [Online] Disponible en:
<https://play.google.com/store/apps/details?id=com.google.android.apps.unveil&hl=es>
- [12] <<Myooni Guía turística global >> [Online] Disponible en:
(<https://play.google.com/store/apps/details?id=com.myooni>
- [13] <<Visit A City>> [Online] Disponible en:
<https://play.google.com/store/apps/details?id=com.visitacity.visitacityapp>
- [14] <<Monumento Guida Turística>> [Online] Disponible en:
<https://play.google.com/store/apps/details?id=com.monumento.app>
- [15] <<Google Maps Platform>> [Online] Disponible en:
<https://cloud.google.com/maps-platform/?hl=es>
- [16] <<GitHub>> [Online] Disponible en: <https://github.com>

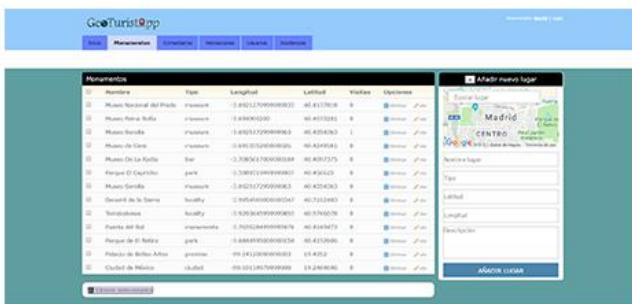
Apéndice 2: Manual de usuario

Manual de usuario para la aplicación web:



Login en la aplicación web GeoTuristApp
Estas credenciales se pueden cambiar en la base de datos y puede haber varios usuarios. Por defecto usuario : admin
contraseña : admin

En la página de inicio se pueden ver los lugares dados de alta, los comentarios, valoraciones y usuarios de la base de datos. El usuario administrador podrá dirigirse a cualquiera de las secciones.



En la sección de monumentos se pueden ver todos los lugares dados de alta, si se hace click en el icono de ver llevará al usuario a la vista del monumento seleccionado. Si se le da a borrar, se eliminará ese lugar

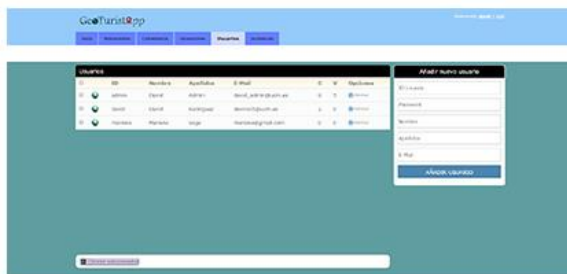
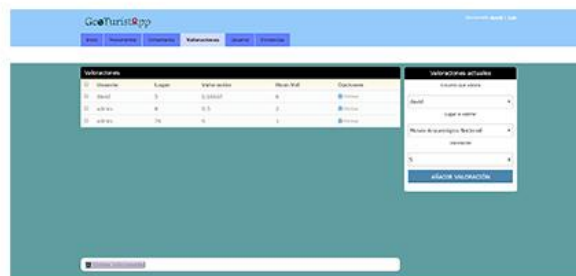
Si se le da al botón de "+" se podrá añadir un nuevo lugar, se busca en el mapa, se selecciona y se rellenan todos los campos. Además se pueden añadir elementos multimedia si el usuario lo desea





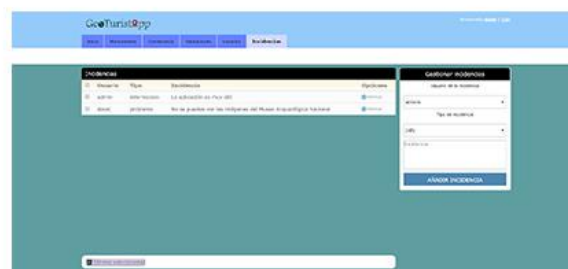
En el apartado de comentarios el usuario administrador podrá ver los comentarios realizados por los usuarios de la app. Podrá verlos o eliminarlos.

En el apartado de valoraciones el usuario administrador podrá ver las valoraciones realizadas por los usuarios de la app. Podrá verlas o eliminarlas.



En el apartado de usuarios el usuario administrador podrá ver los los usuarios dados de alta en la app. Podrá verlos o eliminarlos.

En el apartado de incidencias el usuario administrador podrá ver las incidencias realizadas por los usuarios de la app. Podrá verlas o eliminarlas.

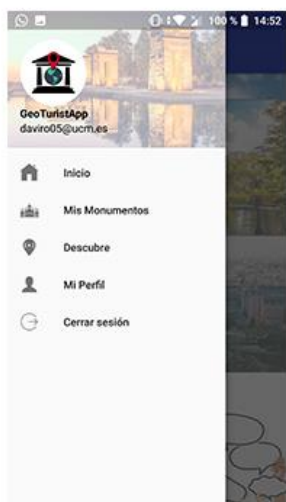


Manual de usuario para la aplicación móvil:



Nada más abrir la app hay que crearse una cuenta. El usuario ingresará sus datos personales y cuándo haya completado el registro podrá autenticarse en la aplicación.

Una vez autenticado se mostrará la pantalla principal de la app.



Dentro de la app hay disponible un menú en la parte lateral que muestra las distintas opciones que permite realizar la app.

En inicio se muestran 3 botones Mis monumentos, Descubre y Mis comentarios.

Además hay una opción de Mi Perfil, para ver los datos del usuario autenticado.



A la hora de buscar monumentos se debe activar la ubicación en ajustes para que el GPS pueda tomar los datos.

Además en los monumentos se puede ver la información de cada uno, y también ver los elementos multimedia si dispone de ellos.



Código de las aplicaciones

Aplicación web

<https://github.com/daviro05/geoturistAppWeb>

Aplicación móvil

<https://github.com/daviro05/geoturistAppMov>