



Universidad de Valladolid

**ESCUELA DE INGENIERÍA INFORMÁTICA
DE SEGOVIA**

**Grado en Ingeniería Informática
de Servicios y Aplicaciones**

Aplicación web de puntos de interés geolocalizados

Alumno: Eva María González Herrero

Tutor: Fernando Díaz Gómez

Fecha: Julio 2024

Aplicación web de puntos de interés geolocalizados

Eva María González Herrero

Índice general

Lista de figuras	V
Lista de tablas	VII
Resumen	XI
Abstract	XIII
I Memoria del Proyecto	1
1. Introducción	3
1.1. Organización del documento	3
1.2. Contexto y justificación	4
1.3. Motivación	4
1.4. Objetivos del trabajo	6
1.5. Alcance	6
1.5.1. Árbol de características	7
1.5.2. Restricciones y limitaciones	8
1.6. Conceptos clave y soporte tecnológico	8
1.6.1. Puntos de interés	8
1.6.2. Rutas	8
1.6.3. Georreferenciación	8
1.6.4. Soporte tecnológico actual	9
1.7. Entorno de aplicación	10
1.7.1. Estado del arte	10
1.7.2. Herramientas utilizadas	17
2. Gestión del Proyecto	25
2.1. Metodología	25
2.1.1. Proceso de desarrollo	25
2.1.2. Características generales de la metodología incremental	26
2.1.3. Justificación de la elección de la metodología incremental	27
2.1.4. Comparación con otras metodologías	28

2.2.	Planificación temporal	28
2.2.1.	Definición de los incrementos	28
2.2.2.	Incremento 1 (Planificación y diseño)	30
2.2.3.	Incremento 2 (Implementación de funcionalidades básicas)	31
2.2.4.	Incremento 3 (Finalización y pruebas)	32
2.3.	Presupuesto económico	34
2.3.1.	Fórmulas de costos	34
2.3.2.	Recursos hardware	35
2.3.3.	Costo total de hardware	35
2.3.4.	Recursos software	36
2.3.5.	Recursos humanos	36
2.3.6.	Otros costos adicionales	37
2.3.7.	Contingencias y margen	38
2.3.8.	Presupuesto total	38
2.4.	Costo real	38
2.4.1.	Recursos hardware revisados	38
2.4.2.	Costo total de hardware revisado	39
2.4.3.	Otros costos adicionales revisados	39
2.4.4.	Presupuesto total revisado	40
II	Documentación técnica	41
3.	Análisis	43
3.1.	Requisitos	43
3.1.1.	Requisitos Funcionales (RF)	44
3.1.2.	Requisitos No Funcionales (RNF)	45
3.1.3.	Atributos de Calidad	46
3.1.4.	Atributos de Información	47
3.1.5.	Restricciones	48
3.2.	Casos de Uso	49
3.2.1.	Diagrama de Casos de Uso	49
3.2.2.	Tablas de Casos de Uso	51
4.	Diseño	57
4.1.	Arquitectura	57
4.1.1.	Arquitectura Lógica	58
4.1.2.	Arquitectura Física	58
4.2.	Diseño de Datos	60
4.2.1.	Diseño lógico de la base de datos	60
4.2.2.	Diccionario de Datos	62
4.2.3.	Perfiles de Acceso	64
4.3.	Diagramas de secuencia	66

4.3.1.	Estereotipos de clase de línea de vida	66
4.3.2.	Diagrama de mostrar puntos cercanos	67
4.3.3.	Diagrama de exportacion	69
5.	Implementación	71
5.1.	Introducción	71
5.1.1.	Inicialización y finalización	74
5.2.	Consultas seguras a la base de datos	76
5.2.1.	Descripción	76
5.2.2.	Ejemplo de inserción segura	76
5.3.	Implementación de mapas con Leaflet	77
5.3.1.	Configuración Básica para la Instalación	77
5.3.2.	Crear un Contenedor para el Mapa	77
5.3.3.	Inicializar el Mapa	78
5.3.4.	Añadir marcadores y capas	78
5.4.	Importación de Archivos KML	80
5.4.1.	Subida del Archivo KML	80
5.4.2.	Validación y Carga del Archivo KML	80
5.4.3.	Procesamiento de Puntos de Interés (PoI)	80
5.4.4.	Inserción en la Base de Datos	81
5.4.5.	Manejo de Rutas y Elementos Desconocidos	81
5.5.	Exportación a KML	82
5.5.1.	Selección de Puntos en el Mapa	82
5.5.2.	Interfaz de Selección de Puntos	82
5.5.3.	Exportación a KML	82
5.5.4.	Cancelación y Restauración de Estado	82
5.5.5.	Implementación en JavaScript	83
5.6.	Retos y Soluciones	83
5.6.1.	Descripción	83
5.7.	Conclusiones	86
5.7.1.	Resumen	86
5.7.2.	Lecciones aprendidas	87
6.	Pruebas	89
6.1.	Pruebas de caja blanca	89
6.2.	Pruebas de caja negra	91
7.	Conclusiones y trabajo futuro	97
7.1.	Conclusiones	97
7.2.	Trabajo futuro	98

III	Manuales de la Aplicación	99
8.	Manual de Usuario	101
8.1.	Manual de usuario	101
8.1.1.	Introducción	101
8.1.2.	Usuario registrado con puntos de interés propios	102
8.1.3.	Usuario registrado sin puntos de interés propios	109
8.2.	Manual de administración	109
8.2.1.	Introducción	109
8.2.2.	Funcionalidades del administrador	109
IV	Apéndices	111
A.	Anexos	113
A.1.	Información complementaria	113
A.1.1.	Definición de siglas y abreviaturas	113
	Bibliografía y Webgrafía	115

Índice de figuras

1.1. Árbol de características.	7
1.2. Imagen de marca de HTML5, CSS y JavaScript.	17
1.3. Imagen de marca de Leaflet.	19
1.4. Imagen de marca de PHP.	20
1.5. Imagen de marca de MySQL.	21
1.6. Imagen de marca de Ubuntu.	22
2.1. Modelo de desarrollo incremental	26
2.2. Incremento 1	31
2.3. Incremento 2	32
2.4. Incremento 3	33
3.1. Diagrama de casos de uso	50
4.1. Arquitectura logica MVC.	58
4.2. Arquitectura física cliente-servidor.	59
4.3. Modelo entidad-relación (ER).	60
4.4. Esterotipos usados en un diagrama de secuencia. De izquierda a derecha: entidad, límite y control.	66
4.5. Diagrama de secuencia de mostrar puntos cercanos al entrar a explorar . . .	68
4.6. Diagrama de secuencia de exportar puntos de interés	70
5.1. Estructura de las carpetas.	72
5.2. Icono para puntos del usuario favoritos (<i>iconoMisSitiosMG.png</i>).	85
5.3. Icono para puntos del usuario (<i>iconoMisSitios.png</i>).	86
5.4. Icono para puntos de otros usuarios favoritos (<i>iconoOtrosSitiosMG.png</i>).	86
5.5. Icono para puntos de otros usuarios (<i>iconoOtrosSitios.png</i>).	86
8.1. Captura de la aplicación Web: Registro	102
8.2. Captura de la aplicación Web: Inicio de sesión o <i>login</i>	102
8.3. Captura de la aplicación Web: Mapa Explorar	103
8.4. Captura de la aplicación Web: Buscador	103
8.5. Captura de la aplicación Web: Clic sobre el mapa	104
8.6. Captura de la aplicación Web: Página de agregar punto	104

8.7. Captura de la aplicación Web: Clic sobre un punto	105
8.8. Captura de la aplicación Web: Página Punto individual	105
8.9. Captura de la aplicación Web: Pulsar sobre “añadir a favoritos”	106
8.10. Captura de la aplicación Web: Visualización del punto como favorito	106
8.11. Captura de la aplicación Web: Exportar a KML	107
8.12. Captura de la aplicación Web: Mis sitios	108
8.13. Captura de la aplicación Web: Subir KML	108

Índice de tablas

2.1.	Costos de uso de los recursos de hardware	36
2.2.	Resumen del presupuesto total	38
2.3.	Costos de uso de los recursos de hardware revisados	39
2.4.	Resumen del presupuesto total revisado	40
3.1.	Requisitos Funcionales	45
3.2.	Requisitos No Funcionales	45
3.3.	Atributos de Calidad	46
3.4.	Atributos de Información	47
3.5.	Restricciones	48
3.6.	CU-01. Registro de usuario	51
3.7.	CU-02. Creación de un sitio de interés	52
3.8.	CU-03. Importar un sitio de interés mediante un archivo KML	53
3.9.	CU-04. Gestionar punto de interés	54
3.10.	CU-05. Exportar puntos de interés a KML	55
4.1.	Diccionario de datos de la entidad “Persona”	62
4.2.	Diccionario de datos de la entidad “PoI”	62
4.3.	Diccionario de datos de la entidad “Comentarios”	63
4.4.	Diccionario de datos de la entidad “Favoritos”	63
4.5.	Diccionario de datos de la entidad “ImágenesPoI”	64
6.1.	Prueba de inicio de sesión de usuarios	91
6.2.	Prueba de creación de puntos de interés	92
6.3.	Prueba de importación de puntos de interés	93
6.4.	Prueba de exportación de puntos de interés	93
6.5.	Prueba de modificación de puntos de interés	94
6.6.	Prueba de eliminación de puntos de interés	95
6.7.	Prueba de redirección al <i>login</i> tras expiración de sesión	96
6.8.	Prueba de redirección al inicio tras URL inválida o no autorizada	96
A.1.	Tabla de siglas y abreviaturas.	113

Agradecimientos

Quiero expresar mi agradecimiento a mis padres, mi hermano y mis abuelos, gracias por su apoyo a lo largo de este camino. Su presencia y aliento han sido fundamentales en mi crecimiento personal.

A mis amigos por estar siempre ahí, compartiendo risas, desafíos y éxitos juntos. Su amistad ha sido increíblemente importante para mí.

También quiero reconocer a mi tutor Fernando, por su orientación y apoyo durante este tiempo. Sus consejos han sido clave para mi desarrollo académico.

A cada uno de vosotros, gracias por formar parte de mi vida y por su contribución invaluable a mi desarrollo y éxito académico.

Resumen

Dado que en la vida moderna la mayoría de los trabajos requieren de estar sentado en una oficina, hacen muy difícil disfrutar del aire libre y de lugares de ocio fuera de nuestro ambiente doméstico. Con esta aplicación, se busca fomentar que las personas incorporen a su vida nuevas formas de disfrutar de diversas experiencias recreativas, ya sea jugar en un billar, tomar una cerveza en un bar, visitar un museo o explorar nuevos paisajes.

El objetivo es crear una aplicación web que permita gestionar puntos de interés, tanto urbanos como rurales, y que ofrezca a los usuarios la posibilidad de marcar sus lugares favoritos. Los usuarios podrán subir fotos de los lugares visitados y añadir una descripción, creando así una comunidad de recomendaciones y experiencias compartidas.

Palabras claves: Aplicación web, puntos de interés, gestión de lugares, experiencias recreativas, favoritos, KML, datos geoespaciales.

Abstract

Given that most modern jobs require sitting in an office, it is very difficult to enjoy the outdoors and leisure places outside our domestic environment. This application aims to encourage people to incorporate new ways of enjoying various recreational experiences into their lives, whether it's playing pool, having a beer at a bar, visiting a museum, or exploring new landscapes.

The goal is to create a web application that allows managing points of interest, both urban and rural, and offers users the possibility to mark their favorite places. Users will be able to upload photos of the places visited and add a description, thus creating a community of shared recommendations and experiences.

Keywords: Web application, points of interest, place management, recreational experiences, favorites, KML, geospatial data.

Parte I

Memoria del Proyecto

Capítulo 1

Introducción

Este Trabajo de Fin de Grado (TFG) se centra en el desarrollo de una plataforma web que permitirá a los usuarios interactuar con puntos de interés geolocalizados de una manera única y personalizada. La plataforma ofrecerá a los usuarios la posibilidad de crear sus propios puntos de interés, así como de importar y exportar estos puntos mediante archivos con formato *Keyhole Markup Language* (KML). Además, se fomentará la interacción entre los usuarios a través de la posibilidad de comentar y guardar los puntos de interés de otros usuarios. Este proyecto representa una oportunidad para aplicar y mejorar mis habilidades técnicas, al tiempo que ofrece una nueva herramienta para promover el turismo en España.

1.1. Organización del documento

La memoria del proyecto comienza con una introducción que establece la estructura organizativa del documento y presenta el contexto actual del turismo digital 1.2, justificando la necesidad de desarrollar una aplicación específica para este sector. Se destacan las motivaciones detrás del proyecto, los objetivos del trabajo se delimitan con claridad 1.4, y se define el alcance del proyecto mediante un árbol de características 1.5.1.

En cuanto a la gestión del proyecto, se adopta una metodología incremental, la cual se describe detalladamente en el punto 2.1.1 y se presenta un plan temporal Planificación temporal que incluye la definición de incrementos específicos y la estimación de recursos necesarios. Finalmente en esta sección se aborda el aspecto económico con un presupuesto detallado .

En la documentación técnica, se realiza un análisis exhaustivo de los requisitos del sistema 3.1, y se mostrará un diagrama de casos de uso, con tablas explicando en detalle las partes más interesantes 3.2.

El diseño del sistema se presenta mediante la arquitectura lógica y física 4.1, proporcionando un marco estructural para el desarrollo del proyecto. Se incluye el diseño detallado de la base de datos 4.2, con un diccionario de datos que especifica las entidades y relaciones clave. Para finalizar la sección se muestran diagramas de secuencia 4.3.

La implementación se detalla con la estructura del código 5.1, la implementación de funcionalidades como la integración de mapas con Leaflet y la importación de archivos KML, destacando los desafíos encontrados y las soluciones aplicadas durante este proceso.

El capítulo de pruebas 6 describe la estrategia utilizada para verificar y validar el sistema implementado, asegurando su funcionalidad y calidad antes de la entrega final. Seguidamente, en el capítulo 7 se detallan las conclusiones sobre el proyecto y posible trabajo a futuro, dando por finalizado el capítulo de documentación.

Los manuales de la aplicación 8 incluyen el uso por parte de los usuarios finales y la administración del sistema, asegurando que todos los usuarios puedan utilizar efectivamente la aplicación según sus necesidades.

1.2. Contexto y justificación

En los últimos años, el uso de plataformas digitales ha transformado la forma en que las personas planean y disfrutan sus vacaciones. El turismo es una de las principales fuentes de ingresos en España, por lo que se ha convertido en un sector clave de la economía del país. En 2019, el país recibió más de 83 millones de turistas extranjeros, lo que generó alrededor del 12 % del PIB y empleó a más de 2,6 millones de personas [6].

En este contexto, las plataformas digitales han revolucionado la forma en que las personas planifican y disfrutan de sus vacaciones. En España, el uso de estas plataformas se ha vuelto cada vez más popular en los últimos años. Este TFG propone una aplicación web que permita gestionar puntos de interés de España (por ahora centrado en Castilla y León) y subir comentarios y fotos sobre ellos.

En resumen, el desarrollo de esta aplicación web busca aprovechar la tecnología y la interacción social para mejorar la experiencia turística en España y fomentar el turismo sostenible y responsable.

El entorno de esta aplicación sería una plataforma web moderna, fácil de usar y accesible desde cualquier dispositivo, basada en una arquitectura cliente-servidor, donde el servidor alojaría la base de datos de puntos de interés, los comentarios y las fotos, y el cliente sería el usuario que accede a la plataforma a través de su navegador web.

1.3. Motivación

El sector turístico español enfrenta muchos desafíos, incluyendo la competencia con otros destinos, la necesidad de diversificar la oferta turística y la presión sobre el patrimonio natural y cultural del país. Estos desafíos son cada vez más complicados de abordar por la rápida evolución tecnológica y las cambiantes preferencias de los turistas modernos, que buscan experiencias personalizadas y enriquecedoras.

La promoción de los atractivos turísticos de España es crucial para mantener y aumentar su competitividad en el mercado global. Una página web que permita a los usuarios conocer los diferentes lugares de interés y compartir sus experiencias podría ser una excelente manera de fomentar el turismo en el país. Esta plataforma no solo ayudaría a distribuir el flujo turístico hacia destinos menos conocidos, aliviando la presión sobre los lugares más visitados, sino que también promovería un turismo más sostenible y diversificado.

Por tanto, la motivación principal detrás de este proyecto es contribuir al desarrollo socioeconómico de las regiones visitadas, promover un turismo sostenible y proporcionar a los turistas una herramienta útil y moderna para gestionar todos aquellos lugares de interés que consideren relevantes y descubrir nuevos para poder disfrutar y viajar a ellos.

1.4. Objetivos del trabajo

Este TFG constituye un proyecto personal altamente gratificante para mí. El desarrollo de una plataforma web requiere el aprendizaje y la mejora de habilidades en áreas como programación web, diseño de interfaces de usuario, gestión de bases de datos y seguridad web. Aprender y mejorar en habilidades técnicas me generaría gran satisfacción personal.

En cuanto a los objetivos teóricos y prácticos, esta plataforma tiene como objetivo principal ayudar a los turistas a descubrir nuevos lugares de interés en España, incluyendo lugares menos conocidos o fuera de las rutas turísticas tradicionales, contribuyendo al desarrollo socioeconómico de las regiones visitadas. Además, permite a los usuarios compartir sus experiencias y recomendaciones sobre los lugares que han visitado, lo que puede ser muy útil para otros viajeros.

Desglosando este objetivo general, podemos identificar los siguientes objetivos específicos:

- **Obj-1: Desarrollo de una plataforma web para la gestión de puntos de interés.**
- **Obj-2: Implementar un sistema robusto y seguro de gestión de usuarios.**
- **Obj-3: Implementación de funcionalidades de interacción entre usuarios y puntos de interés.**
- **Obj-4: Facilitar la colaboración entre usuarios a través de comentarios e imágenes.**
- **Obj-5: Incorporar funcionalidades para la creación, importación y exportación de puntos de interés, utilizando formatos estándar como KML.**

1.5. Alcance

El alcance de la aplicación incluye el desarrollo de una plataforma web con diversas funcionalidades destinadas a mejorar la experiencia del usuario al gestionar y explorar puntos de interés en España. A continuación, se presenta un árbol de características que detalla las funcionalidades principales de la aplicación. Cada rama del árbol se describe brevemente.

1.5.1. Árbol de características

El árbol de características es una representación visual de las funcionalidades principales que tendrá la aplicación. Cada rama funcional se describe a continuación:

- **Gestión de usuarios:** Esta funcionalidad permite a los usuarios registrarse, iniciar sesión y gestionar sus cuentas de forma segura. Incluye la capacidad de modificar y eliminar cuentas.
- **Gestión de puntos de interés:** Los usuarios podrán crear, ver, editar y eliminar sus propios puntos de interés. Además, se les permitirá explorar puntos de interés creados por otros usuarios.
- **Interacción con puntos de interés:** Los usuarios tendrán la opción de comentar sobre los puntos de interés de otros, subir imágenes y guardar sus puntos de interés favoritos en su perfil personal.
- **Creación, importación y exportación de puntos de interés:** Esta funcionalidad incluye la creación manual de puntos de interés, así como la capacidad de importar y exportar puntos de interés en formato KML.

El árbol de características se detallará en el siguiente diagrama, el cual ilustra visualmente las ramas mencionadas.

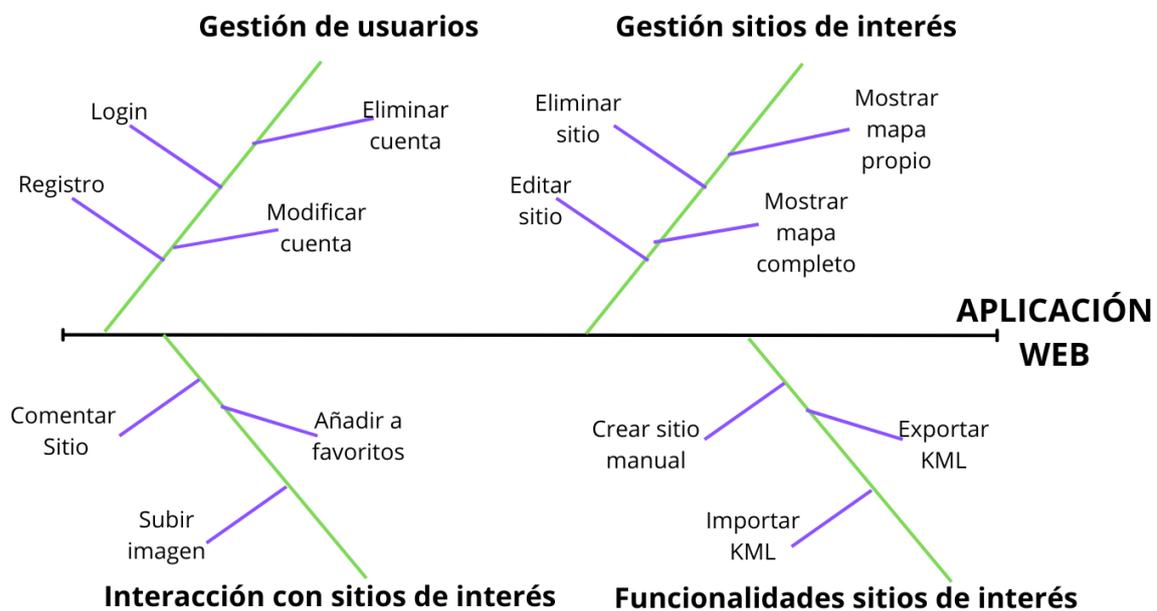


Figura 1.1: Árbol de características.

1.5.2. Restricciones y limitaciones

Existen limitaciones en cuanto a la cantidad de información disponible para la base de datos y la veracidad de la información subida por los usuarios. Además, se deben considerar restricciones en cuanto a la disponibilidad de recursos y tiempo para la implementación y desarrollo de la plataforma web, en el caso de este proyecto, al ser la asignatura del TFG, la carga de trabajo corresponde a 12 Sistema Europeo de Transferencia de Créditos o *European Credit Transfer System* (ECTS), entre 300 y 360 horas.

1.6. Conceptos clave y soporte tecnológico

Para poder desarrollar una aplicación robusta y eficiente, es necesario definir claramente los términos y conceptos con los que se va a trabajar. Estos conceptos son fundamentales no solo para el diseño y la implementación del sistema, sino también para asegurar que todos los involucrados en el proyecto tengan una comprensión común de los elementos clave. A continuación, se definen los términos de punto de interés, ruta y georreferenciación, y se explora el soporte tecnológico actual que permite la implementación de estas funcionalidades en aplicaciones modernas.

1.6.1. Puntos de interés

Un punto de interés (PoI, *Point of Interest*) se refiere a una ubicación específica que puede resultar atractiva o útil para los turistas [28]. Estos pueden incluir monumentos históricos, museos, parques naturales, restaurantes, entre otros. La plataforma permitirá a los usuarios descubrir y compartir información sobre estos puntos, enriqueciendo así la oferta turística disponible.

1.6.2. Rutas

Una ruta se define como una secuencia de puntos de interés, con un origen y un destino final, que los turistas pueden seguir para maximizar su experiencia de visita [11]. Las rutas pueden ser temáticas (por ejemplo, rutas gastronómicas o culturales) y ayudan a los turistas a planificar su itinerario de manera eficiente. La aplicación incluirá la funcionalidad de crear y compartir rutas personalizadas.

1.6.3. Georreferenciación

La georreferenciación es el proceso de asignar coordenadas geográficas (latitud y longitud) a puntos de interés para ubicarlos en un mapa. Este proceso es fundamental para el funcionamiento de la plataforma, ya que permite a los usuarios visualizar los puntos de interés y las rutas en un mapa interactivo. La aplicación utilizará tecnologías de georreferenciación modernas, como el uso de APIs de mapas, para proporcionar una experiencia de usuario enriquecida. [14, 27]

1.6.4. Soporte tecnológico actual

El soporte tecnológico actual para la implementación de funcionalidades como puntos de interés, rutas y georreferenciación ha avanzado significativamente. En el ámbito del turismo digital, muchas plataformas utilizan sistemas de información geográfica (SIG) y tecnologías de geolocalización para mejorar la experiencia del usuario.

Las APIs de mapas, como Google Maps y Leaflet, permiten integrar mapas interactivos en aplicaciones web, proporcionando herramientas para la georreferenciación y visualización de datos geoespaciales [9, 13]. Estas tecnologías facilitan la creación de aplicaciones que pueden mostrar puntos de interés en un mapa, calcular rutas optimizadas y ofrecer navegación en tiempo real.

Además, el uso de bases de datos geoespaciales, como PostgreSQL con la extensión PostGIS, permite almacenar y gestionar grandes volúmenes de datos geográficos de manera eficiente. Estas bases de datos soportan operaciones complejas de análisis espacial, lo que es crucial para aplicaciones que manejan datos de ubicación [17].

Plataformas como TripAdvisor y Yelp también utilizan estos principios para permitir a los usuarios encontrar y revisar puntos de interés. Estas plataformas incorporan sistemas de recomendación y personalización basados en el historial de usuario y las preferencias, lo que mejora la relevancia de las sugerencias de puntos de interés [24, 30].

En resumen, el soporte tecnológico actual ofrece un amplio conjunto de herramientas y metodologías que permiten desarrollar aplicaciones avanzadas y eficientes para la gestión de puntos de interés, rutas y datos georreferenciados.

1.7. Entorno de aplicación

En la primera sección 1.7.1 se aborda el contexto tecnológico y metodológico necesario para el desarrollo de la aplicación propuesta. El **estado del arte** se refiere al panorama actual y evolución de las tecnologías, métodos y prácticas relevantes en el campo específico de estudio. En el contexto de este trabajo, el estado del arte proporciona una visión general de las plataformas digitales utilizadas en el sector turístico, destacando las tendencias, innovaciones y mejores prácticas.

Además, se explorarán las **tecnologías utilizadas** 1.7.2, que comprenden los lenguajes de programación, *frameworks*, herramientas de gestión de bases de datos y otras tecnologías específicas empleadas en el desarrollo de la aplicación. Esta sección proporcionará un marco de referencia crucial para entender las decisiones y enfoques adoptados en la implementación del proyecto.

1.7.1. Estado del arte

Propuesta 1: Portal oficial de turismo en España

Se trata de un portal dedicado a la promoción del turismo natural en España [5]. Su contenido se centra en la descripción y presentación de los diferentes destinos naturales del país, como parques nacionales, reservas naturales, montañas, playas, y otros lugares de interés natural.

La página web está dividida en varias secciones, las cuales se corresponden con:

- **Menú superior**, consta de las opciones “Dónde ir”, “Qué hacer” y “Planifica tu viaje”.
- **Sección “Dónde ir”**, la cual presenta los parques nacionales de Europa.
- **Sección “Qué hacer”**, con paseos y rutas naturales, entre otros.
- **Sección “Buscador”**, en caso de querer buscar información concreta sobre un lugar.
- **Sección “Te puede interesar”** y **“Otras propuestas”**, las cuales se centran sobretodo en el turismo sostenible.
- **Sección “Galería de imágenes”**, con fotos de los principales puntos de interés.

En el menú superior la opción **“Dónde ir”** se centra en ciudades españolas, **“Qué hacer”** contempla actividades como gastronomía, deporte y rutas naturales y, por último, **“Planifica tu viaje”** te muestra opciones como el tiempo, descarga de folletos, reserva de experiencias. En la página principal, se encuentran imágenes y vídeos de los destinos naturales más destacados, así como enlaces a mapas interactivos o secciones temáticas como **“Playas y calas”**, **“Senderismo y montañismo”**, entre otras.

En las secciones temáticas, se encuentran descripciones detalladas de cada destino, con información sobre su ubicación, características naturales, atractivos turísticos, servicios y actividades disponibles. También se ofrecen recomendaciones y consejos prácticos para los visitantes, como recomendaciones de hoteles y restaurantes cercanos, así como consejos sobre cómo disfrutar de la naturaleza de forma sostenible.

Ventajas:

- **Fácil navegación:** La página web está diseñada de forma clara y sencilla, lo que facilita la navegación y la búsqueda de información.
- **Información detallada:** La plataforma ofrece una gran cantidad de información detallada y actualizada sobre los diferentes destinos naturales de España, además, incluye recomendaciones y consejos prácticos para los visitantes, lo que puede ayudar a mejorar su experiencia de viaje.
- **Idiomas:** La página web está disponible en varios idiomas, lo que puede resultar útil para los visitantes extranjeros.

Desventajas:

- **Sobrecarga de información:** La página web ofrece una gran cantidad de información, lo que puede resultar en un efecto contrario al inicial de ayudar a los turistas, convirtiéndose en una experiencia abrumadora para aquellos usuarios que buscan una experiencia más simplificada.
- **Poca exactitud:** Aunque la página web está disponible en varios idiomas, puede que no sea tan completa o detallada en algunos idiomas como en otros.
- **Limitada:** La página web se enfoca principalmente en destinos naturales en España, por lo que puede no ser útil para aquellos que buscan información sobre destinos naturales en otros países.

Propuesta 2: Wikiloc

Wikiloc es una plataforma web que se dedica a difundir y compartir rutas de senderismo y actividades al aire libre [22]. Esta aplicación cuenta con una base de datos con una gran cantidad de rutas en diferentes regiones del mundo, las cuales pueden ser visualizadas en mapas y descargadas por los usuarios. Además, Wikiloc permite la subida de nuevas rutas por parte de los usuarios, los cuales pueden añadir comentarios, fotos y detalles sobre la ruta.

La página web se encuentra organizada en diferentes secciones, las cuales son:

- **Menú superior**, consta de todas las rutas divididas por tipo de actividad.
- **Sección “Explorar”**, En este apartado se pueden buscar rutas por ubicación geográfica, tipo de actividad, dificultad, distancia, duración, entre otros filtros. También se pueden ver las rutas cercanas a la ubicación actual del usuario.
- **Sección “Rutas por país”**, en esta opción se encuentran los mapas interactivos con las rutas disponibles en cada zona geográfica. Los usuarios pueden ver las rutas sobre el mapa, visualizar los detalles de cada una y descargarlas.
- **Sección “Subir ruta”**, permite a los usuarios subir sus propias rutas y compartir sus experiencias. Se pueden añadir detalles sobre la ruta, fotos, comentarios y valoraciones.
- **Sección “Comunidad”**, en este apartado se pueden ver las rutas compartidas por otros usuarios, seguir a otros usuarios, comentar y valorar rutas, entre otras funcionalidades sociales.
- **Sección “Wikiloc Planet”**, sección con artículos, entrevistas y consejos sobre viajes.

Una vez que se ha seleccionado una categoría en la plataforma, se redirige al usuario a una página que presenta una lista de las distintas rutas que se corresponden con el filtro seleccionado. Las publicaciones de las rutas, desde el punto de vista del usuario, ofrecen información como fotografías, opiniones subjetivas, la extensión en kilómetros y la altitud que implica la ruta. Al elegir una ruta específica, se puede obtener información más detallada, incluyendo la opción de visualizarla en el mapa, descargarla o enviarla a un dispositivo GPS compatible.

Ventajas:

- **Gran cantidad de rutas:** La plataforma cuenta con una extensa base de datos de rutas de senderismo, ciclismo, *running*, esquí de montaña, entre otras, lo que la convierte en una herramienta muy completa para los usuarios.

- **Fácil navegación:** La página web y la aplicación móvil de Wikiloc son muy intuitivas y fáciles de usar, lo que facilita la búsqueda y selección de rutas.
- **Información detallada:** Cada ruta cuenta con información sobre la dificultad, duración, desnivel, comentarios y fotografías, lo que permite al usuario conocer mejor la ruta antes de realizarla.
- **Comunidad de usuarios:** Wikiloc cuenta con una gran comunidad de usuarios que comparten sus rutas y experiencias, lo que permite a los usuarios interactuar y compartir información sobre rutas.
- **Funciones avanzadas:** La plataforma ofrece funciones avanzadas como la posibilidad de descarga de rutas para GPS, seguimiento en tiempo real de la ruta, y estadísticas de la actividad realizada.

Desventajas:

- **Fiabilidad de la información:** Debido a que Wikiloc se basa en la contribución de usuarios, la información puede ser inexacta, incompleta o desactualizada. Por lo tanto, es importante verificar la información de la ruta antes de seguir cualquier ruta. También puede haber gran cantidad de opiniones poco útiles o fotos con baja calidad.
- **Sobrecarga de información:** Wikiloc cuenta con una gran cantidad de rutas, lo que puede dificultar la búsqueda de la ruta adecuada. Es posible que los usuarios tengan que pasar mucho tiempo buscando una ruta que se ajuste a sus necesidades.
- **No todas las funciones son gratuitas:** Algunas funciones de Wikiloc, como descargar mapas o acceder a ciertas rutas *premium*, pueden requerir una suscripción *premium* o un pago adicional.

Propuesta 3: Outdooractive

Se trata de una de las plataformas *outdoor* más grandes de Europa, tanto en formato web como en aplicación [7]. Su enfoque principal es proporcionar información detallada y recursos para actividades como senderismo, ciclismo, esquí, montañismo o *trail running*. Permite a los usuarios buscar y descubrir una amplia variedad de rutas y destinos al aire libre en diferentes ubicaciones geográficas, tanto en España como en otros países. Para ello posee mapas basados en datos oficiales y una tecnología propia de enrutamiento para la planificación de rutas.

La estructura de la página web está segmentada en múltiples secciones, cada una de las cuales se corresponde con un resumen de las distintas páginas que componen el menú superior:

- **Menú superior**, se divide en mapa, buscador de rutas, planificador de rutas, guía de viajes y comunidad.
- **Sección “Mapa”**, En este apartado se proporciona un mapa propio y completamente personalizable para la búsqueda de actividades, ya que permite, por ejemplo, poner la vista en las distintas estaciones del año y el tipo de ruta que prefieras.
- **Sección “Buscador de rutas”**, En este apartado se pueden buscar rutas por ubicación geográfica, tipo de actividad, dificultad, estación del año.
- **Sección “Planificador de ruta”**, sección que permite crear un mapa y marcar los puntos por los que quieres pasar. Una vez acabado, tienes la opción de descargarla para poder acceder a ella sin conexión.
- **Sección “Guía de viajes”**, en este apartado se encuentra información referente a actividades ya planeadas u organizadas por país. Añade además una sección de retos en los que puedes inscribirte y ganar insignias para tu perfil.
- **Sección “Comunidad”**, permite intercambiar informaciones con personas con intereses comunes a la vez que puede ser usada como fuente de inspiración.

Ventajas:

- **Amplia variedad de actividades al aire libre:** La página web ofrece una amplia gama de actividades al aire libre, lo que la convierte en una plataforma versátil para aquellos que buscan explorar diferentes opciones de turismo activo.
- **Información detallada y precisa:** Outdooractive proporciona descripciones detalladas, perfiles de elevación y puntos de interés para las rutas, lo que permite a los usuarios planificar y prepararse de manera efectiva para sus actividades al aire libre.

- **Interfaz intuitiva y fácil de usar:** La página web tiene una interfaz amigable, lo que facilita la navegación a la vez que la búsqueda y exploración de rutas y destinos de interés.
- **Mapas interactivos:** Los mapas interactivos permiten a los usuarios visualizar las rutas y puntos de interés en detalle, lo que ayuda en la orientación y planificación durante las actividades al aire libre.
- **Idiomas:** La página web está disponible en varios idiomas, lo que puede resultar útil para los visitantes extranjeros.

Desventajas:

- **Limitaciones en la cobertura geográfica:** Aunque la página web ofrece información y rutas en varios países, puede haber áreas geográficas con menos contenido o cobertura limitada, lo que puede ser una desventaja para aquellos que buscan explorar áreas menos conocidas o remotas.
- **Dependencia de la conexión a internet:** Para aprovechar al máximo la página web y acceder a toda la información y funcionalidades, es necesario tener una conexión estable a internet. Esto puede ser un inconveniente en áreas sin conexión o con una señal débil.
- **Pro y Pro+:** : Ciertas características como poder acceder *offline* a los mapas o el acceso planificación en 3D de las rutas exigen una suscripción de pago mensual.

Tabla comparativa

Opción	Portal Turismo	Wikiloc	Outdooractive
Enfoque principal	T	S, C, T	S, C, T
Cobertura geográfica	N	I	I
Funcionalidades <i>premium</i>	×	✓	✓
Amplio catalogo de actividades	×	✓	✓
Interfaz de usuario	✓	✓	✓
Hacer rutas con compañía	×	×	✓
Herramientas de navegación	✓	✓	✓
Comunidad de usuarios	×	✓	✓
Aplicaciones móviles	✓	✓	✓
Sin información de pago	✓	×	×
Integración con redes sociales	✓	✓	✓
Varios idiomas	✓	✓	✓
Guías y consejos	×	✓	✓
Fiabilidad de la información	✓	✓	✓
Soporte al cliente	✓	✓	✓

Anexo:

T: Turismo

C: Ciclismo

S: Senderismo

N: Nacional

I: Internacional

1.7.2. Herramientas utilizadas

Frontend

HTML, CSS y JavaScript

Para la estructura y presentación de las páginas web se han utilizado HTML y CSS, respectivamente. HTML (*HyperText Markup Language*) se encargó de definir la estructura y el contenido básico de las páginas, mientras que CSS (*Cascading Style Sheets*) se usó para estilizar y diseñar el contenido, proporcionando una apariencia visual atractiva y coherente. JavaScript, por otro lado, se empleó para agregar interactividad y funcionalidades dinámicas, permitiendo que las páginas respondan a las acciones del usuario en tiempo real.

Las principales ventajas de utilizar HTML y CSS son que son estándares ampliamente soportados por todos los navegadores web, lo que asegura una apariencia y funcionalidad consistentes en diferentes plataformas y dispositivos. Además, son lenguajes fáciles de aprender y usar, lo que permite un desarrollo eficiente y rápido. Sin embargo, una desventaja es que pueden existir pequeñas diferencias en la forma en que distintos navegadores interpretan y renderizan el código, lo que puede requerir ajustes específicos y pruebas exhaustivas.

En el caso de JavaScript, su principal ventaja es que permite una experiencia de usuario interactiva y dinámica, haciendo posible la implementación de funcionalidades avanzadas como validación de formularios y actualizaciones de contenido sin necesidad de recargar la página. Además, cuenta con un amplio ecosistema de bibliotecas y *frameworks* que facilitan el desarrollo de aplicaciones complejas. No obstante, una desventaja es que en aplicaciones grandes, el código JavaScript puede volverse complejo y difícil de mantener sin un marco adecuado, lo que puede llevar a problemas de mantenimiento y potenciales vulnerabilidades de seguridad.



Figura 1.2: Imagen de marca de HTML5, CSS y JavaScript.

HTML5 (*HyperText Markup Language 5*)

Se trata del lenguaje estándar utilizado para crear y estructurar el contenido de las páginas web. Define la estructura de una página web utilizando una variedad de etiquetas semánticas que representan diferentes tipos de contenido (encabezados, párrafos, listas, etc.).

- Características:
 1. Introducción de elementos semánticos para contener y definir la información, como con `<head>`, `<title>` y `<body>`.
 2. Soporte nativo para audio y video.
 3. Funciones para almacenamiento local y geolocalización.

CSS (*Cascading Style Sheets*)

CSS es un lenguaje de estilo utilizado para definir el aspecto visual y el diseño de una página web. Funciona en conjunto con HTML mediante reglas de estilo que especifican propiedades como colores, fuentes, márgenes, tamaños, etc.

- Características:
 1. Separación entre contenido y presentación.
 2. Capacidad para aplicar estilos a través de selectores que apuntan a elementos individuales o grupos de elementos (en este caso referidos a los elementos creados en el HTML).

JavaScript

Es un lenguaje de programación de alto nivel utilizado para agregar interactividad y dinamismo a las páginas web. Se ejecuta en el navegador del usuario y permite responder a la interacción de los usuarios y cambiar la distribución del contenido en la página web.

- Características:
 1. Orientado a objetos y basado en prototipos.
 2. Implementa funciones de primera clase y funciones anónimas.
 3. Amplia variedad de bibliotecas y *frameworks* disponibles (como jQuery, React.js, Angular, Vue.js).
 4. Posibilidad de desarrollo tanto en el lado del cliente como en el lado del servidor (Node.js).

Leaflet

Es una biblioteca de JavaScript utilizada para crear mapas interactivos. Está basado en el navegador y diseñado con simplicidad pero con todas las características necesarias para los desarrolladores, ya que la puedes ampliar con muchos complementos al tener una API de fácil uso.

Las principales ventajas de utilizar Leaflet son su ligereza y facilidad de uso, respaldadas por una buena documentación. Además, permite integrar capas y marcadores personalizados, lo que ofrece flexibilidad en la visualización de datos geoespaciales.

Por otro lado, una de las desventajas es que Leaflet tiene menos funcionalidades en comparación con otras bibliotecas más robustas como Google Maps API. Esto puede ser una limitación para proyectos que requieren características avanzadas de mapeo.



Figura 1.3: Imagen de marca de Leaflet.

- Características:

1. Soporte para marcadores, capas, *popups*, y vectores.
2. Control de eventos interactivos.
3. Compatibilidad con múltiples proveedores de mapas base.
4. Soporte para geoJSON.
5. Posibilidad de agregar *plugins* para funcionalidades adicionales.

Backend

PHP (*Hypertext Preprocessor*)

PHP es un lenguaje de programación del lado servidor, ampliamente utilizado para el desarrollo web. Se utiliza para generar contenido dinámico, procesar formularios, acceder a bases de datos y gestionar sesiones de usuario.

Las principales ventajas de utilizar PHP son su facilidad de aprendizaje y su amplio soporte en la comunidad de desarrollo web. Esto hace que sea accesible para nuevos desarrolladores y que haya abundantes recursos disponibles para aprender y resolver problemas. Además, PHP se integra muy bien con bases de datos MySQL, lo que facilita la creación de aplicaciones web dinámicas y basadas en bases de datos.

Sin embargo, PHP también presenta algunas desventajas. Puede tener problemas de rendimiento en aplicaciones muy grandes, especialmente si no se optimiza adecuadamente. Además, es crucial tener cuidado con la seguridad, ya que PHP es susceptible a vulnerabilidades comunes como la inyección SQL si no se implementan medidas de seguridad adecuadas.



Figura 1.4: Imagen de marca de PHP.

■ Características:

1. Integración con una amplia variedad de bases de datos, siendo MySQL la opción más común.
2. Soporte para programación orientada a objetos y procedimental.
3. Facilidad de integración con HTML y otras tecnologías web.
4. Versatilidad para desarrollar aplicaciones web complejas o simples sitios web.

Base de Datos

MySQL

MySQL es un sistema de gestión de bases de datos relacional de código abierto. Se utiliza para almacenar, organizar y gestionar datos de forma eficiente y segura en el servidor.

Entre las principales ventajas de utilizar MySQL está su amplia documentación y comunidad de soporte, lo que facilita la resolución de problemas y el acceso a recursos educativos. Además, MySQL es altamente compatible con PHP y otras tecnologías web, lo que simplifica el desarrollo de aplicaciones web dinámicas.

Las desventajas vienen por ser menos escalable que otras opciones como PostgreSQL cuando se trata de manejar grandes volúmenes de datos, lo que puede limitar su uso

en aplicaciones de gran escala. Además, MySQL requiere una configuración y gestión adecuadas para evitar problemas de rendimiento, lo que puede añadir complejidad al mantenimiento del sistema.



Figura 1.5: Imagen de marca de MySQL.

■ Características:

1. Soporte para tablas, índices, vistas, procedimientos almacenados y disparadores.
2. Lenguaje SQL estándar para realizar consultas y manipular datos.
3. Implementa transacciones para garantizar la integridad de la base de datos en caso de fallos de energía u otros inconvenientes.
4. Seguridad robusta con opciones de autenticación y privilegios de usuario.

Sistema Operativo

Ubuntu

Ubuntu es un sistema operativo basado en Linux utilizado ampliamente para el desarrollo y despliegue de aplicaciones. Proporciona una plataforma estable y segura para ejecutar diversos servicios y aplicaciones web.

Una de las ventajas de Ubuntu es que está disponible de manera gratuita y bajo licencia de código abierto, lo que permite su uso, modificación y distribución sin costos adicionales. Cuenta con una gran comunidad de usuarios y desarrolladores que proporcionan soporte técnico, y es conocido por su alta seguridad y estabilidad, lo cual es crucial para entornos de desarrollo y producción.

Los inconvenientes residen en la curva de aprendizaje, ya que puede requerir cierta familiarización para usuarios acostumbrados a otros sistemas operativos como Windows, especialmente en términos de la línea de comandos y configuración avanzada. Otro inconveniente sería que algunas aplicaciones comerciales pueden no tener soporte nativo para Ubuntu, lo que puede requerir soluciones alternativas o emulación.



Figura 1.6: Imagen de marca de Ubuntu.

■ Características:

1. Utiliza el núcleo Linux como base, proporcionando estabilidad y seguridad.
2. Existencias varias versiones adaptadas para servidores, estaciones de trabajo y otros entornos.
3. Ofrece actualizaciones regulares de seguridad y soporte a largo plazo para versiones específicas.

Apache

Apache es un servidor web de código abierto ampliamente utilizado en el desarrollo y despliegue de sitios web. Actúa como un intermediario entre el cliente y el servidor mediante el navegador, sirviendo todos los ficheros necesarios para visualizar la web.

Apache presenta varias ventajas significativas como servidor web, entre las cuales destacan su robusta modularidad y extensibilidad mediante módulos adicionales, lo cual permite ampliar y personalizar su funcionalidad según las necesidades específicas de cada proyecto. Además, ofrece una configuración flexible que se adapta fácilmente a diferentes requisitos de alojamiento web, asegurando una gestión eficiente y personalizada del servidor.

No obstante, Apache también presenta algunas desventajas. Aunque es altamente configurable, esta flexibilidad puede llevar a una configuración compleja que requiere conocimientos técnicos profundos para optimizar el rendimiento y la seguridad. Además, en comparación con servidores web más ligeros como Nginx, Apache puede mostrar un rendimiento ligeramente inferior en términos de manejo de conexiones concurrentes y uso eficiente de recursos del sistema.

- Características:
 1. Modularidad y extensibilidad mediante módulos adicionales.
 2. Configuración flexible para adaptarse a diferentes requisitos de alojamiento web.
 3. Soporte para SSL/TLS para conexiones seguras.
 4. Control de acceso y autenticación mediante archivos de configuración.
 5. Monitoreo y registro de actividad para diagnóstico y análisis.

Formatos y Archivos Específicos

Archivos KML

KML es un formato de archivo XML utilizado para representar datos geoespaciales. Permite especificar puntos, líneas, polígonos, imágenes superpuestas, estilos de visualización, etc., para personalizar la visualización de mapas.

Sus ventajas principales son su fácil integración con herramientas como Google Earth y Leaflet, y que es ampliamente utilizado en aplicaciones de mapas, por lo que tiene una amplia documentación. En cuanto a inconvenientes, la elaboración de estos archivos es laboriosa y manual, y la cantidad de datos a manejar está limitada.

- Características:
 1. Soporta la representación de datos geográficos en 2D y 3D.
 2. Facilita la creación de mapas interactivos con información descriptiva y enriquecida.
 3. Compatible con herramientas de visualización como Google Earth.
 4. Integración con servicios de mapas y APIs de geolocalización.
 5. Extensibilidad mediante el uso de estilos y etiquetas personalizadas.

Capítulo 2

Gestión del Proyecto

El proyecto se gestionó bajo una metodología incremental, permitiendo un desarrollo iterativo y adaptable a los cambios. Esta metodología 2.1.1 facilitó un proceso estructurado desde la planificación hasta la implementación final del sistema.

La planificación temporal 2.2 se dividió en tres incrementos (Sección 2.2.2 a 2.2.4), enfocándose en la planificación y diseño inicial, la implementación de funcionalidades básicas, y la finalización con pruebas exhaustivas para asegurar la calidad del producto.

En términos económicos 2.3, se detalló un presupuesto inicial que más tarde se comparará con el coste real.

2.1. Metodología

2.1.1. Proceso de desarrollo

El proceso de desarrollo de este proyecto se basa en la metodología incremental. Esta metodología es un proceso de desarrollo de software en el que los requisitos se dividen en múltiples módulos independientes del ciclo de desarrollo de software. Es la base de otros métodos de desarrollo como RUP y los métodos ágiles.

El desarrollo incremental se realiza en pasos encadenados desde el diseño del análisis, la implementación, las pruebas/verificación y el mantenimiento. Esto se repite el número de veces necesario hasta finalizar el proyecto. Mientras que en las primeras iteraciones los requisitos y el análisis serán más relevantes, a medida que avance el proyecto ganará importancia el desarrollo y la implementación.

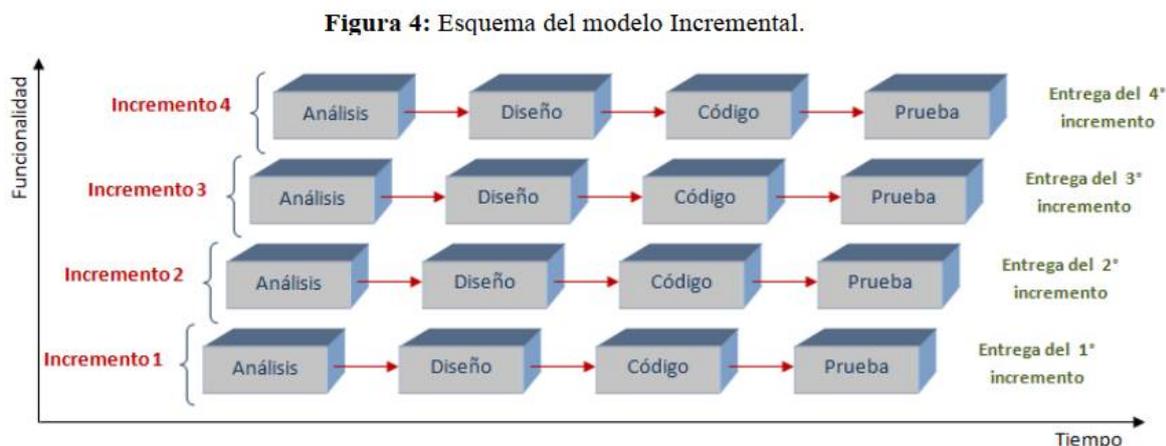


Figura 2.1: Modelo de desarrollo incremental

2.1.2. Características generales de la metodología incremental

La metodología incremental ofrece numerosas ventajas que la hacen adecuada para el desarrollo de proyectos de software. Algunas de estas ventajas son:

- **Entrega gradual de funcionalidades:** desde el inicio del desarrollo permite a los usuarios utilizar una versión parcial del sistema, lo que facilita la retroalimentación temprana y la adaptación a las necesidades del usuario.
- **Flexibilidad para el cambio:** La metodología incremental facilita la incorporación de cambios en los requisitos en etapas posteriores del desarrollo. Esto es especialmente útil en proyectos donde los requisitos pueden variar o desarrollarse con el tiempo.
- **Reducción de riesgos:** Esta metodología facilita identificar y mitigar riesgos de manera temprana y continua, lo que aumenta la probabilidad de éxito del proyecto.
- **Mejora continua:** Cada incremento es una oportunidad para revisar y mejorar el trabajo realizado. Esto permite la optimización continua del software y la corrección temprana de errores.

Además de estas ventajas, la metodología incremental se caracteriza por:

- **Los incrementos son pequeños:** Esto permite una fácil administración de las tareas en cada iteración y facilita la planificación y el seguimiento del proyecto.
- **La inversión se materializa a corto plazo:** Al entregar funcionalidad de manera gradual y priorizada, la inversión en el proyecto comienza a dar frutos desde las primeras etapas del desarrollo.

- **Es un modelo propicio a cambios o modificaciones:** La flexibilidad inherente a la metodología incremental permite adaptarse a las necesidades cambiantes del proyecto y del usuario.

2.1.3. Justificación de la elección de la metodología incremental

Esta metodología es particularmente adecuada para proyectos donde los requisitos pueden evolucionar con el tiempo y permite la entrega gradual de funcionalidad.

La metodología incremental fue seleccionada por diversas razones. En primer lugar, favorece un desarrollo ágil y eficiente del software. Además, permite probar las funciones de un software de forma interactiva. Esto es especialmente útil en el desarrollo de una página web, donde los requisitos pueden evolucionar con el tiempo y permite la entrega gradual de funcionalidad, por lo que la capacidad de adaptarse a las necesidades cambiantes son cruciales.

Otro aspecto a destacar de esta metodología es que promueve la participación del cliente durante todo el proceso, evitando tener que retroceder en el desarrollo debido a la insatisfacción del cliente.

En cuanto al paradigma de programación, se ha utilizado la **programación orientada a objetos**. Este paradigma es muy adecuado para el desarrollo de software y páginas web debido a su capacidad para modelar el mundo real a través de objetos que encapsulan datos y comportamientos. En este proyecto, se han utilizado lenguajes de programación como PHP 1.7.2 y JavaScript 1.7.2, que soportan la programación orientada a objetos, lo que ha permitido un desarrollo más eficiente y una mejor organización del código.

Además, aunque se ha utilizado una metodología incremental para el desarrollo de este proyecto, esto no excluye el uso de prácticas ágiles. Las metodologías ágiles son muy populares en el desarrollo de software y permiten un desarrollo rápido y eficiente, adaptándose a los cambios y necesidades del proyecto. Aunque la metodología ágil y la metodología incremental pueden parecer diferentes, en realidad pueden complementarse entre sí. La metodología ágil se centra en la entrega continua de valor al cliente, mientras que la metodología incremental se centra en la construcción y entrega de características en incrementos. Juntas, estas dos metodologías pueden proporcionar un marco eficaz para el desarrollo de software.

En resumen, se ha optado por la metodología incremental en este proyecto debido a su adaptabilidad y su capacidad para entregar funcionalidad de manera gradual y priorizada, lo que la convierte en una opción robusta para el desarrollo de software moderno y páginas web dinámicas.

2.1.4. Comparación con otras metodologías

La metodología incremental se diferencia significativamente de otras metodologías tradicionales, como la programación estructurada, en varios aspectos clave.

En la programación estructurada, el enfoque se centra principalmente en las acciones y funciones como las unidades fundamentales de programación. Este paradigma, a menudo caracterizado por un flujo de control basado en la secuencia, selección y repetición, resulta muy adecuado para problemas bien definidos y algoritmos con un flujo lógico claro y lineal. Sin embargo, esta metodología puede resultar limitada cuando se trata de abordar proyectos complejos y cambiantes, ya que su rigidez puede dificultar la adaptación a nuevos requisitos o cambios en el proyecto.

Por otro lado, la metodología incremental se centra en la interacción entre objetos, lo que permite descomponer el sistema en módulos funcionales independientes que pueden ser desarrollados y mejorados de forma iterativa. Cada incremento agrega funcionalidad al sistema, permitiendo que las partes del software se prueben y validen a medida que se desarrollan. Este enfoque no solo facilita la gestión de la complejidad del software, sino que también mejora la capacidad de respuesta a los cambios en los requisitos del cliente.

En comparación con otras metodologías ágiles, como Scrum o Kanban, la metodología incremental también enfatiza la entrega continua de valor al cliente, aunque con menos estructura formal que Scrum o menos enfoque en la visualización del flujo de trabajo que Kanban. No obstante, todas estas metodologías comparten principios comunes de flexibilidad, adaptabilidad y enfoque en la colaboración del equipo y la satisfacción del cliente.

2.2. Planificación temporal

En este capítulo describiré cómo se ha estructurado el trabajo en incrementos basados en el modelo de la metodología incremental 2.1.1, se detallará cómo se aplicó esta metodología en la planificación temporal, incluyendo la estructura de los sprints y otros aspectos relevantes. Finalmente, en la sección 2.3, se presentará el presupuesto económico del proyecto.

2.2.1. Definición de los incrementos

Para la estructuración del trabajo se ha seguido el modelo de desarrollo incremental. Este enfoque permite dividir el proyecto en varias fases o incrementos, cada uno de los cuales añade una parte funcional del sistema hasta completar el proyecto final. A continuación, se definen formalmente los incrementos utilizados en este proyecto:

Incremento 1: Planificación y diseño

- **Objetivo:** Establecer las bases del proyecto mediante la recopilación de requisitos, análisis del entorno, y definición de la metodología y planificación.
- **Actividades principales:**
 - Análisis detallado de requisitos y definición de objetivos.
 - Investigación del entorno de negocio y tendencias del mercado.
 - Definición de la estructura y organización de la memoria del TFG.
 - Diseño del marco metodológico.
 - Elaboración de un plan de trabajo detallado.
 - Establecimiento de un presupuesto preliminar.
 - Identificación de los entregables principales del proyecto.

Incremento 2: Implementación de funcionalidades básicas

- **Objetivo:** Desarrollar las funcionalidades esenciales del sistema, asegurando una base sólida para las etapas posteriores.
- **Actividades principales:**
 - Desarrollo del sistema de registro y inicio de sesión de usuarios.
 - Diseño y desarrollo de la página principal y la página personal del usuario.
 - Implementación de un buscador de rutas y filtros de búsqueda.
 - Desarrollo de la funcionalidad de subida, procesamiento y exportación de archivos KML.
 - Creación de puntos de interés manuales en el mapa interactivo.
 - Diseño y documentación de diagramas de flujo y navegación.
 - Preparación y configuración del entorno de desarrollo para la base de datos.
 - Realización de pruebas iniciales de las funcionalidades implementadas.

Incremento 3: Finalización y pruebas

- **Objetivo:** Completar el desarrollo del sistema con funcionalidades adicionales, realizar pruebas exhaustivas y preparar la documentación final.
- **Actividades principales:**
 - Implementación de la funcionalidad de modificación de puntos de interés y datos personales.

- Desarrollo de nuevas funcionalidades, como la gestión de favoritos.
- Elaboración de documentación técnica detallada.
- Realización de pruebas exhaustivas de integración y sistema.
- Conclusión del proyecto con la documentación técnica final y preparación de la presentación oral.

La planificación temporal del proyecto se ha estructurado de acuerdo con los incrementos del modelo de desarrollo incremental. A continuación, se detalla el cronograma del proyecto:

2.2.2. Incremento 1 (Planificación y diseño)

1. Descripción del proyecto:

- 1.1. Realizar un análisis detallado de los requisitos del proyecto, incluyendo la definición de los objetivos y alcance del TFG. (Tiempo estimado: 3 días)
- 1.2. Investigar el entorno de negocio relacionado con el turismo en España, incluyendo tendencias actuales, competidores y necesidades del mercado. (Tiempo estimado: 3 días)

2. Metodología:

- 2.1. Definir la estructura y organización de la memoria del TFG, incluyendo los capítulos, secciones y contenido relevante. (Tiempo estimado: 2 días)
- 2.2. Diseñar el marco metodológico del TFG, seleccionando la metodología de desarrollo adecuada y justificando su elección. (Tiempo estimado: 3 días)

3. Planificación:

- 3.1. Elaborar un plan de trabajo detallado, incluyendo la distribución de tareas, asignación de recursos y estimación de tiempos. (Tiempo estimado: 2 días)
- 3.2. Establecer un presupuesto preliminar para el desarrollo del TFG, incluyendo costos de recursos humanos, tecnológicos y otros gastos asociados. (Tiempo estimado: 3 días)
- 3.3. Identificar y definir los entregables principales del TFG, incluyendo la aplicación web, la memoria escrita, la presentación oral, entre otros. (Tiempo estimado: 3 días)

	Tarea	Duracion	Marzo 4, 2024				Marzo 11, 2024				Marzo 18, 2024				Marzo 25, 2024								
			4	5	6	7	8	11	12	13	14	15	18	19	20	21	22	25	26	27	28	29	
Sprint 1	1.1 Análisis detallado de los requisitos del proyecto	Duracion de 3 semanas y 3 días(4/4/2024 - 27/4/2024)	█																				
	1.2 Investigación del entorno turístico en España					█	█																
	2.1 Definición de la estructura y organización de la memoria							█	█														
	2.2 Selección justificada de la metodología de desarrollo									█	█	█											
	3.1 Elaboración de un plan de trabajo detallado.											█	█										
	3.2 Establecimiento de un presupuesto preliminar.													█	█	█							
	3.3 Definición de los entregables principales.																	█	█	█			

Figura 2.2: Incremento 1

2.2.3. Incremento 2 (Implementación de funcionalidades básicas)

1. Desarrollo de funcionalidades:

- 1.1. Implementar el registro de usuarios: Desarrollar un sistema de registro seguro y eficiente para los usuarios. (Tiempo estimado: 3 días)
- 1.2. Desarrollar la funcionalidad de inicio de sesión: Crear un sistema de inicio de sesión que permita a los usuarios acceder a sus cuentas de forma segura. (Tiempo estimado: 2 días)
- 1.3. Crear la página principal: Diseñar y desarrollar una página principal intuitiva y fácil de navegar basada en todos los datos almacenados. (Tiempo estimado: 4 días)
- 1.4. Crear la página personal (mis sitios): Diseñar y desarrollar una página personal parecida a la página principal pero con los datos propios del usuario. (Tiempo estimado: 3 días)
- 1.5. Diseñar y desarrollar un buscador de rutas por ciudad: Implementar una función de búsqueda que permita a los usuarios encontrar rutas por ciudad. (Tiempo estimado: 3 días)
- 1.6. Añadir funcionalidad y filtros al buscador: Poder filtrar por mis sitios o por favoritos. (Tiempo estimado: 2 días)

2. Desarrollo de la funcionalidad de subir KML:

- 2.1. Diseñar e implementar la funcionalidad de subida de archivos KML: Desarrollar una función que permita a los usuarios subir archivos KML de forma segura y eficiente. (Tiempo estimado: 2 días)
- 2.2. Desarrollar la lógica para procesar y almacenar la información de los archivos KML: Crear un sistema que pueda procesar y almacenar la información de los archivos KML de manera efectiva. (Tiempo estimado: 4 días)
- 2.3. Implementar funcionalidad de exportación de datos a archivos KML: Desarrollar la capacidad de exportar datos almacenados en la aplicación a archivos KML, permitiendo a los usuarios descargar información en este formato. (Tiempo estimado: 4 días)

- 2.4. Crear punto de interés manual en mapa interactivo: Desarrollar una interfaz que permita a los usuarios agregar puntos de interés manualmente en el mapa interactivo, proporcionando herramientas intuitivas para ubicar y describir puntos relevantes. (Tiempo estimado: 1 día)

3. Diseño y documentación:

- 3.1. Crear diagramas de flujo: Diseñar diagramas de flujo claros y comprensibles para varias funciones. (Tiempo estimado: 2 días)
- 3.2. Elaborar un diagrama de navegación detallado: Crear un diagrama de navegación que destaque la funcionalidad de subida de KML. (Tiempo estimado: 2 días)
- 3.3. Diseñar el esquema de la base de datos: Diseñar un esquema de base de datos eficiente para manejar la información relacionada con los archivos KML. (Tiempo estimado: 3 días)

4. Preparación para la base de datos:

- 4.1. Configurar el entorno de desarrollo para la base de datos: Preparar el entorno de desarrollo para trabajar con la base de datos. (Tiempo estimado: 2 días)
- 4.2. Crear scripts SQL: Desarrollar scripts SQL para la creación de tablas y definición de relaciones. (Tiempo estimado: 3 días)

5. Pruebas iniciales:

- 5.1. Realizar pruebas iniciales: Ejecutar pruebas en todas las funcionalidades implementadas, con especial atención a la funcionalidad de subida de KML. (Tiempo estimado: 3 días)

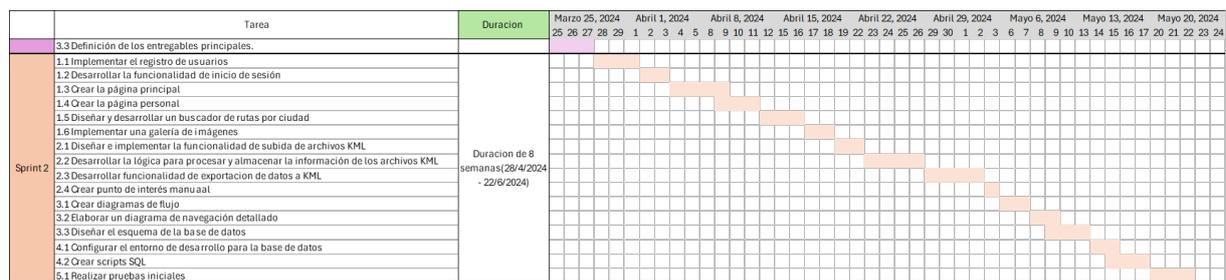


Figura 2.3: Incremento 2

2.2.4. Incremento 3 (Finalización y pruebas)

1. Implementación de funcionalidades restantes:

- 1.1. Implementar la funcionalidad de modificación de puntos de interés, permitiendo a los usuarios editar y actualizar la información relevante de los puntos registrados en la aplicación. (Tiempo estimado: 3 días)
- 1.2. Desarrollar la funcionalidad de modificación de datos personales, asegurando que los usuarios puedan gestionar y actualizar sus perfiles de manera efectiva y segura. (Tiempo estimado: 3 días)

2. Desarrollo de nuevas funcionalidades:

- 2.1. Desarrollar la funcionalidad para añadir a favoritos, facilitando a los usuarios la capacidad de marcar y gestionar sus elementos preferidos dentro de la aplicación. (Tiempo estimado: 3 días)
- 2.2. Desarrollar la funcionalidad de los filtros por “Favoritos” o “Mis sitios” del buscador. (Tiempo estimado: 3 días)

3. Documentación y pruebas:

- 3.1. Elaborar la documentación técnica detallada del desarrollo de las nuevas funcionalidades, incluyendo descripciones de diseño, implementación y pruebas realizadas. (Tiempo estimado: 4 días)
- 3.2. Realizar pruebas exhaustivas de las nuevas funcionalidades para garantizar su funcionamiento correcto y su integración adecuada con el sistema existente. Esto incluye pruebas unitarias, de integración y de sistema. (Tiempo estimado: 6 días)

4. Conclusión y entrega:

- 4.1. Finalizar la documentación técnica final del proyecto, asegurando que esté completa y bien estructurada para su entrega final. (Tiempo estimado: 4 días)
- 4.2. Preparar la presentación oral destacando los aspectos técnicos clave del proyecto, así como los resultados obtenidos durante el desarrollo y las pruebas realizadas. (Tiempo estimado: 3 días)



Figura 2.4: Incremento 3

2.3. Presupuesto económico

La elaboración del presupuesto es esencial para llevar a cabo el proyecto ya que permite tener una idea clara y realista de los recursos necesarios. Además, permite establecer prioridades, organizar y controlar los costos involucrados a la vez que se busca evitar desviaciones que puedan afectar la calidad del trabajo.

Asimismo, es importante destacar que este presupuesto es una estimación basada en los datos y parámetros proporcionados. El coste real puede variar dependiendo de varios factores, como la ubicación geográfica, los precios del mercado y las circunstancias específicas del proyecto. Por lo tanto, es fundamental revisar y ajustar el presupuesto a medida que avanza el proyecto y surgen nuevos requisitos o contingencias.

A continuación, se presenta un análisis paso a paso del presupuesto, considerando los elementos principales que influyen en los costos. Para ello se dividirá entre recursos hardware, recursos software, recursos humanos (RRHH), costos adicionales y contingencias y margen.

2.3.1. Fórmulas de costos

Para determinar el coste real asociado al uso del hardware y del software, es necesario utilizar fórmulas que consideren la frecuencia de uso, el tiempo utilizado y el costo mensual de estas herramientas. Estas fórmulas se detallan a continuación, junto con las unidades correspondientes a cada magnitud entre paréntesis.

Para calcular el costo de uso del hardware y del software, se utilizan las siguientes fórmulas:

$$\text{Costo de uso del hardware} = \frac{\text{Costo inicial del hardware (€)}}{\text{Vida útil del hardware (meses)}} \times \text{Tiempo de uso (meses)} \quad (2.1)$$

$$\text{Costo de uso del software} = \frac{\text{Costo de la licencia (€)}}{\text{Periodo de la licencia (meses)}} \times \text{Tiempo de uso (meses)} \quad (2.2)$$

Para calcular el tiempo de uso requerido para completar el proyecto, se considera una jornada laboral completa de 8 horas diarias. Dado que el trabajo fin de grado (TFG) implica un total de 360 horas, esto resulta en una duración estimada de aproximadamente 1,5 meses. Esta estimación se basa en la suposición de una dedicación constante y sin interrupciones a lo largo del periodo.

2.3.2. Recursos hardware

En cuanto a los recursos hardware, el proyecto se llevará a cabo utilizando varios equipos. Para el cálculo del costo real, se considera un porcentaje de uso total mensual del 33,3% por parte del trabajador. Sin embargo, otros gastos corrientes, como la electricidad para cargar el ordenador, no se incluyen en el cálculo debido a la dificultad de determinar su porcentaje de uso atribuible específicamente al proyecto. En este caso, para calcular los costos se usa la fórmula de “Costo de uso del hardware” 2.1 de la sección anterior.

A continuación, se detallan los costos de uso de cada recurso de hardware:

1. MSI Modern A15 A10RAS Core i5 de 10^a generación:

- Costo inicial: 1.200,00 euros
- Vida útil: 60 meses
- Tiempo de uso para el proyecto: 1,5 meses

$$\frac{1.200,00 \text{ €}}{60 \text{ meses}} \times 1,5 \text{ meses} = 30,00 \text{ €}$$

2. Monitor Samsung gama media:

- Costo inicial: 200,00 euros
- Vida útil: 60 meses
- Tiempo de uso para el proyecto: 1,5 meses

$$\frac{200,00 \text{ €}}{60 \text{ meses}} \times 1,5 \text{ meses} = 5,00 \text{ €}$$

3. Ratón Logitech gama media:

- Costo inicial: 50,00 euros
- Vida útil: 36 meses
- Tiempo de uso para el proyecto: 1.5 meses

$$\frac{50,00 \text{ €}}{36 \text{ meses}} \times 1,5 \text{ meses} = 2,08 \text{ €}$$

2.3.3. Costo total de hardware

El costo total de los recursos de hardware para este proyecto sería la suma de los costos de uso de cada recurso, es decir, 30,00 euros + 5,00 euros + 2,08 euros = 37.08 euros.

Recurso de hardware	Costo de uso (€)
MSI Modern A15 A10RAS Core i5 de 10 ^a generación	30,00
Monitor Samsung gama media	5,00
Ratón Logitech gama media	2,08
Costo total de hardware	37,08

Cuadro 2.1: Costos de uso de los recursos de hardware

2.3.4. Recursos software

En cuanto a los recursos software, se utilizarán varias herramientas para el desarrollo y la gestión del proyecto. Aunque todas estas herramientas son gratuitas, es importante tenerlas en cuenta en el presupuesto para tener una visión completa de los recursos necesarios para el proyecto.

Las herramientas de software que se utilizarán son las siguientes:

- **Visual Studio Code (VS Code):** Es un editor de código fuente desarrollado por Microsoft. Se utilizará para la programación y el desarrollo del proyecto.
- **pgAdmin:** Es una plataforma de administración y desarrollo de código abierto para PostgreSQL. Se utilizará para la gestión de la base de datos del proyecto.
- **Máquina virtual:** Se utilizará una máquina virtual para probar y ejecutar el proyecto en un entorno controlado.

Dado que todas estas herramientas son gratuitas, el costo de uso del software para este proyecto es de 0,00 euros.

2.3.5. Recursos humanos

En cuanto a los recursos humanos, una sola persona asumirá todos los roles necesarios para el proyecto. Estos roles incluyen:

- **Desarrollador web:** Encargado de programar en PHP, HTML y CSS, y trabajar con la base de datos PostgreSQL.
- **Diseñador gráfico:** Responsable de crear la interfaz de la página web y los elementos visuales necesarios.
- **Tester/QA:** Encargado de realizar pruebas de calidad y asegurarse de que la página web funcione correctamente.

Para estimar el costo de los recursos humanos, se considerará un costo por hora o salario promedio. Supongamos un costo por hora de 15,00 euros. Se ha considerado un total de 300 horas en lugar de las 360 horas que típicamente se emplean en un trabajo de fin de grado (TFG). Esta reducción se debe a varios factores clave. En primer lugar, se ha supuesto una dedicación de medio mes exclusivamente a tareas de análisis e investigación, lo que supone aproximadamente 60 horas de trabajo. Este tiempo es crucial para comprender los requisitos del proyecto, realizar la planificación detallada y asegurar una base sólida antes de iniciar las fases de desarrollo y ejecución. Al descontar estas 60 horas del tiempo total, el tiempo efectivo de trabajo para el desarrollo y la implementación del proyecto se ajusta a 300 horas.

Esta estimación más precisa permite una planificación de recursos humanos y económicos más realista y ajustada a las necesidades específicas del proyecto. Por lo tanto, el costo total de los recursos humanos se puede calcular utilizando la siguiente fórmula:

$$\text{Costo total de RRHH (€)} = \text{Tiempo estimado (horas)} \times \text{Costo por hora (€)} \quad (2.3)$$

Aplicando la fórmula, el costo total de los recursos humanos sería:

$$\text{Costo total de RRHH} = 300 \text{ horas} \times 15,00 \text{ €/hora} = 4.500,00 \text{ €}$$

Además, es necesario considerar el coste de la Seguridad Social, que se estima en un 30,90 % del sueldo bruto. Por lo tanto, el coste asociado a la Seguridad Social se calcula de la siguiente manera:

$$\text{Costo de Seguridad Social} = 0,309 \times 4.500,00 \text{ €} = 1.390,50 \text{ €}$$

El costo total de los recursos humanos incluyendo la Seguridad Social sería:

$$\text{Costo total de RRHH (incluyendo Seguridad Social)} = 4.500,00 \text{ €} + 1.390,50 \text{ €} = 5.890,50 \text{ €}$$

2.3.6. Otros costos adicionales

Además de los recursos de hardware, software y humanos, hay otros costos adicionales que deben tenerse en cuenta para el proyecto. Uno de estos costos es la conexión a Internet. Supongamos un costo mensual promedio de 30,00 euros para una conexión de banda ancha.

Considerando que el proyecto tomará 360 horas, y si suponemos que se trabajará 8 horas al día durante 28 días, el costo de la conexión a Internet se puede calcular utilizando la siguiente fórmula:

$$\text{Costo total (€)} = \text{Costo mensual de Internet (€)} \times \text{Meses para el proyecto} \quad (2.4)$$

Aplicando la fórmula, el costo total de la conexión a Internet sería:

$$\text{Costo total de la conexión a Internet} = 30,00 \text{ €} \times 1,5 \text{ meses} = 45,00 \text{ €}$$

2.3.7. Contingencias y margen

Para cubrir posibles imprevistos o contingencias, se agregará un margen del 10 % al costo total del proyecto.

2.3.8. Presupuesto total

Sumando todos los costos, el presupuesto total para el proyecto sería:

Concepto	Costo (€)
Costo total de hardware	37,08
Costo total de software	0,00
Costo total de recursos humanos	5.890,50
Costo total de la conexión a Internet	45,00
Contingencias y margen (10 %)	589,05
Presupuesto total	6.565,63

Cuadro 2.2: Resumen del presupuesto total

2.4. Costo real

En la revisión del presupuesto inicial, es necesario ajustar algunos supuestos clave. Originalmente, se estimó una duración de 1,5 meses considerando una jornada completa de 8 horas al día, lo que daba un total de 360 horas. Sin embargo, se ha observado que este proyecto tomó 4 meses debido a que no se tuvieron en cuenta los fines de semana ni las fiestas, y además, se trabajó a media jornada. Por lo tanto, la dedicación fue de 4 horas al día.

Con estos nuevos parámetros, el cálculo del tiempo de uso del hardware y otros costos adicionales se ajustan para reflejar el periodo real de trabajo.

2.4.1. Recursos hardware revisados

A continuación, se detallan los costos de uso de cada recurso de hardware considerando una duración de 4 meses y media jornada (4 horas al día):

1. MSI Modern A15 A10RAS Core i5 de 10^a generación:

- Costo inicial: 1.200,00 euros
- Vida útil: 60 meses
- Tiempo de uso para el proyecto: 4 meses

$$\frac{1.200,00 \text{ €}}{60 \text{ meses}} \times 4 \text{ meses} = 80,00\text{€}$$

2. Monitor Samsung gama media:

- Costo inicial: 200,00 euros
- Vida útil: 60 meses
- Tiempo de uso para el proyecto: 4 meses

$$\frac{200,00 \text{ €}}{60 \text{ meses}} \times 4 \text{ meses} = 13,33\text{€}$$

3. Ratón Logitech gama media:

- Costo inicial: 50,00 euros
- Vida útil: 36 meses
- Tiempo de uso para el proyecto: 4 meses

$$\frac{50,00 \text{ €}}{36 \text{ meses}} \times 4 \text{ meses} = 5,56\text{€}$$

2.4.2. Costo total de hardware revisado

El costo total de los recursos de hardware revisado para este proyecto sería la suma de los costos de uso de cada recurso, es decir, 80,00 euros + 13,33 euros + 5,56 euros = 98,89 euros.

Recurso de hardware	Costo de uso (€)
MSI Modern A15 A10RAS Core i5 de 10 ^a generación	80,00
Monitor Samsung gama media	13,33
Ratón Logitech gama media	5,56
Costo total de hardware	98,89

Cuadro 2.3: Costos de uso de los recursos de hardware revisados

2.4.3. Otros costos adicionales revisados

Además de los recursos de hardware, es necesario ajustar los costos adicionales como la conexión a Internet. Supongamos un costo mensual promedio de 30,00 euros para una conexión de banda ancha. Dado que el proyecto tomó 4 meses, el costo total de la conexión a Internet se puede calcular de la siguiente manera:

$$\text{Costo total (€)} = \text{Costo mensual de Internet (€)} \times \text{Meses para el proyecto} \quad (2.5)$$

Aplicando la fórmula, el costo total de la conexión a Internet sería:

$$\text{Costo total de la conexión a Internet} = 30,00 \text{ €} \times 4 \text{ meses} = 120,00 \text{ €}$$

2.4.4. Presupuesto total revisado

Sumando todos los costos revisados, el presupuesto total para el proyecto sería:

Concepto	Costo (€)
Costo total de hardware	98,89
Costo total de software	0,00
Costo total de RRHH (incluyendo Seguridad Social)	5.890,50
Costo total de conexión a Internet	120,00
Contingencias y margen (10 %)	610,94
Costo total del proyecto	6.720,33

Cuadro 2.4: Resumen del presupuesto total revisado

Calculado bajo la suposición de que el proyecto se completaría en 1,5 meses, el presupuesto inicial se estableció en 6.565,63 euros. Sin embargo, la extensión real del proyecto a 4 meses resultó en un aumento de los costos totales. Después de revisar y ajustar los costos de hardware y otros gastos adicionales, el presupuesto total corregido ascendió a 6.720,33 euros. Esta diferencia de 154,70 euros, aunque no parece significativa en términos absolutos, refleja un problema subyacente mucho más importante: la sobreestimación del tiempo.

El principal desafío fue la duplicación de la duración del proyecto, lo cual tuvo un impacto considerable en la gestión de recursos y en la planificación general. Esta extensión temporal no se había previsto inicialmente, y subraya la importancia de realizar estimaciones de tiempo y costos más precisas durante la fase de planificación del proyecto. Una estimación adecuada desde el inicio podría haber ayudado a evitar desviaciones significativas y a asegurar una mejor gestión de los recursos financieros.

Además, una planificación más precisa habría permitido anticipar y mitigar los riesgos asociados con la prolongación del proyecto, garantizando así la calidad del trabajo final. En proyectos futuros, es crucial considerar todos los factores que pueden influir en la duración y el costo, como días no laborables, festividades y una posible reducción en la jornada laboral efectiva. De esta manera, se podrán establecer presupuestos más realistas y gestionar los recursos de manera más eficiente, mejorando así la viabilidad y la calidad de los resultados del proyecto.

Parte II

Documentación técnica

Capítulo 3

Análisis

La sección de análisis se centra en la evaluación detallada de los requisitos en la sección 3.1, detallando las acciones y comportamientos esperados del sistema, mientras que la siguiente sección 3.2 se centra en los Casos de uso, que proporcionan una representación gráfica y descriptiva de las interacciones entre usuarios y sistema, mostrando cómo se utilizará el sistema en escenarios concretos.

3.1. Requisitos

Los requisitos son elementos esenciales en la ingeniería de software, ya que definen las funcionalidades y características que el sistema debe cumplir para satisfacer las necesidades de los usuarios finales y otras partes interesadas. Los Requisitos Funcionales (RF) describen las acciones específicas que el sistema debe realizar en respuesta a las entradas del usuario, eventos del sistema o condiciones específicas (ver sección 3.1.1).

Los Requisitos No Funcionales (RNF), por otro lado, establecen los estándares de calidad que el sistema debe cumplir en términos de rendimiento, seguridad, usabilidad y otros atributos de calidad (ver sección 3.1.2). Estos requisitos son cruciales para asegurar que el sistema no solo funcione correctamente bajo condiciones normales, sino que también cumpla con las expectativas de rendimiento y experiencia del usuario en situaciones adversas o de carga intensa.

Además, los Atributos de Calidad se enfocan en aspectos como la eficiencia, la confiabilidad y la mantenibilidad del sistema (ver sección 3.1.3). Estos atributos son fundamentales para garantizar que el sistema no solo opere correctamente, sino que también sea fácil de mantener y evolucionar a lo largo del tiempo.

Los Atributos de Información definen cómo se manejan, almacenan y protegen los datos dentro del sistema, asegurando la integridad, disponibilidad y confidencialidad de la información (ver sección 3.1.4). Estos requisitos son críticos para cumplir con las regulaciones de privacidad y seguridad de los datos, así como para garantizar la confianza de los usuarios y otras partes interesadas.

Finalmente, las Restricciones delimitan los límites y condiciones bajo los cuales el sistema debe operar (ver sección 3.1.5).

3.1.1. Requisitos Funcionales (RF)

Los requisitos funcionales (RF) describen las funcionalidades específicas que el sistema debe proporcionar. Estos requisitos se detallan en la Tabla 3.1.

ID	Descripción
RF_01	El sistema debe permitir a los usuarios registrarse.
RF_01_Error	Si se escriben datos incorrectos o incompletos, el sistema debe mostrar mensajes de error específicos.
RF_02	El sistema debe permitir a los usuarios iniciar sesión.
RF_02_Error	Si las credenciales son incorrectas, el sistema debe mostrar un mensaje de error.
RF_03	Los usuarios deben poder modificar y eliminar sus cuentas.
RF_05	Los usuarios deben poder crear, ver, editar y eliminar sus propios sitios de interés.
RF_06	Los usuarios deben poder ver y comentar los sitios de interés de otros usuarios y los suyos propios.
RF_06_Error1	Si el comentario está vacío o supera el límite de caracteres, se debe mostrar un mensaje de error.
RF_06_Error2	Si la imagen no tiene el formato adecuado debe mostrar un mensaje de error.
RF_07	Los usuarios deben poder añadir sitios de interés de otros usuarios y propios a favoritos.
RF_08	Los usuarios deben poder buscar sitios de interés por diferentes criterios (ciudad, favoritos, etc.).
RF_09	Los usuarios deben poder crear un sitio de interés introduciendo los datos manualmente.
RF_10	Los usuarios deben poder importar sitios de interés mediante archivos KML.
RF_11	Los usuarios deben poder exportar los sitios de interés en formato KML.
RF_12	Los usuarios deben poder agregar imágenes a los sitios de interés.
RF_12_Error	Si la imagen no tiene el formato adecuado debe mostrar un mensaje de error.

RF_13	Los usuarios deben poder ver un historial de sus sitios importados.
-------	---

Cuadro 3.1: Requisitos Funcionales

3.1.2. Requisitos No Funcionales (RNF)

Los requisitos no funcionales (RNF) especifican criterios que pueden utilizarse para juzgar el funcionamiento de un sistema, en lugar de comportamientos específicos. Estos requisitos se detallan en la Tabla 3.2.

ID	Descripción
RNF_01	El sistema debe ser seguro, protegiendo la información personal de los usuarios.
RNF_02	La plataforma debe ser fácil de usar, con una interfaz de usuario intuitiva.
RNF_03	El sistema debe ser capaz de manejar un gran número de usuarios y sitios de interés sin degradar el rendimiento.
RNF_04	El sistema debe ser compatible con varios navegadores web y dispositivos.
RNF_05	El sistema debe tener un tiempo de respuesta aceptable, inferior a 3 segundos para operaciones comunes.
RNF_06	La plataforma debe ser accesible, cumpliendo con las directrices de accesibilidad web (WCAG).
RNF_07	El sistema debe ser mantenible y escalable para facilitar futuras mejoras y expansiones.
RNF_08	El sistema debe realizar copias de seguridad periódicas para garantizar la recuperación de datos en caso de fallo.
RNF_09	El sistema debe ser resiliente, manejando adecuadamente los errores y recuperándose sin pérdida de datos.
RNF_10	El sistema debe permitir el despliegue continuo (CI/CD) para facilitar la integración y entrega de nuevas funcionalidades.
RNF_11	El sistema debe ser auditable, permitiendo el registro y monitoreo de las acciones de los usuarios y del sistema.

Cuadro 3.2: Requisitos No Funcionales

3.1.3. Atributos de Calidad

Los atributos de calidad especifican los criterios relacionados con el rendimiento, la seguridad, la robustez y otros aspectos de la calidad del sistema. Estos atributos se detallan en la Tabla 3.3.

ID	Tipo	Descripción
AC_01	Rendimiento	El sistema debe tener un tiempo de respuesta inferior a 3 segundos para la mayoría de las operaciones comunes.
AC_02	Rendimiento	Las visualizaciones de sitios de interés en el mapa deben renderizarse de forma eficiente, incluso con un gran volumen de datos.
AC_03	Seguridad	El sistema debe proteger la información personal de los usuarios mediante encriptación y otras medidas de seguridad.
AC_04	Robustez	El sistema debe ser capaz de manejar entradas incorrectas de los usuarios sin causar fallos.
AC_05	Usabilidad	La interfaz debe ser intuitiva y fácil de usar, permitiendo a los usuarios navegar y utilizar las funcionalidades sin dificultad.
AC_06	Escalabilidad	El sistema debe ser capaz de manejar un aumento en el número de usuarios y sitios de interés sin una degradación significativa del rendimiento.
AC_07	Mantenibilidad	El sistema debe ser mantenible y escalable, facilitando futuras mejoras y expansiones.
AC_08	Compatibilidad	El sistema debe ser compatible con múltiples navegadores web (como Chrome, Firefox, Safari) y dispositivos (móviles y de escritorio).
AC_09	Accesibilidad	La plataforma debe cumplir con las directrices de accesibilidad web (WCAG) para garantizar que sea usable por personas con discapacidades.
AC_10	Reusabilidad	Los componentes del sistema deben ser modulares y reutilizables para facilitar su adaptación y reutilización en el futuro.

Cuadro 3.3: Atributos de Calidad

3.1.4. Atributos de Información

Los atributos de información describen las características relacionadas con los datos manejados por el sistema, como su formato, integridad y accesibilidad. Estos atributos se detallan en la Tabla 3.4.

ID	Descripción
INF_01	El sistema debe permitir la exportación de sitios de interés a archivos KML.
INF_02	Los sitios de interés deben incluir información detallada como nombre, descripción y ubicación.
INF_03	Los usuarios deben poder comentar y añadir imágenes a los sitios de interés.
INF_04	Los sitios de interés deben poder ser marcados como favoritos por los usuarios.
INF_05	Los usuarios que han creado un sitio de interés deben poder editar y eliminar dichos sitios.
INF_06	La aplicación debe permitir la visualización de sitios de interés en un mapa interactivo.
INF_07	El sistema debe proporcionar una funcionalidad de búsqueda y filtrado de sitios de interés.
INF_08	Los datos de los sitios de interés deben ser actualizables en tiempo real.
INF_09	El sistema debe ofrecer notificaciones a los usuarios sobre cambios o actualizaciones en los sitios de interés que siguen.
INF_10	Los sitios de interés deben ser compartibles a través de redes sociales o por correo electrónico.

Cuadro 3.4: Atributos de Información

3.1.5. Restricciones

Las restricciones son limitaciones que deben tenerse en cuenta en el desarrollo y la implementación del sistema. Pueden incluir limitaciones técnicas, legales o de otro tipo que influyan en el diseño y desarrollo del sistema. Estas restricciones se detallan en la Tabla 3.5.

ID	Descripción
RES_01	El sistema debe estar desarrollado en PHP, HTML y JavaScript.
RES_02	El sistema debe ser compatible con el formato KML para la importación y exportación de sitios de interés.
RES_03	El sistema debe ser implementado siguiendo las mejores prácticas de desarrollo web y seguridad.
RES_04	El sistema debe operar en una infraestructura que soporte CI/CD.
RES_05	La base de datos debe ser MySQL.

Cuadro 3.5: Restricciones

3.2. Casos de Uso

Los casos de uso son una técnica utilizada en la ingeniería de software para capturar los requisitos funcionales del sistema desde la perspectiva de los usuarios finales. Cada caso de uso representa una interacción específica entre un actor (usuario) y el sistema, describiendo el flujo de eventos que ocurren durante esa interacción. Estos casos de uso son fundamentales para definir y comprender los comportamientos esperados del sistema, asegurando que se cumplan las necesidades y expectativas de los usuarios.

En la primera subsección 3.2.1 se mostrará el diagrama de casos completo, mientras que la siguiente subsección 3.2.2 se mostrarán las tablas de casos de uso específicos

3.2.1. Diagrama de Casos de Uso

En esta sección se presenta el diagrama de casos de uso del sistema, el cual ilustra las interacciones principales entre los actores y el sistema. Este diagrama proporciona una visión general de los diferentes roles de usuario y las funcionalidades del sistema a las que tienen acceso. Cada actor está representado por un muñeco y los casos de uso por óvalos conectados por líneas que indican las interacciones entre ellos.

El diagrama de casos de uso sirve como una herramienta visual poderosa para entender la estructura del sistema desde la perspectiva de sus usuarios. A continuación 3.1, se muestra el diagrama:

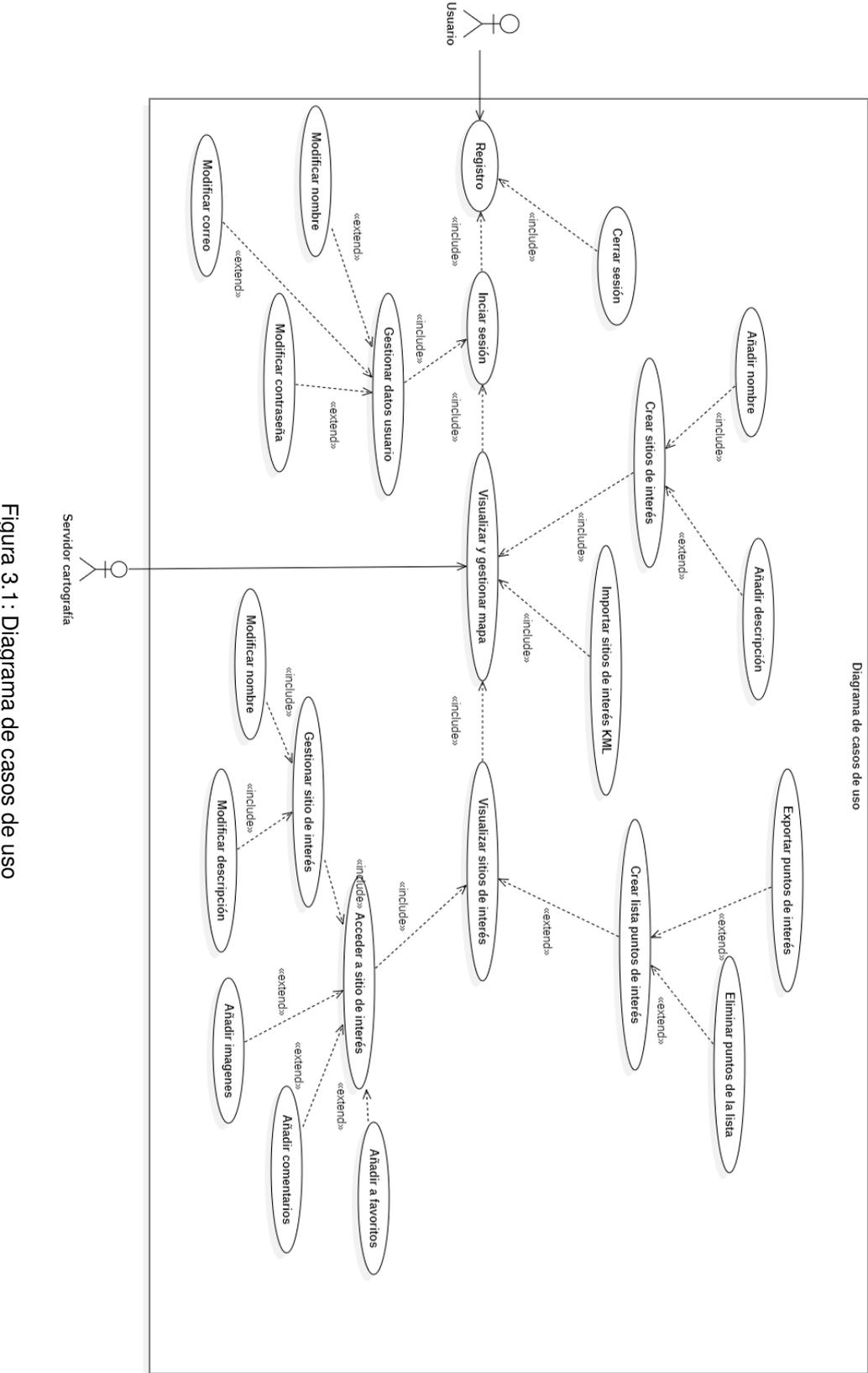


Figura 3.1 : Diagrama de casos de uso

3.2.2. Tablas de Casos de Uso

A continuación se detallan algunos de los casos de uso individuales identificados para el sistema, cada uno describiendo una funcionalidad específica desde la perspectiva del usuario. Las tablas proporcionan información detallada sobre cada caso de uso, incluyendo sus actores involucrados, precondiciones, flujo básico de eventos y postcondiciones esperadas.

Nombre e ID del CU	CU-01. Registro de usuario
Actor	Usuario
Descripción	El usuario accede a la página de registro, introduce sus datos personales y crea una cuenta en el sistema.
Precondiciones	PRE-1. El usuario no está registrado en el sistema.
Postcondiciones	POST-1. La cuenta del usuario queda creada y almacenada en el sistema.
Flujo normal	<p>FN1 El actor accede a la página de registro.</p> <p>FN2 El actor introduce sus datos personales.</p> <p>FN3 El sistema verifica la validez de los datos introducidos.</p> <p>FN4 El sistema crea la cuenta del usuario y confirma el registro.</p>
Flujo alternativo 1	<p>FA3 Si los datos son incorrectos o están incompletos, se informa al usuario del error y se le solicita corregirlos.</p>
Excepciones	<p>E1 El correo electrónico ya está registrado.</p> <p>E2 El usuario ha dejado campos requeridos sin rellenar.</p> <p>E3 El sistema está temporalmente fuera de servicio.</p>
Prioridad	Alta

Cuadro 3.6: CU-01. Registro de usuario

Nombre e ID del CU	CU-02. Creación de un sitio de interés
Actor	Usuario
Descripción	El usuario selecciona un punto en el mapa principal, confirma la creación del sitio de interés y completa los datos necesarios para añadirlo a su perfil.
Precondiciones	PRE-1. El usuario está registrado e identificado en el sistema.
Postcondiciones	POST-1. El nuevo sitio de interés queda registrado y vinculado al perfil del usuario.
Flujo normal	<p>FN1 El actor selecciona un punto en el mapa principal.</p> <p>FN2 El sistema pregunta al actor si desea crear un nuevo sitio de interés en esa ubicación.</p> <p>FN3 El actor confirma la creación del nuevo sitio de interés.</p> <p>FN4 El sistema redirige al actor a la página de creación de sitios de interés con los datos de ubicación ya completados.</p> <p>FN5 El actor introduce los datos restantes del sitio de interés (nombre y descripción opcional).</p> <p>FN6 El sistema verifica la validez de los datos introducidos.</p> <p>FN7 El sistema añade el sitio de interés al perfil del usuario y confirma la creación.</p>
Flujo alternativo 1	<p>FA5 Si los datos son incorrectos o están incompletos (excepto la descripción), se informa al usuario del error y se le solicita corregirlos.</p>
Excepciones	<p>E1 El usuario no tiene permisos para crear sitios de interés.</p> <p>E2 El usuario ha dejado campos requeridos sin rellenar (nombre).</p> <p>E3 El sistema está temporalmente fuera de servicio.</p>
Prioridad	Alta
Otra información	El sistema diferencia entre la acción de mover el mapa y la acción de hacer clic en un punto específico. Al hacer clic en el mapa, se despliega una ventana emergente <i>pop-up</i> que pregunta al usuario si desea crear un nuevo sitio de interés en esa ubicación, permitiendo cancelar la acción si se decide no crear el punto.

Cuadro 3.7: CU-02. Creación de un sitio de interés

Nombre e ID del CU	CU-03. Importar un sitio de interés mediante un archivo KML
Actor	Usuario
Descripción	El usuario accede a la página de importación de sitios de interés, selecciona un archivo KML de su dispositivo y lo importa a su perfil.
Precondiciones	PRE-1. El usuario está registrado e identificado en el sistema.
Postcondiciones	POST-1. Los datos del sitio de interés del archivo KML quedan registrados y vinculados al perfil del usuario.
Flujo normal	<p>FN1 El actor accede a la página de importación de sitios de interés.</p> <p>FN2 El actor selecciona un archivo KML desde su dispositivo.</p> <p>FN3 El sistema valida el archivo KML.</p> <p>FN4 El sistema importa los datos del archivo KML y los vincula al perfil del usuario.</p>
Flujo alternativo 1	<p>FA3 Si el archivo KML es inválido o está corrupto, se informa al usuario del error y no se procede a la importación.</p>
Excepciones	<p>E1 El archivo KML contiene datos incompletos.</p> <p>E2 El archivo KML no es compatible con el sistema.</p> <p>E3 El sistema está temporalmente fuera de servicio.</p>
Prioridad	Alta
Otra información	El usuario puede visualizar y editar los sitios de interés importados.

Cuadro 3.8: CU-03. Importar un sitio de interés mediante un archivo KML

Nombre e ID del CU	CU-04. Gestionar punto de interés
Actor	Usuario
Descripción	El usuario que ha subido el sitio de interés puede comentar, añadir imágenes, añadir a favoritos y editar los atributos del sitio.
Precondiciones	PRE-1. El usuario está registrado e identificado en el sistema.
Postcondiciones	POST-1. Los comentarios, imágenes, favoritos o ediciones quedan registrados y vinculados al perfil del usuario.
Flujo normal	<p>FN1 El actor accede a la página del sitio de interés.</p> <p>FN2 El actor introduce un comentario, añade una imagen, marca el sitio de interés como favorito o edita los atributos del sitio.</p> <p>FN3 El sistema verifica la validez de los datos introducidos.</p> <p>FN4 El sistema guarda el comentario, imagen, favorito o edición y confirma la acción.</p>
Flujo alternativo 1	<p>FA3 Si los datos son incorrectos o están incompletos, se informa al usuario del error y se le solicita corregirlos.</p>
Excepciones	<p>E1 El usuario no tiene permisos para gestionar el punto de interés.</p> <p>E2 El usuario ha dejado campos requeridos sin rellenar.</p> <p>E3 El sistema está temporalmente fuera de servicio.</p>
Prioridad	Media
Otra información	Los atributos del sitio pueden ser editados por el usuario en cualquier momento. Los usuarios que no han subido el sitio de interés pueden comentar, añadir imágenes y añadir a favoritos, pero no pueden editar estos atributos.

Cuadro 3.9: CU-04. Gestionar punto de interés

Nombre e ID del CU	CU-05. Exportar puntos de interés a KML
Actor	Usuario
Descripción	El usuario puede seleccionar puntos de interés en el mapa y exportarlos a un archivo KML.
Precondiciones	PRE-1. El usuario está registrado e identificado en el sistema.
Postcondiciones	POST-1. Se genera y descarga un archivo KML con los puntos de interés seleccionados.
Flujo normal	<p>FN1 El actor accede al mapa donde se muestran los puntos de interés con marcadores.</p> <p>FN2 El actor selecciona puntos de interés haciendo clic en los marcadores mientras mantiene presionada la tecla Control (Ctrl).</p> <p>FN3 Los puntos de interés seleccionados se añaden a una lista de selección.</p> <p>FN4 El actor presiona el botón "Exportar".</p> <p>FN5 El sistema genera un archivo KML con los puntos de interés seleccionados.</p> <p>FN6 El sistema ofrece la descarga del archivo KML al usuario.</p>
Flujo alternativo 1	<p>FA4 El actor puede cancelar la operación en cualquier momento, eliminando todos los puntos de la lista de selección.</p>
Excepciones	<p>E1 El usuario no tiene permisos para exportar puntos de interés.</p> <p>E2 El sistema está temporalmente fuera de servicio.</p>
Prioridad	Media
Otra información	El actor puede eliminar puntos de la lista de selección desde la lista misma o volviendo a hacer clic en los marcadores en el mapa.

Cuadro 3.10: CU-05. Exportar puntos de interés a KML

Capítulo 4

Diseño

El capítulo de Diseño proporciona una guía detallada para estructurar y organizar la arquitectura del sistema, definiendo tanto su arquitectura lógica como física (ver sección 4.1), así como la organización y estructura de los datos dentro del sistema (ver sección 4.2). También permite visualizar las interacciones dinámicas entre los componentes del sistema mediante diagramas de secuencia (ver sección 4.3), asegurando que el diseño del sistema cumpla con los requisitos definidos en las etapas anteriores del ciclo de vida del desarrollo de software (ver sección 3.1).

4.1. Arquitectura

La arquitectura de software es una disciplina crucial en la ingeniería de software que se encarga del diseño y planificación a un nivel superior de un sistema. Este concepto, que surgió en los años 60, se refiere a una planificación basada en modelos, patrones y abstracciones teóricas, necesaria para desarrollar software de cierta complejidad antes de cualquier implementación. La arquitectura de software proporciona una guía detallada para entender cómo las diferentes partes de un sistema encajan entre sí, facilitando decisiones críticas antes de iniciar la programación.

La arquitectura no solo nos permite elegir el conjunto de herramientas y metodologías más adecuadas para un proyecto, sino que también influye en el ritmo de desarrollo, los costos y la gestión de recursos humanos. En este contexto, los patrones de arquitectura son soluciones generales y reutilizables para problemas recurrentes en la ingeniería de software, similares a los patrones en programación pero aplicados a un nivel más abstracto y general.

El Instituto de Ingeniería de Software (SEI) define la arquitectura de software como “el conjunto de estructuras necesarias para razonar sobre el sistema, comprendiendo elementos de software, relaciones entre ellos y propiedades de ambos” [2]. Esta definición resalta la importancia de los elementos de software, que incluyen módulos, clases, objetos e hilos, y sus interfaces, que son esenciales para la integración y prueba de las partes del sistema.

4.1.1. Arquitectura Lógica

La arquitectura lógica de un sistema de software se centra en la organización de sus componentes en términos de módulos y sus relaciones lógicas. En este proyecto, se ha basado en un patrón Modelo-Vista-Controlador (MVC) para estructurar la lógica de la aplicación. Este patrón divide el sistema en tres componentes principales:

- **Modelo:** Maneja la lógica de datos de la aplicación e interactúa con la base de datos MySQL.
- **Vista:** Genera la interfaz de usuario dinámica a partir de los datos recibidos del controlador. El frontend de la aplicación está desarrollado utilizando HTML, CSS y JavaScript.
- **Controlador:** Gestiona la lógica de negocio, recibe solicitudes del cliente y coordina entre el modelo y la vista.

Este enfoque modular facilita la separación de responsabilidades y mejora la mantenibilidad y escalabilidad del sistema.

La arquitectura física de la aplicación se detalla en la Figura 4.2.

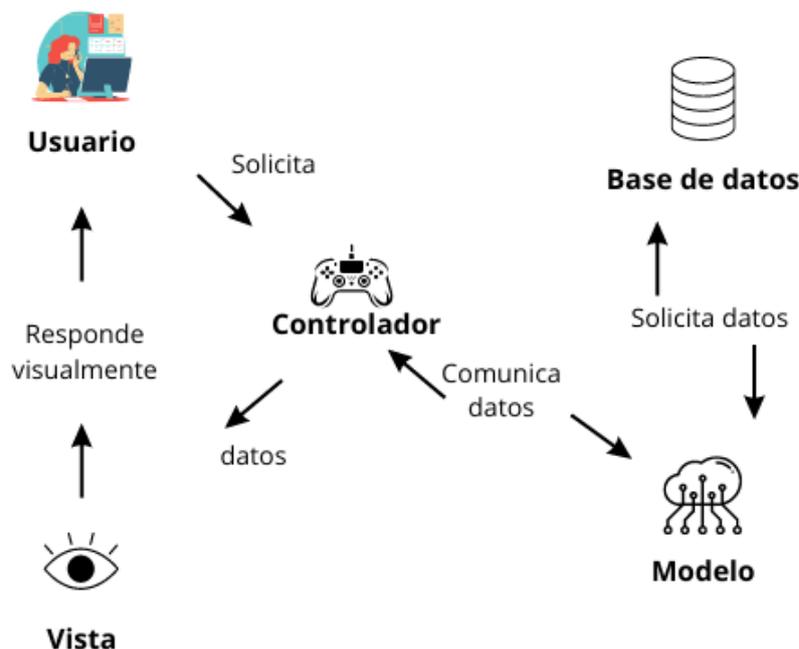


Figura 4.1: Arquitectura logica MVC.

4.1.2. Arquitectura Física

La arquitectura física de un sistema de software se refiere a la disposición y configuración de los componentes de hardware y software en el entorno de ejecución. En nuestro

proyecto, la arquitectura física está basada en un modelo cliente-servidor, que incluye los siguientes elementos:

- **Servidores:** La aplicación se despliega en servidores con el sistema operativo Ubuntu, utilizando Apache como servidor web.
- **Red:** La comunicación entre el cliente y el servidor se gestiona a través de una red segura, utilizando HTTPS para cifrar los datos transmitidos.
- **Almacenamiento:** Los datos se almacenan en una base de datos MySQL alojada en los servidores de la Universidad de Valladolid.
- **Servicios Externos:** La aplicación integra servicios externos como APIs de Leaflet para la visualización de mapas y datos geoespaciales.

La arquitectura física de la aplicación se detalla en la Figura 4.2. La infraestructura incluye:

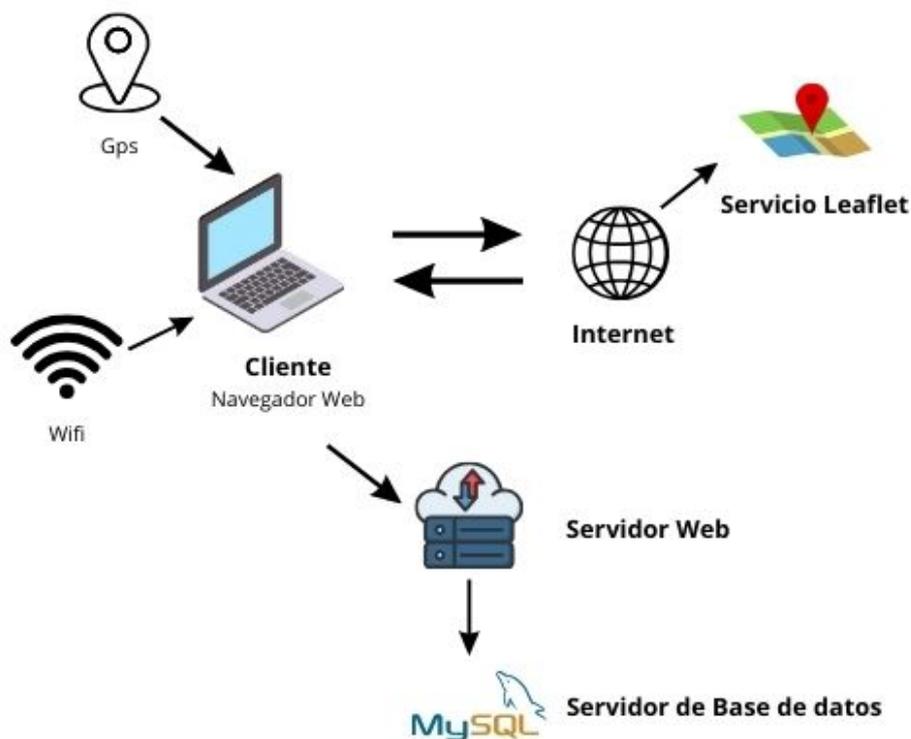


Figura 4.2: Arquitectura física cliente-servidor.

4.2. Diseño de Datos

En esta sección se presenta el diseño lógico de la base de datos utilizada en el sistema, así como el diccionario de datos correspondiente. El diseño de datos es crucial para garantizar la integridad y eficiencia en la gestión de la información dentro del sistema.

4.2.1. Diseño lógico de la base de datos

El modelo de datos se diseñó para cubrir las necesidades de la aplicación, permitiendo a los usuarios registrarse, crear y gestionar puntos de interés, así como interactuar con los mismos a través de comentarios, imágenes y favoritos. A continuación se presenta el diagrama entidad-relación (ER) y el modelo relacional de la base de datos.

Modelo entidad-relación:

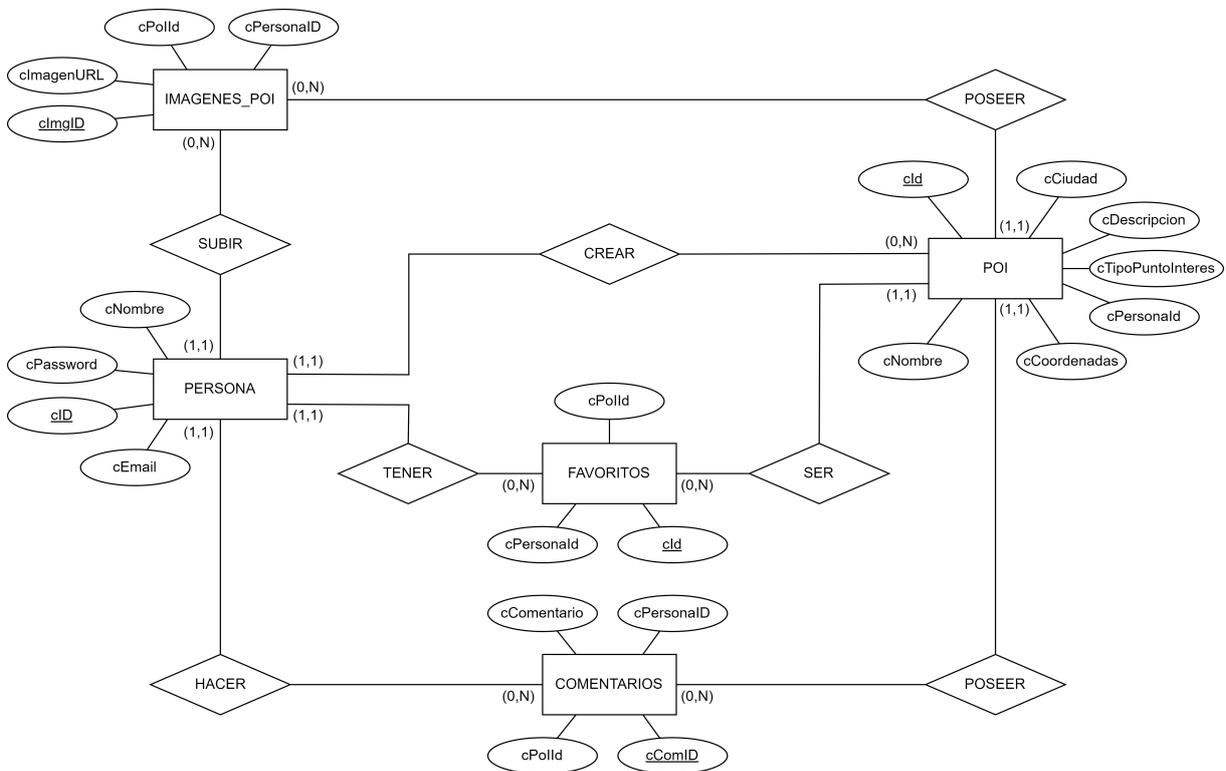


Figura 4.3: Modelo entidad-relación (ER).

Modelo relacional:**Persona** (cID, cNombre, cEmail, cPassword)**PoI** (cId, cNombre, cCiudad, cCoordenadas, cPersonaId, cDescripcion, cTipoPuntoInteres)

- **Fk:** cPersonaId → Persona(cID)

Comentarios (cComID, cPoIID, cPersonaId, cComentario)

- **Fk:** cPoIID → PoI(cId)
- **Fk:** cPersonaId → Persona(cID)

Favoritos (cId, cPersonaId, cPoIID)

- **Fk:** cPersonaId → Persona(cID)
- **Fk:** cPoIID → PoI(cId)

ImagenesPoI (cImgID, cPoIID, cPersonaId, cImagenURL)

- **Fk:** cPoIID → PoI(cId)
- **Fk:** cPersonaId → Persona(cID)

4.2.2. Diccionario de Datos

El diccionario de datos detalla los atributos de cada entidad en la base de datos, especificando su tipo, si es requerido, y si es único.

Entidad: Persona

Atributo	Tipo	Requerido	Único
cID	int	Sí	Sí
cNombre	varchar(255)	Sí	No
cEmail	varchar(255)	Sí	No
cPassword	varchar(64)	Sí	No

Cuadro 4.1: Diccionario de datos de la entidad "Persona"

- **cID:** Identificador único de la persona.
- **cNombre:** Nombre de la persona.
- **cEmail:** Correo electrónico de la persona.
- **cPassword:** Contraseña de la persona (hash).

Entidad: PoI

Atributo	Tipo	Requerido	Único
cId	int	Sí	Sí
cNombre	varchar(50)	Sí	No
cCiudad	varchar(50)	No	No
cCoordenadas	text	Sí	No
cPersonaId	int	Sí	No
cDescripcion	text	No	No
cTipoPuntoInteres	ENUM(P,R)	No	No

Cuadro 4.2: Diccionario de datos de la entidad "PoI"

- **cId:** Identificador único del punto de interés.

- **cNombre:** Nombre del punto de interés.
- **cCiudad:** Ciudad donde se ubica el punto de interés.
- **cCoordenadas:** Coordenadas geográficas del punto de interés.
- **cPersonaId:** Identificador de la persona que creó el punto de interés.
- **cDescripcion:** Descripción del punto de interés.
- **cTipoPuntoInteres:** Tipo de punto de interés. P se refiere a punto de interés y R a ruta, facilitando su distinción.

Entidad: Comentarios

Atributo	Tipo	Requerido	Único
cComID	int	Sí	Sí
cPoIID	int	Sí	No
cPersonaId	int	Sí	No
cComentario	text	Sí	No

Cuadro 4.3: Diccionario de datos de la entidad "Comentarios"

- **cComID:** Identificador único del comentario.
- **cPoIID:** Identificador del punto de interés comentado.
- **cPersonaId:** Identificador de la persona que hizo el comentario.
- **cComentario:** Texto del comentario.

Entidad: Favoritos

Atributo	Tipo	Requerido	Único
cId	int	Sí	Sí
cPersonaId	int	Sí	No
cPoIID	int	Sí	No

Cuadro 4.4: Diccionario de datos de la entidad "Favoritos"

- **cId:** Identificador único del favorito.

- **cPersonaId:** Identificador de la persona que agregó el favorito.
- **cPoIID:** Identificador del punto de interés marcado como favorito.

Entidad: ImagenesPoI

Atributo	Tipo	Requerido	Único
cImgID	int	Sí	Sí
cPoIID	int	Sí	No
cPersonaId	int	Sí	No
cImagenURL	varchar(500)	Sí	No

Cuadro 4.5: Diccionario de datos de la entidad “ImagenesPoI”

- **cImgID:** Identificador único de la imagen.
- **cPoIID:** Identificador del punto de interés asociado a la imagen.
- **cPersonaId:** Identificador de la persona que subió la imagen.
- **cImagenURL:** URL de la imagen.

Nota: Se omite la especificación de otras características en las tablas como multivaloradas o compuestas, ya que todas son negativas.

4.2.3. Perfiles de Acceso

Dependiendo del rol del usuario, se le otorgarán diferentes niveles de acceso a la aplicación:

- **Administrador:** Tiene acceso completo para gestionar todas las funcionalidades y datos de la aplicación. Puede crear, modificar y eliminar puntos de interés de cualquier usuario. Además, puede gestionar comentarios, imágenes, rutas y favoritos. También tiene la capacidad de ver y administrar la lista de usuarios registrados en la plataforma.
- **Usuario Registrado con Puntos de Interés Propios:** Puede crear, modificar y eliminar sus propios puntos de interés. También puede añadir puntos de interés existentes a su cuenta y marcarlos como favoritos. Cuando un usuario añade un punto de interés a su cuenta, se crea una copia bajo su perfil. Esta copia puede ser modificada libremente por el usuario según sus necesidades. Los usuarios con puntos de interés también pueden gestionar sus datos personales y contribuciones dentro de la aplicación.

- **Usuario Registrado sin Puntos de Interés Propios:** Puede agregar comentarios y marcar puntos de interés como favoritos. Aunque no puede modificar puntos de interés existentes directamente, tiene la opción de guardar puntos de interés en su cuenta. Al guardar un punto de interés, se crea una copia bajo su perfil, la cual puede modificar a su antojo posteriormente. Los usuarios sin puntos de interés también pueden gestionar sus datos personales y contribuciones dentro de la aplicación.

Este diseño de perfiles garantiza que los usuarios puedan interactuar según su nivel de acceso específico, permitiendo a los usuarios gestionar puntos de interés de manera personalizada mientras el administrador mantiene el control integral sobre la plataforma.

4.3. Diagramas de secuencia

4.3.1. Estereotipos de clase de línea de vida

En los diagramas de secuencia, se utilizan tres estereotipos principales de clase de línea de vida para representar diferentes roles y responsabilidades dentro del sistema.

Entidad

Las **entidades** representan contenedores de datos fundamentales en el sistema. Son los objetos que contienen y manipulan datos relevantes para el funcionamiento del sistema.

Límite

Los estereotipos de clase de línea de vida llamados **límites** representan interfaces que interactúan con actores externos, como interfaces de usuario. Están diseñados para manejar la comunicación entre el sistema y los usuarios externos.

Control

Por último, los estereotipos de clase de línea de vida denominados **controles** son responsables de la lógica y la mediación entre las entidades y los límites en el sistema. Gestionan el flujo de control y la coordinación entre los diferentes componentes del sistema.



Figura 4.4: Estereotipos usados en un diagrama de secuencia. De izquierda a derecha: entidad, límite y control.

4.3.2. Diagrama de mostrar puntos cercanos

El siguiente diagrama 4.5 ilustra el proceso seguido en el código para mostrar los puntos de interés en el mapa a un usuario. En la página principal “Explorar”, se cargan los *scripts* `favoritos.js` y `utilidades.js`. El proceso comienza cuando el usuario interactúa con esta vista principal, lo que provoca que se invoque la función `cargarFavoritos()` en `favoritos.js`.

El primer paso consiste en obtener la lista de IDs de los puntos favoritos del usuario. Esta obtención se realiza entre los pasos 3 y 8 del diagrama. La función `cargarFavoritos()` realiza una llamada al controlador, el cual solicita al modelo la lista de favoritos del usuario. El modelo devuelve esta lista de IDs al controlador, quien posteriormente la reenvía a `favoritos.js`. Una vez finalizado el paso 7 del diagrama, se guarda la lista de IDs de favoritos en `favoritos.js`.

Una vez que se han obtenido y almacenado los favoritos, `favoritos.js` llama a la función `obtenerPuntosCercanos()` en `utilidades.js`. Este *script*, `utilidades.js`, solicita al controlador los puntos cercanos. El controlador, a su vez, consulta al modelo para obtener estos datos. El modelo devuelve los puntos cercanos al controlador, y este los retorna a `utilidades.js`.

Finalmente, en el paso 14 del diagrama, el *script* `utilidades.js` invoca la función `mostrarPuntosEnMapa(puntos)`, obteniendo estos puntos de `obtenerPuntosCercanos()`. Durante este proceso, se utiliza la lista de IDs de favoritos obtenida previamente para determinar qué marcador se debe dibujar para cada punto en el mapa. La diferenciación entre `Mis sitios` y `Otros sitios` se realiza directamente, comparando el `cPersonaId` del punto de interés (Poi) con el ID del usuario. No obstante, los puntos favoritos se almacenan en una tabla separada, lo que hace necesario este paso adicional de obtención y comparación.

Este flujo de trabajo asegura que los puntos de interés se muestren correctamente en el mapa, destacando adecuadamente los favoritos del usuario y distinguiéndolos de otros puntos. De esta manera, se facilita una experiencia de usuario más personalizada y útil.

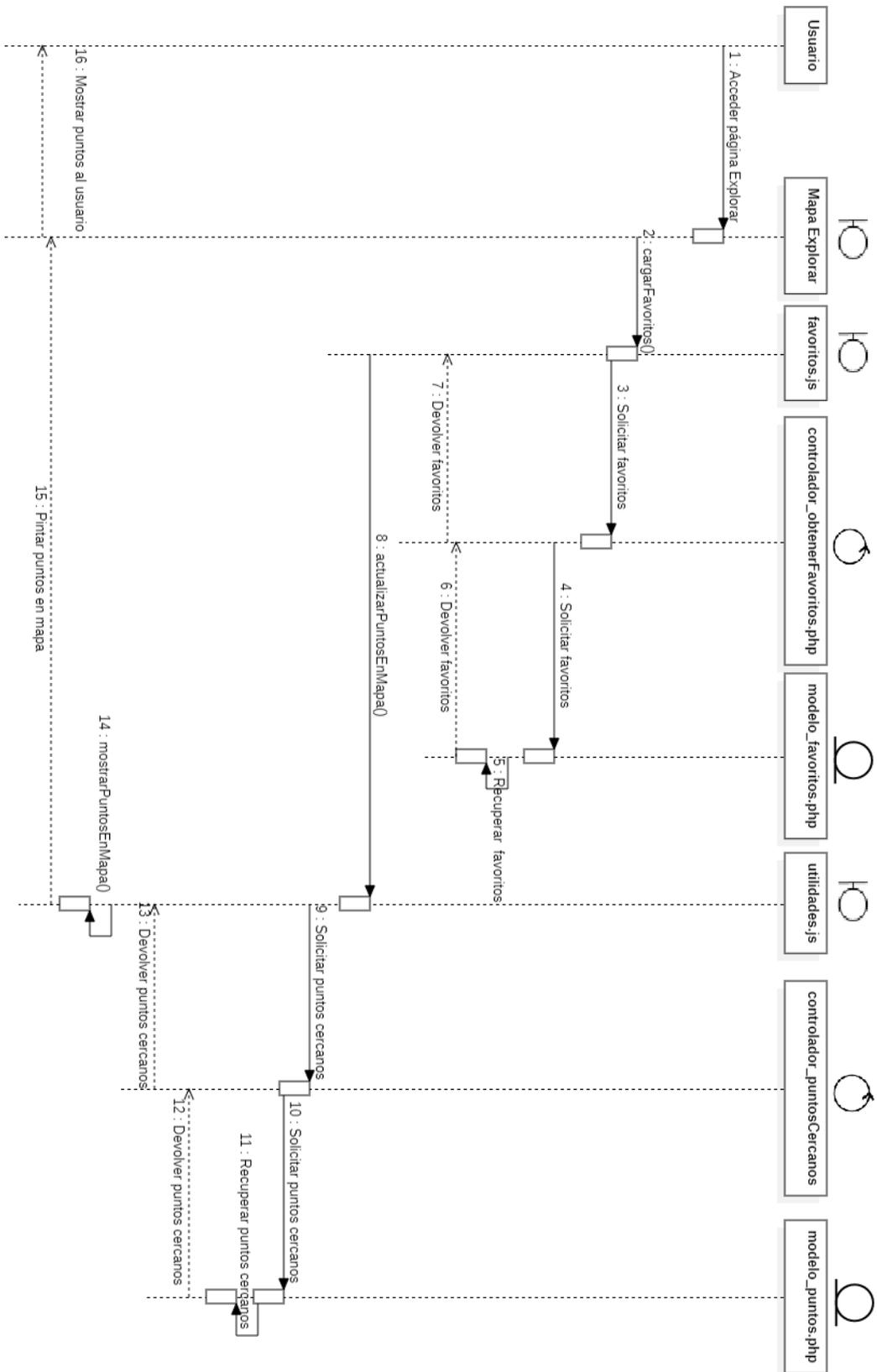


Figura 4.5: Diagrama de secuencia de mostrar puntos cercanos al entrar a explorar

4.3.3. Diagrama de exportacion

El siguiente diagrama 4.6 ilustra el proceso seguido en el código para hacer la selección y exportación de los puntos elegidos por el usuario en el mapa. El proceso de exportación de puntos seleccionados en la aplicación web comienza cuando el usuario interactúa con la interfaz principal del mapa, es decir, partimos justo después del paso anterior de mostrar todos los puntos.

Durante la interacción, el usuario puede seleccionar varios puntos en el mapa haciendo CTRL+clic sobre los marcadores que representan esos puntos. Este proceso de selección se realiza en un ciclo entre el usuario con el mapa, `utilidades.js` que es el encargado de gestionar qué tipo de clic es, y `exportar.js` que contiene toda la logica de la manipulación de la información los puntos. El usuario puede añadir puntos moviéndose por el mapa. Los puntos seleccionados se irán mostrando en una lista que aparece por encima del mapa, a la vez que sus marcadores se cambiarán a un color rojo para mejorar su distinción.

Una vez que el usuario ha completado la selección de puntos, tiene dos opciones disponibles: exportar los puntos seleccionados como un archivo KML o cancelar la operación de exportación. En este caso solo se ha modelado la opción de exportación sin excepción, pero si el usuario pulsase en “cancelar” se seguiría el subproceso de cancelación, que coincide con los pasos que se siguen al terminar la exportación.

Cuando el usuario decide exportar los puntos seleccionados, al pulsar el botón “exportar”, la función `exportarSeleccionados()` recopila los detalles de los puntos seleccionados y genera dinámicamente un archivo KML. Después de generar el archivo KML, se inicia automáticamente la descarga del archivo en el navegador del usuario. Este proceso asegura que el usuario pueda guardar localmente el archivo KML generado para su uso posterior en aplicaciones compatibles con KML, como Google Earth.

Una vez acaba la lógica de la exportacion se elimina la lista de los puntos seleccionados y el mapa vuelve a su estado original.

El ciclo de selección de puntos y las opciones de exportar o cancelar proporcionan una experiencia interactiva y flexible al usuario, facilitando la gestión y manipulación de datos geoespaciales dentro de la aplicación web.

Capítulo 5

Implementación

En este capítulo se mostrarán los detalles más relevantes en cuanto a la implementación del sistema.

5.1. Introducción

En esta sección se explicará brevemente el objetivo de la parte de implementación del proyecto. Se dará una visión general de las principales áreas cubiertas: inicialización y finalización (ver en sección 5.1.1), consultas seguras (ver en sección 5.2), integración de mapas con Leaflet (ver en sección 5.3) y la importación/exportación KML (ver en las secciones 5.4 y 5.5).

Estructura del Proyecto

En esta sección se describe la organización de los archivos y directorios del proyecto. Se proporciona una visión general de las carpetas principales 5.1 y una descripción detallada de las subcarpetas y archivos contenidos en ellas 5.1.

Estructura de las carpetas (alto nivel)

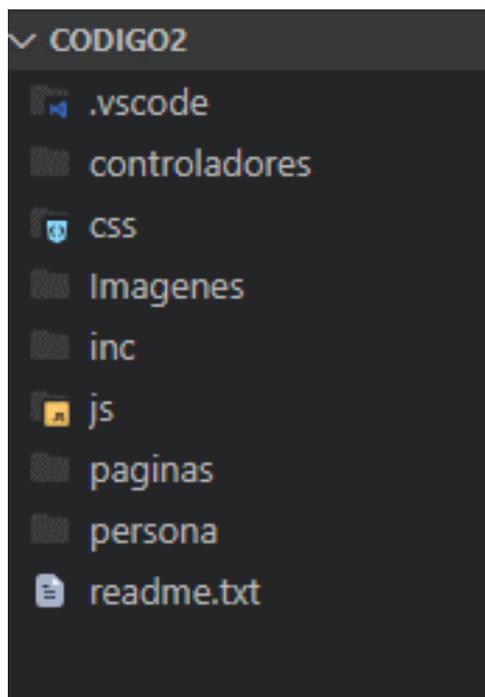
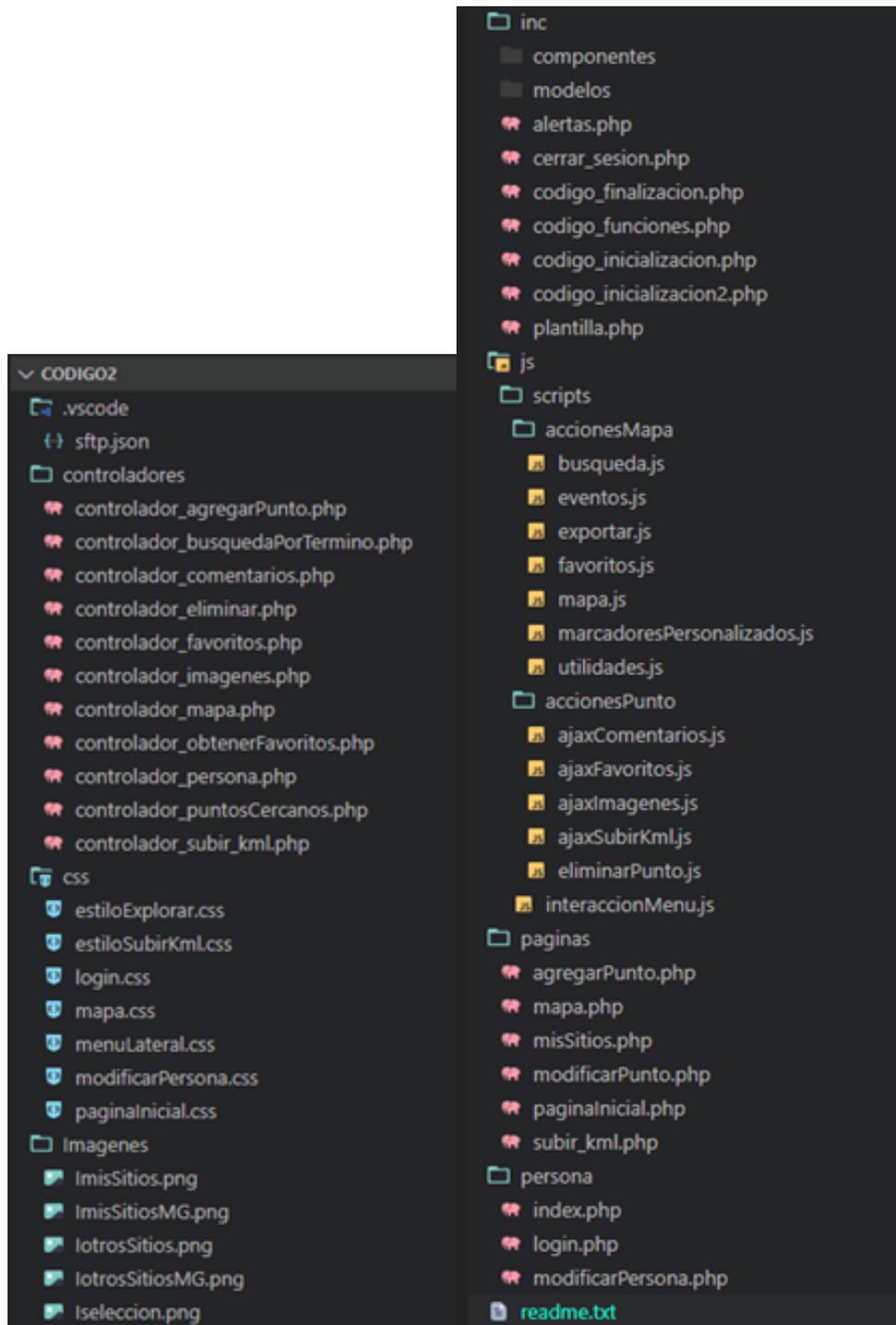


Figura 5.1: Estructura de las carpetas.

Estructura de las carpetas (bajo nivel)



Explicación de las carpetas

- **.vscode.** Contiene configuraciones específicas de Visual Studio Code, como el archivo `sftp.json`, que está configurado para conexiones SFTP a la máquina virtual.
- **controladores.** Agrupa *scripts* PHP que manejan las diversas operaciones lógicas del lado del servidor, como agregar puntos, búsqueda por término, manejo de comentarios, eliminación de datos, gestión de favoritos, operaciones con imágenes, control del mapa, gestión de usuarios y obtención de favoritos.
- **css.** Contiene archivos CSS que definen el estilo visual de las páginas web del proyecto, incluyendo estilos específicos para explorar, subir KML, *login*, mapa, menú lateral, modificar persona y página inicial.
- **Imágenes.** Almacena imágenes utilizadas en el proyecto, como los íconos de los marcadores 5.6.1.
- **inc.** Contiene *scripts* PHP compartidos y funciones comunes, incluyendo archivos para alertas, cierre de sesión, inicialización del código, plantilla base y componentes como formularios para *login*, modificación de persona, modificación de punto y creación de persona. También incluye modelos para agregar punto, comentarios, copia de puntos de interés (PoI), eliminación, favoritos, imágenes, modelo de mapa, gestión de personas, gestión de puntos y subida de KML.
- **js.** Contiene *scripts* JavaScript utilizados para la funcionalidad interactiva del lado del cliente, incluyendo comportamiento del cliente, *scripts* externos, validaciones y acciones organizadas en directorios específicos para la interacción con el menú, acciones en el mapa, marcadores personalizados y utilidades.
- **páginas.** Almacena *scripts* PHP que representan diferentes vistas o páginas que los usuarios pueden ver e interactuar, como agregar punto, mapa, mis sitios, modificar punto, página inicial y subir KML.
- **persona.** Contiene *scripts* PHP relacionados con la gestión de usuarios o perfiles dentro del sistema, incluyendo el índice de la página, el inicio de sesión y la modificación de la información personal del usuario.

Esta estructura organizada del proyecto facilita el desarrollo modular, la colaboración y el mantenimiento, permitiendo una clara separación de responsabilidades y un flujo de trabajo eficiente entre los distintos componentes del equipo de desarrollo.

5.1.1. Inicialización y finalización

Para la conexión a la base de datos y gestión de sesiones, he creado dos archivos: `codigoInicializacion.php` y `codigoFinalizacion.php`. Estos archivos se incluyen al inicio y al final de cada *script* PHP para asegurar una gestión adecuada de la conexión a la base de datos y las sesiones.

Archivo de inicialización (init.php)

El archivo `init.php` contiene el código necesario para configurar y preparar el entorno al inicio de cada *script* PHP. A continuación se destacan las partes más importantes:

- Inicio de sesión y configuración del *buffer* de salida:

```
session_start();
ob_start();
```

- Definición de constantes para la gestión de errores y estados:

```
define("OK", 0);
define("ERROR_LOGIN", 7);
```

- Conexión a la base de datos:

```
$conexionBD = mysqli_connect($host, $usuario, $password)
    or die("No se pudo conectar a la base de datos");
mysqli_select_db($conexionBD, "BASEDEDATOS");
```

- Inclusión de otros archivos necesarios para la funcionalidad del sistema:

```
include_once "plantilla.php";
include_once "modelos/persona.php";
```

Archivo de finalización (close.php)

El archivo `close.php` contiene el código necesario para limpiar y cerrar el entorno al final de cada *script* PHP. Las partes más importantes son:

- Cierre de la conexión con la base de datos:

```
mysqli_close($conexionBD);
```

- Volcado del *buffer* de salida y desactivación del *buffering*:

```
ob_end_flush();
```

5.2. Consultas seguras a la base de datos

5.2.1. Descripción

Para proteger la aplicación de inyecciones SQL, se han utilizado consultas preparadas en todas las interacciones con la base de datos.

5.2.2. Ejemplo de inserción segura

```
<?php
$stmt2 = mysqli_prepare($conexionBD, "INSERT INTO RutasCoordenadas (cPoIID,
cCoordenadas) VALUES (?, ?)");
if (!$stmt2) {
    return "Error preparando la inserción en la tabla coordenadasRuta:
    " . mysqli_error($conexionBD);
}
mysqli_stmt_bind_param($stmt2, "is", $poiId, $coordinates);
?>
```

5.3. Implementación de mapas con Leaflet

Se ha integrado Leaflet para la visualización de mapas interactivos en la aplicación. Leaflet es una librería de código abierto que permite crear mapas interactivos de manera sencilla. Para poder implementarlo en código primero hay que incluir la librería de Leaflet en el proyecto. Finalmente se creará un contenedor HTML para el mapa, el cual se inicializará en JavaScript. Siendo esta la configuración básica, es después cuando puedes añadir capas y marcadores sobre esta base.

5.3.1. Configuración Básica para la Instalación

En esta sección se describe cómo incluir los archivos CSS y JS de Leaflet en el proyecto para poder utilizar esta biblioteca en la creación de mapas interactivos.

Incluir los archivos CSS y JS de Leaflet en el proyecto

Para comenzar a utilizar Leaflet en el proyecto, es necesario incluir los archivos CSS y JS que proporciona esta biblioteca. Esto se logra mediante los siguientes pasos:

```
<!-- Incluir Leaflet en el proyecto -->
<link rel="stylesheet" href="https://unpkg.com/leaflet/dist/leaflet.css" />
<script src="https://unpkg.com/leaflet/dist/leaflet.js"></script>
```

Archivos CSS y JS de Leaflet: Los archivos CSS y JS de Leaflet se enlazan desde la CDN de `unpkg.com`. Esto permite el proyecto tenga acceso a los estilos y funcionalidades esenciales de Leaflet para la representación y manipulación de mapas interactivos.

5.3.2. Crear un Contenedor para el Mapa

Una vez que se han incluido los archivos necesarios de Leaflet, es crucial crear un contenedor HTML donde se visualizará el mapa interactivo.

Crear un contenedor HTML para el mapa

Se debe crear un elemento HTML con un identificador único donde se mostrará el mapa generado por Leaflet. A continuación se muestra cómo configurar este contenedor:

```
<!-- Contenedor HTML para el mapa -->
<div id="map" style="height: 500px;"></div>
```

Contenedor del Mapa: Se crea un `<div>` con el atributo `id="map"` y un estilo en línea para especificar la altura del contenedor (`height: 500px;`). Este `div` actuará como el contenedor del mapa interactivo donde Leaflet renderizará la información geográfica.

5.3.3. Inicializar el Mapa

Una vez configurado el contenedor del mapa, se procede a inicializar y configurar el mapa utilizando Leaflet.

Inicializar el mapa con Leaflet

El siguiente código JavaScript muestra cómo inicializar un mapa básico utilizando Leaflet:

```
// Inicialización básica de un mapa con Leaflet
var map = L.map('map').setView([51.505, -0.09], 13);

L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png',
{
  attribution: '&copy;
  <a href="https://www.openstreetmap.org/copyright">
  OpenStreetMap</a> contributors'
}).addTo(map);
```

- **Inicialización del Mapa:** Se crea un nuevo mapa Leaflet utilizando `L.map('map')`, donde `'map'` es el identificador del contenedor HTML creado anteriormente. Se establece la vista inicial del mapa con `.setView([51.505, -0.09], 13)`, especificando las coordenadas geográficas y el nivel de *zoom*.
- **Capa Base de OpenStreetMap:** Se añade una capa base de OpenStreetMap al mapa utilizando `L.tileLayer`. Esta capa proporciona los azulejos (*tiles*) del mapa desde el servidor de OpenStreetMap y se configura con atribuciones de derechos de autor.

Este proceso de configuración básica y creación de mapa proporciona una base sólida para empezar a trabajar con Leaflet en la creación de aplicaciones web interactivas basadas en mapas.

5.3.4. Añadir marcadores y capas

Es posible añadir un marcador a una capa y asociarle un *popup* utilizando el siguiente código JavaScript:

```
<script>
// Crear un mapa Leaflet centrado en coordenadas específicas
var map = L.map('mapid').setView([51.5, -0.09], 13);

// Añadir una capa base de OpenStreetMap al mapa
L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
```

```
    maxZoom: 19,
  }).addTo(map);

// Añadir un marcador en una ubicación específica
var marker = L.marker([51.5, -0.09]).addTo(map);

// Vincular un popup al marcador creado
marker.bindPopup("<b>Hello world!</b><br>I am a popup.").openPopup();
</script>
```

Explicación del Código

- **Creación del Mapa:** Se inicializa un mapa Leaflet con `L.map('mapid')` y se establece la vista inicial en las coordenadas `[51.5, -0.09]` con un nivel de *zoom* de 13.
- **Capa Base de OpenStreetMap:** Se añade una capa base de OpenStreetMap al mapa utilizando `L.tileLayer`, que proporciona los azulejos del mapa desde el servidor de OpenStreetMap.
- **Añadir Marcador:** Se crea un marcador en las coordenadas `[51.5, -0.09]` utilizando `L.marker()` y se añade al mapa con `addTo(map)`.
- **Vincular *popup* al Marcador:** Se utiliza `bindPopup()` para asociar un *popup* al marcador. En este ejemplo, el *popup* contiene texto HTML formateado que se muestra al hacer clic en el marcador. `openPopup()` se utiliza para abrir automáticamente el *popup* cuando se carga el mapa.

Este código y explicación demuestran cómo implementar efectivamente la funcionalidad de añadir marcadores, capas base y *popups* en Leaflet, mejorando la interactividad y la visualización de datos geográficos en aplicaciones web.

5.4. Importación de Archivos KML

En este capítulo se detalla el proceso de importación de archivos KML en la aplicación, explicando las funciones involucradas, sus interrelaciones y las diferencias en el manejo de puntos de interés (PoI) y rutas.

5.4.1. Subida del Archivo KML

El proceso inicia con la subida del archivo KML mediante un formulario HTML estándar que permite al usuario seleccionar y enviar el archivo al servidor para su procesamiento.

5.4.2. Validación y Carga del Archivo KML

El proceso comienza con la validación del archivo KML subido por el usuario, asegurando que cumpla con el formato necesario para el procesamiento. Utilizamos la función `validarArchivoKML()` para este propósito, la cual verifica la extensión del archivo y lanza una excepción si el formato es incorrecto.

```
function validarArchivoKML($file) {  
    // Implementación de la validación del archivo KML  
}
```

Una vez validado, el archivo se carga utilizando `cargarKML`, convirtiéndolo en un objeto `SimpleXML` para facilitar la manipulación de los datos XML del KML.

```
function cargarKML($file) {  
    // Implementación de la carga del archivo KML  
}
```

5.4.3. Procesamiento de Puntos de Interés (PoI)

Subida y Validación de PoI

El archivo KML se desglosa en sus componentes principales: puntos de interés (PoI) y rutas. Para cada PoI encontrado, se extraen detalles como nombre, descripción y coordenadas. Se determina la ciudad asociada utilizando la función `obtenerCiudad`.

```
$cityName = obtenerCiudad($kml);
```

5.4.4. Inserción en la Base de Datos

Función de Inserción (`insertarPoi`)

Para insertar los PoI en la base de datos, se utiliza la función `insertarPoi()`, diseñada para manejar la inserción segura de datos en la tabla `PoI`. Esta función prepara y ejecuta consultas SQL parametrizadas para evitar vulnerabilidades de seguridad como la inyección SQL.

```
function insertarPoi($conexionBD, $nombreItem, $cityName, $combinedCoordinates,
                    $personaId, $tipoPuntoInteres, $descripcion) {
    // Implementación de la inserción segura de PoI en la base de datos
}
```

5.4.5. Manejo de Rutas y Elementos Desconocidos

Rutas y Elementos Desconocidos

Si se detectan rutas en el archivo KML, se informa al usuario de que no serán procesadas debido a las limitaciones de la aplicación. Los elementos cuya ciudad no puede ser determinada se almacenan temporalmente y se solicita al usuario que proporcione la ciudad a través de un formulario adicional.

Solicitud de Ciudad al Usuario

En caso de que la ciudad no esté disponible en el archivo KML, se solicita al usuario que ingrese esta información manualmente a través de un formulario HTML adicional.

Guardado de la Ciudad

Una vez que el usuario proporciona la ciudad, se utiliza la función `guardarCiudad()` para insertar los elementos que no pudieron ser procesados anteriormente debido a la falta de información sobre la ciudad.

5.5. Exportación a KML

En esta sección se detalla el proceso de exportación de puntos seleccionados a un archivo KML desde la aplicación web. La funcionalidad permite al usuario interactuar con marcadores en un mapa, seleccionar puntos mediante clics controlados y exportar estos puntos como un archivo KML que contiene información geoespacial estructurada.

5.5.1. Selección de Puntos en el Mapa

El proceso comienza con la visualización de puntos geográficos en el mapa. Cada punto está representado por un marcador interactivo que puede ser clicado por el usuario. Al hacer clic en un marcador mientras se mantiene presionada la tecla `CTRL`, se selecciona o deselecciona el punto correspondiente. Esto se logra mediante la función JavaScript `mostrarPuntosEnMapa()` que gestiona la visualización de marcadores y su interacción con los usuarios.

5.5.2. Interfaz de Selección de Puntos

Se muestra una ventana de selección que inicialmente está oculta y se activa al seleccionar al menos un punto en el mapa. En esta ventana, se listan los puntos seleccionados junto con botones para exportar los puntos seleccionados como KML o cancelar la operación. La lista de puntos seleccionados se actualiza dinámicamente mediante la función `actualizarListaSeleccionados()`.

5.5.3. Exportación a KML

Cuando el usuario decide exportar los puntos seleccionados, se genera un archivo KML que contiene los nombres y las coordenadas de los puntos seleccionados. Esto se realiza con la función `exportarSeleccionados()`, que crea dinámicamente el contenido XML del archivo KML basado en los puntos almacenados en la lista de puntos seleccionados.

El contenido del archivo KML se estructura de manera que cada punto se representa como un elemento `<Placemark>`, dentro del documento `<Document>`, especificando su nombre y coordenadas geográficas.

5.5.4. Cancelación y Restauración de Estado

El usuario tiene la opción de cancelar la operación de exportación, lo cual restaura los marcadores a su estado original (color y estado de selección). Esto se logra mediante la función `cancelarSeleccionado()`, que restaura los iconos de los marcadores y vacía la lista de puntos seleccionados.

5.5.5. Implementación en JavaScript

La funcionalidad descrita se implementa utilizando JavaScript y Leaflet. Los principales *scripts* involucrados incluyen:

- `marcadoresPersonalizados.js`: Define los iconos personalizados para los marcadores.
- `mapa.js`: Configura y maneja la visualización del mapa Leaflet.
- `favoritos.js`: Gestiona la lógica relacionada con los marcadores favoritos.
- `utilidades.js`: Contiene funciones útiles para el manejo de datos, en este caso, la operación del control clic esta en la función de `mostrarPuntosEnMapa()` y operaciones auxiliares.
- `eventos.js`: Maneja eventos del *zoom* y movimiento en el mapa.
- `busqueda.js`: Implementa la funcionalidad de búsqueda y filtrado de puntos en el mapa.
- `exportar.js`: Lógica específica para la exportación de puntos seleccionados a un archivo KML.

Estos *scripts* colaboran para crear una experiencia interactiva donde los usuarios pueden seleccionar, exportar y gestionar puntos geográficos de manera intuitiva desde el mapa interactivo.

5.6. Retos y Soluciones

5.6.1. Descripción

Durante la implementación se encontraron diversos retos técnicos que se detallarán a continuación junto con las soluciones adoptadas.

Desafíos con la Integración de Leaflet

- **Retos:** Inicialmente, se presentaron dificultades al integrar Leaflet con otros *scripts* personalizados para la gestión de marcadores y eventos en el mapa.
- **Solución:** Se abordó este reto mediante la modularización del código JavaScript, separando la lógica en diferentes archivos y funciones específicas, lo que permitió una mayor claridad y facilidad de mantenimiento del código.

Manejo de Eventos de Selección

- **Reto:** Detectar y manejar correctamente los eventos de clic en los marcadores, especialmente distinguiendo entre el clic para mover el mapa, el clic en una ubicación aleatoria para crear un marcador normal, y el CTRL+clic para seleccionar múltiples marcadores.
- **Solución:** Se utilizó la función `mostrarPuntosEnMapa()` para gestionar la lógica de eventos, asegurándose de que los marcadores pudieran cambiar de estado correctamente al ser seleccionados o deseleccionados, y que la lista de puntos seleccionados se actualizara dinámicamente. Se implementaron condiciones para diferenciar entre los distintos tipos de clic, garantizando que cada acción se manejara adecuadamente.

Exportación a KML

- **Reto:** Generar un archivo KML válido que pudiera ser descargado y utilizado en otras aplicaciones GIS.
- **Solución:** Se implementó la función `exportarSeleccionados()`, que construye el contenido del archivo KML en formato XML basado en los puntos seleccionados y genera un archivo descargable usando `Blob` y `URL.createObjectURL`.

Mostrar Puntos en el Mapa

Para mostrar los puntos en el mapa con los iconos correctos, es fundamental distinguir entre los diferentes tipos de puntos: propios, de otros usuarios, y favoritos. Esto permite una visualización clara y personalizada para el usuario.

El proceso comienza cargando la lista de puntos y recorriéndola para determinar las características de cada punto. Basado en si un punto es favorito, pertenece al usuario actual, o es de otro usuario, se asigna el icono correspondiente:

```
function mostrarPuntosEnMapa(puntos) {
  puntos.forEach(function(punto) {
    // Verifica si el punto es un favorito
    var esFavorito = favoritos.includes(punto.cId);
    var icono;

    // Determinar el icono adecuado según las características del punto
    if (punto.cPersonaId == userId && esFavorito) {
      // Icono para puntos del usuario actual que también son favoritos
      icono = iconoMisSitiosMG;
    } else if (punto.cPersonaId == userId) {
      // Icono para puntos del usuario actual que no son favoritos
      icono = iconoMisSitios;
    } else if (esFavorito) {
```

```
        // Icono para puntos de otros usuarios que son favoritos
        icono = iconoOtrosSitiosMG;
    } else {
        // Icono para puntos de otros usuarios que no son favoritos
        icono = iconoOtrosSitios;
    }

    // Crear y añadir el marcador al mapa con el icono seleccionado
    var marker = L.marker([punto.latitud, punto.long], { icon: icono })
    .addTo(markersLayer);
});
}
```

Este enfoque asegura que cada marcador en el mapa tenga el icono adecuado en función de sus características, mejorando la claridad y usabilidad del mapa para el usuario. Los iconos fueron creados utilizando Canva, lo que permitió diseñar distintivos visuales claros y diferenciados para cada tipo de punto.

Los iconos asignados a los marcadores son:)

- `iconoMisSitiosMG.png` (Figura 5.2): Icono para puntos del usuario actual que también son favoritos.
- `iconoMisSitios.png` (Figura 5.3): Icono para puntos del usuario actual que no son favoritos.
- `iconoOtrosSitiosMG.png` (Figura 5.4): Icono para puntos de otros usuarios que son favoritos.
- `iconoOtrosSitios.png` (Figura 5.5): Icono para puntos de otros usuarios que no son favoritos.

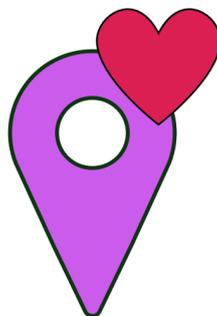


Figura 5.2: Icono para puntos del usuario favoritos (`iconoMisSitiosMG.png`).



Figura 5.3: Icono para puntos del usuario (`iconoMisSitios.png`).



Figura 5.4: Icono para puntos de otros usuarios favoritos (`iconoOtrosSitiosMG.png`).



Figura 5.5: Icono para puntos de otros usuarios (`iconoOtrosSitios.png`).

5.7. Conclusiones

5.7.1. Resumen

Se resumirán brevemente los puntos clave de la implementación.

- La integración de Leaflet proporcionó una plataforma robusta para la visualización y manipulación de datos geoespaciales.
- La modularización del código permitió una mejor gestión y mantenimiento de la lógica de la aplicación.

- La funcionalidad de selección y exportación de puntos a KML añadió un valor significativo, permitiendo a los usuarios trabajar con datos geoespaciales de manera eficiente.
- La implementación de eventos y la gestión dinámica de marcadores mejoraron la interactividad y la experiencia del usuario.

5.7.2. Lecciones aprendidas

- **Modularización del Código:** La separación de la lógica en módulos específicos facilita la gestión y el mantenimiento del código a largo plazo.
- **Gestión de Eventos Complejos:** La importancia de manejar correctamente los eventos de usuario para asegurar una experiencia interactiva y sin errores.
- **Exportación de Datos:** Comprender cómo generar y manipular archivos en formato KML para la exportación de datos geoespaciales.
- **Compatibilidad de Navegadores:** La necesidad de realizar pruebas en múltiples navegadores y dispositivos para asegurar una experiencia de usuario consistente.

Capítulo 6

Pruebas

En el ámbito de la ingeniería de software y la garantía de calidad, las pruebas juegan un rol esencial. Son cruciales para asegurar que el sistema cumpla con los requisitos especificados, funcione correctamente bajo diversas condiciones y esté libre de fallos críticos. Dos enfoques ampliamente utilizados para llevar a cabo estas pruebas son las pruebas de caja negra y las pruebas de caja blanca.

Las pruebas de caja negra se enfocan en la funcionalidad del software sin considerar su estructura interna. Este tipo de pruebas verifica que las entradas proporcionen las salidas esperadas, evaluando el comportamiento del sistema desde la perspectiva del usuario final. Es especialmente útil para identificar problemas de usabilidad, errores en la interfaz y problemas de integración.

Por otro lado, las pruebas de caja blanca examinan la estructura interna del código. Este enfoque permite a los desarrolladores verificar que todas las rutas posibles del código se ejecuten correctamente, asegurando la coherencia lógica del software. Las pruebas de caja blanca son fundamentales para detectar errores en el flujo de control, problemas de seguridad y otros defectos a nivel de implementación.

En esta sección, se describen las pruebas realizadas tanto de caja negra como de caja blanca para garantizar la calidad y fiabilidad del sistema desarrollado. A continuación, se detallan las pruebas caja blanca llevadas a cabo 6.1, seguidas de las pruebas de caja negra 6.2.

6.1. Pruebas de caja blanca

- **Prueba de ejecución completa del código:** Se verifica que todas las líneas de código se ejecuten al menos una vez durante las pruebas para asegurar que no haya segmentos inalcanzables.
- **Prueba de conexión con la base de datos:** Se verifica que la aplicación establezca correctamente la conexión con la base de datos configurada, permitiendo operaciones de lectura y escritura sin errores.

- **Prueba de inserción de datos:** Se comprueba que los datos introducidos desde los formularios de la aplicación se guarden adecuadamente en la base de datos según el diseño establecido.
- **Prueba de eliminación de datos:** Se asegura que los datos eliminados desde la interfaz de usuario sean eliminados correctamente de la base de datos sin dejar datos residuales.
- **Prueba de actualización de datos:** Se verifica que los cambios realizados en los datos desde la interfaz de usuario se reflejen correctamente en la base de datos, asegurando la consistencia de los datos.
- **Prueba de consultas a la base de datos:** Se comprueba que las consultas realizadas desde la aplicación retornen los datos esperados de la base de datos según los criterios de búsqueda especificados.
- **Prueba de manejo de errores:** Se verifica que los errores generados por la aplicación se muestren adecuadamente al usuario final, proporcionando mensajes claros sobre el tipo de error ocurrido y sugerencias para su corrección.
- **Prueba de seguridad contra inyección XSS:** Se comprueba que la aplicación esté protegida contra ataques de inyección de código JavaScript (XSS, *Cross-Site Scripting*), asegurando que no sea posible inyectar código malicioso a través de campos de entrada.
- **Prueba de seguridad contra inyección SQL:** Se verifica que la aplicación esté protegida contra ataques de inyección SQL, garantizando que no sea posible ejecutar comandos SQL maliciosos desde la interfaz de usuario.

6.2. Pruebas de caja negra

P1: Inicio de sesión de usuarios

Propósito	Permitir a los usuarios iniciar sesión en la aplicación.
Prerequisitos	El usuario debe estar registrado previamente.
Datos de entrada	<ul style="list-style-type: none"> ▪ Credenciales de usuario: email y contraseña.
Procedimiento	<ol style="list-style-type: none"> 1. Acceder a la página de inicio de sesión. 2. Introducir email y contraseña válidos. 3. Pulsar el botón de “Iniciar sesión”.
Resultado esperado	El usuario es autenticado y redirigido a su página principal.
Resultado obtenido	Resultado esperado

Cuadro 6.1: Prueba de inicio de sesión de usuarios

P2: Creación de puntos de interés

Propósito	Permitir a los usuarios crear nuevos puntos de interés en el mapa.
Prerequisitos	El usuario debe estar autenticado y en el mapa de “Explorar” o de “Mis sitios”.
Datos de entrada	<ul style="list-style-type: none"> ▪ Datos del punto de interés: nombre, descripción, coordenadas.

Procedimiento	<ol style="list-style-type: none"> 1. Hacer clic sobre un punto en el mapa. 2. Hacer clic sobre “Crear punto nuevo aquí”. 3. Introducir los datos requeridos. 4. Pulsar el botón de “Crear”.
Resultado esperado	El punto de interés se crea y se muestra en el mapa.
Resultado obtenido	Resultado esperado

Cuadro 6.2: Prueba de creación de puntos de interés

P3: Importación de puntos de interés

Propósito	Permitir la importación de puntos de interés desde un archivo KML.
Prerequisitos	El usuario debe estar autenticado y en la sección de importación de puntos de interés.
Datos de entrada	<ul style="list-style-type: none"> ■ Archivo KML válido.
Procedimiento	<ol style="list-style-type: none"> 1. Acceder a la sección de importación de puntos de interés. 2. Seleccionar y cargar el archivo KML. 3. Pulsar el botón de “Importar”.
Resultado esperado	Los puntos de interés del archivo KML se importan y se muestran en el mapa.

Resultado obtenido	Resultado esperado
---------------------------	--------------------

Cuadro 6.3: Prueba de importación de puntos de interés

P4: Exportación de puntos de interés

Propósito	Permitir la exportación de puntos de interés a un archivo KML.
Prerequisitos	El usuario debe estar autenticado y en el mapa de “Explorar” o de “Mis sitios”.
Datos de entrada	<ul style="list-style-type: none"> ▪ Selección de puntos de interés a exportar.
Procedimiento	<ol style="list-style-type: none"> 1. Seleccionar los puntos de interés a exportar. 2. Pulsar el botón de “Exportar”.
Resultado esperado	Se genera y descarga un archivo KML con los puntos de interés seleccionados.
Resultado obtenido	Resultado esperado

Cuadro 6.4: Prueba de exportación de puntos de interés

P5: Modificación de puntos de interés

Propósito	Permitir a los usuarios modificar sus puntos de interés existentes.
Prerequisitos	El usuario debe estar autenticado y en la sección de detalle del punto de interés.

Datos de entrada	<ul style="list-style-type: none"> ▪ Datos del punto de interés a modificar: nombre, descripción, coordenadas.
Procedimiento	<ol style="list-style-type: none"> 1. Seleccionar el punto de interés a modificar. 2. Acceder a la página de detalle de ese punto. 3. Pulsar en “Modificar datos”. 4. Actualizar los datos requeridos. 5. Pulsar el botón de “Guardar”.
Resultado esperado	El punto de interés se actualiza con los nuevos datos y se refleja en el mapa.
Resultado obtenido	Resultado esperado

Cuadro 6.5: Prueba de modificación de puntos de interés

P6: Eliminación de puntos de interés

Propósito	Permitir a los usuarios eliminar sus puntos de interés existentes.
Prerequisitos	El usuario debe estar autenticado y en la sección de detalle del punto de interés.
Datos de entrada	<ul style="list-style-type: none"> ▪ Selección del punto de interés a eliminar.

Procedimiento	<ol style="list-style-type: none"> 1. Seleccionar el punto de interés a modificar. 2. Acceder a la página de detalle de ese punto. 3. Pulsar el botón de “Eliminar”. 4. Confirmar la eliminación.
Resultado esperado	El punto de interés se elimina y ya no se muestra en el mapa.
Resultado obtenido	Resultado esperado

Cuadro 6.6: Prueba de eliminación de puntos de interés

P7: Redirección al *login* tras expiración de sesión

Propósito	Redirigir al usuario al <i>login</i> cuando la sesión haya expirado.
Prerequisitos	La sesión del usuario debe haber expirado.
Datos de entrada	<ul style="list-style-type: none"> ▪ Intento de realizar una acción que requiera autenticación. ▪ Recargar la página.
Procedimiento	<ol style="list-style-type: none"> 1. Intentar realizar cualquier acción que requiera el ID de usuario. 2. Recargar la página.
Resultado esperado	El usuario es redirigido a la página de <i>login</i> .

Resultado obtenido	Resultado esperado
---------------------------	--------------------

Cuadro 6.7: Prueba de redirección al *login* tras expiración de sesión

P8: Redirección al inicio tras URL inválida o no autorizada

Propósito	Redirigir al usuario al inicio cuando acceda a una URL inválida o no autorizada.
Prerequisitos	Intento de acceder a una URL no válida o no autorizada.
Datos de entrada	<ul style="list-style-type: none"> ▪ URL no válida o no autorizada.
Procedimiento	<ol style="list-style-type: none"> 1. Intentar modificar los parámetros de la URL para acceder a recursos no autorizados o inexistentes.
Resultado esperado	El usuario es redirigido a la página de inicio como si acabase de hacer el <i>login</i> .
Resultado obtenido	Resultado esperado

Cuadro 6.8: Prueba de redirección al inicio tras URL inválida o no autorizada

Capítulo 7

Conclusiones y trabajo futuro

7.1. Conclusiones

Mis conclusiones personales sobre la investigación realizada y el trabajo desarrollado en este proyecto abarcan múltiples aspectos de la gestión de proyectos y la implementación de herramientas de planificación. A lo largo del proyecto, he observado la importancia de una correcta planificación y gestión de los recursos, lo que ha permitido optimizar los procesos y mejorar los resultados obtenidos. Las metodologías ágiles se han mostrado efectivas en la gestión del cambio y la adaptación a nuevas circunstancias, facilitando una mayor flexibilidad y rapidez en la toma de decisiones.

Al inicio del proyecto, desconocía hasta qué punto podría desarrollar funcionalidades, ya que no tenía plena conciencia de las capacidades y la utilidad del lenguaje `JavaScript`. Sin embargo, a medida que avanzaba en el desarrollo, descubrí la amplitud de aplicaciones y soluciones que pueden implementarse con este lenguaje. `JavaScript` se ha manifestado como una herramienta poderosa y versátil, capaz de abordar una amplia gama de necesidades de programación, desde la manipulación del DOM (*DocumentObject Model*) hasta la creación de aplicaciones web dinámicas y complejas. Esta realización ha ampliado significativamente mi perspectiva y confianza en el uso de `JavaScript`, permitiéndome explotar su potencial al máximo para cumplir con los objetivos del proyecto.

Otro aspecto destacado es la integración de herramientas tecnológicas que han permitido una mejor visualización y seguimiento del progreso del proyecto. Además, poseo nuevos conocimientos sobre herramientas como *Leaflet* y la creación de documentos en formatos como `.kml`, lo que ha enriquecido mi conjunto de habilidades y mi capacidad para abordar futuros proyectos con mayor eficacia y competencia.

7.2. Trabajo futuro

En futuros desarrollos del proyecto, se identifican diversas áreas de mejora y ampliación de funcionalidades que podrían enriquecer la experiencia del usuario y aumentar la utilidad de la aplicación. Estas áreas incluyen:

- **Incorporación de rutas:** Se propone la integración de funcionalidades que permitan a los usuarios incorporar rutas personalizadas. Esta característica facilitaría la planificación de itinerarios y la navegación a través de puntos de interés, mejorando significativamente la usabilidad de la aplicación. Además, permitiría a los usuarios tener una visión más completa y organizada de su recorrido, optimizando así su experiencia de viaje.
- **Adición de etiquetas a los puntos de interés:** La adición de etiquetas contribuiría a una mejor categorización y organización de la información. Las etiquetas permitirían a los usuarios filtrar y buscar puntos de interés de manera más eficiente, mejorando la accesibilidad y la personalización de la información presentada. Esto también podría facilitar la gestión y actualización de los datos dentro de la aplicación.
- **Verificación y control de fotos y comentarios:** Implementar un sistema para verificar y controlar las fotos y comentarios asociados con los puntos de interés más conocidos garantizaría la calidad y relevancia del contenido generado por los usuarios. La verificación podría incluir algoritmos de moderación automática y la posibilidad de reportar contenido inapropiado, asegurando así un entorno de información confiable y útil.
- **Sección para compartir experiencias:** Se sugiere la creación de una sección dedicada a compartir experiencias, donde los usuarios puedan publicar relatos y recomendaciones sobre sus visitas a los puntos de interés. Esta sección fomentaría la interacción y la comunidad entre los usuarios, proporcionando una fuente valiosa de información y consejos prácticos basados en experiencias reales. La implementación de esta funcionalidad podría incluir opciones para valorar y comentar las experiencias compartidas, promoviendo un intercambio dinámico y enriquecedor de información.

Estas propuestas para futuros desarrollos subrayan el compromiso continuo con la mejora de la aplicación y la optimización de la experiencia del usuario. La incorporación de estas nuevas funcionalidades no solo amplía las capacidades actuales del proyecto, sino que también abre nuevas posibilidades para la innovación y el crecimiento.

Parte III

Manuales de la Aplicación

Capítulo 8

Manual de Usuario

Este capítulo final se dedica a desarrollar el manual del usuario final, así como del usuario administrador, si fuese necesario.

8.1. Manual de usuario

Este manual está diseñado para guiar a los usuarios registrados con puntos de interés propios y aquellos sin puntos de interés propios en el uso de la plataforma web.

8.1.1. Introducción

La plataforma web permite a los usuarios gestionar puntos de interés turísticos en España. Los usuarios pueden registrarse, iniciar sesión, explorar puntos de interés, crear nuevos puntos, y mucho más. Este manual cubrirá las funcionalidades disponibles para diferentes tipos de usuarios.

8.1.2. Usuario registrado con puntos de interés propios

Inicio de sesión y registro

- **Registro:** En la página de inicio, haz clic en “Registro”. Completa el formulario con tu nombre, correo electrónico y contraseña. Una vez registrado, podrás iniciar sesión.



Registro de Personas

Nombre

E-mail

Contraseña

Confirmar Contraseña

Registrar

¿Ya tienes una cuenta? Iniciar sesión

Figura 8.1: Captura de la aplicación Web: Registro

- **Inicio de sesión:** Introduce tu correo electrónico y contraseña en la página de inicio de sesión y haz clic en “Iniciar sesión”.



Iniciar Sesión

Correo Electrónico:

Contraseña:

Iniciar Sesión

¿No tienes una cuenta? Regístrate

Figura 8.2: Captura de la aplicación Web: Inicio de sesión o *login*

Página principal: Explorar

- **Mapa:** Al iniciar sesión, serás dirigido a la página principal “Explorar”. Aquí verás un mapa con todos los puntos de interés cercanos a tu ubicación actual.

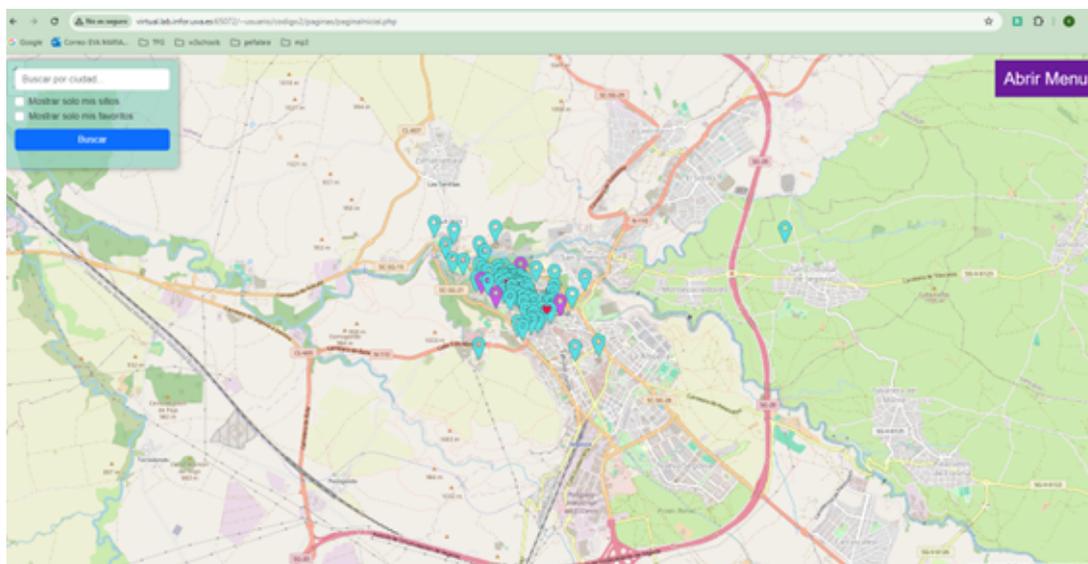


Figura 8.3: Captura de la aplicación Web: Mapa Explorar

- **Buscador:** Utiliza el buscador para buscar por ciudad y filtrar por “Mis sitios”, “Mis favoritos”, o “Otros sitios”.
 - **Mis sitios:** Tus puntos de interés se mostrarán con un marcador morado.
 - **Otros sitios:** Los puntos de otros usuarios se mostrarán en azul.
 - **Favoritos:** Los favoritos tendrán un corazón en la esquina superior derecha del marcador.

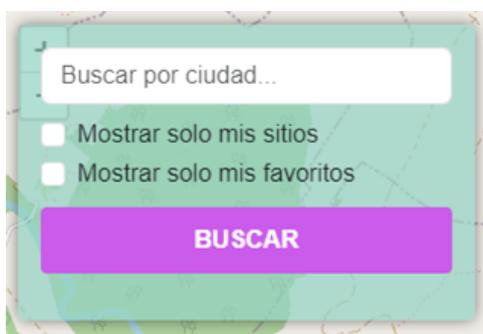


Figura 8.4: Captura de la aplicación Web: Buscador

Crear un punto de interés

- **Añadir nuevo punto:** Haz clic en cualquier parte del mapa para crear un nuevo punto de interés. Serás redirigido a una página donde puedes completar la información del punto de interés, incluyendo nombre y descripción (opcional). Haz clic en “Añadir” para guardar.

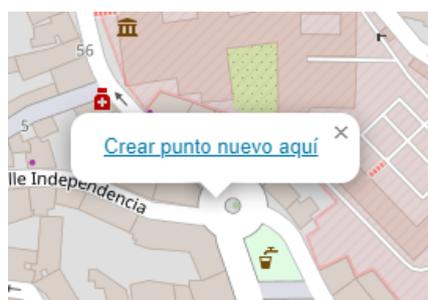


Figura 8.5: Captura de la aplicación Web: Clic sobre el mapa

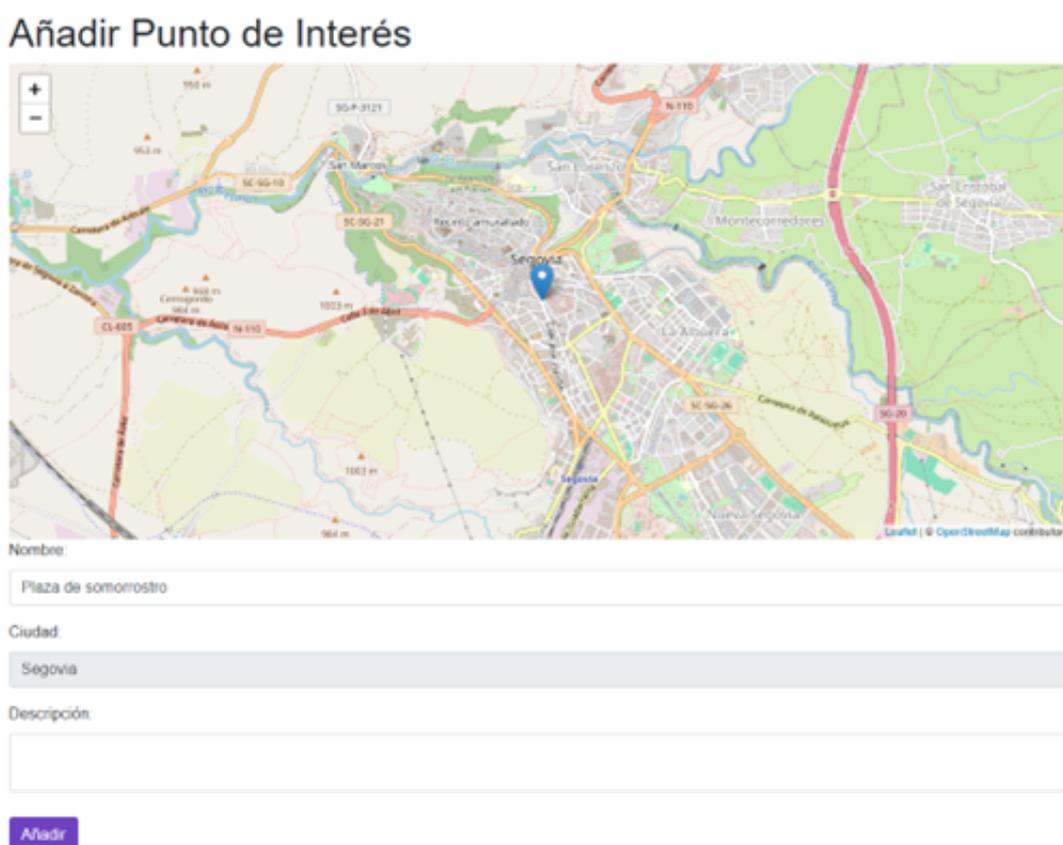


Figura 8.6: Captura de la aplicación Web: Página de agregar punto

Gestionar puntos de interés

- **Ver detalles:** Haz clic en un marcador en el mapa para ver una ventana emergente con el nombre del punto. Haz clic en el nombre para ir a la página de detalles del punto de interés.

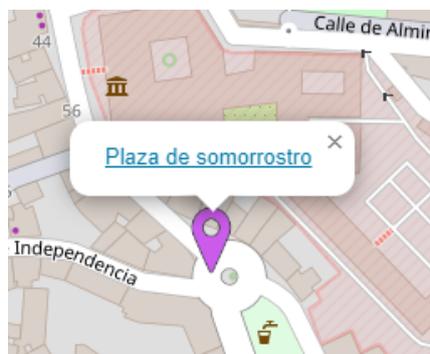


Figura 8.7: Captura de la aplicación Web: Clic sobre un punto

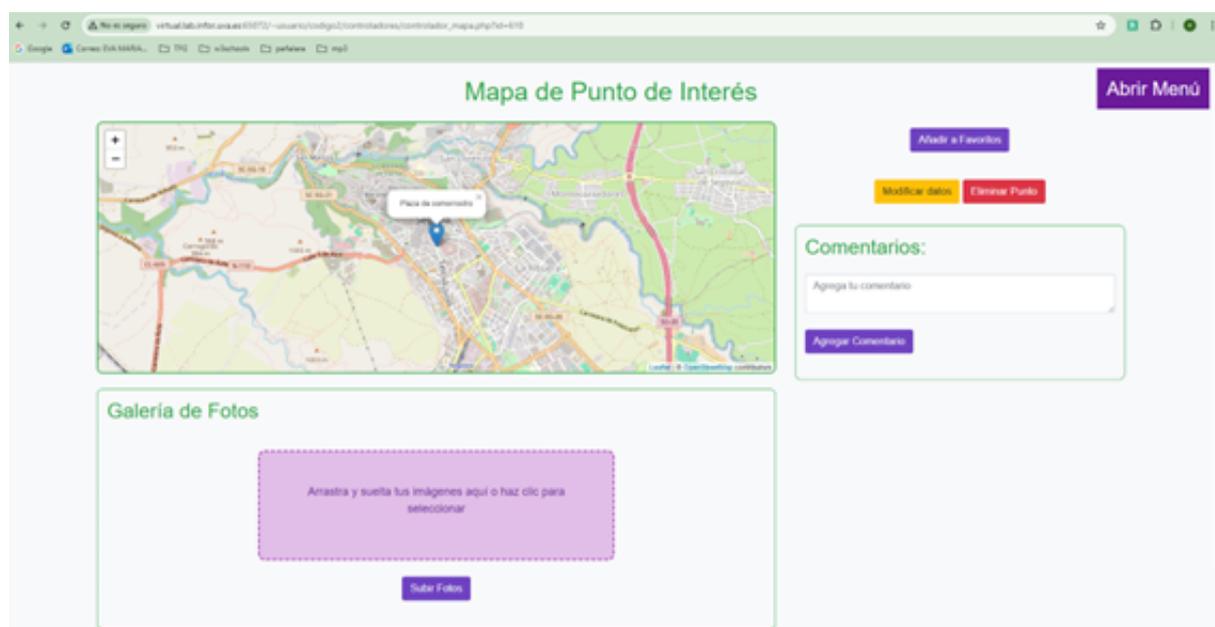


Figura 8.8: Captura de la aplicación Web: Página Punto individual

- **Editar/eliminar:** Si eres el propietario del punto de interés, en la página de detalles podrás editar los datos o eliminar el punto.

Interacción con puntos de interés

- **Comentarios y fotos:** En la página de detalles de un punto de interés, puedes añadir comentarios y fotos.
- **Añadir a favoritos:** Haz clic en el botón de favoritos para añadir el punto a tu lista de favoritos.



Figura 8.9: Captura de la aplicación Web: Pulsar sobre “añadir a favoritos”



Figura 8.10: Captura de la aplicación Web: Visualización del punto como favorito

Exportar puntos de interés

- **Selección múltiple:** Mantén presionada la tecla Control (Ctrl) y haz clic en varios puntos en el mapa para seleccionarlos. Aparecerán en una lista sobre el mapa.
- **Exportar:** Haz clic en “Exportar” para crear un archivo KML con los puntos seleccionados.

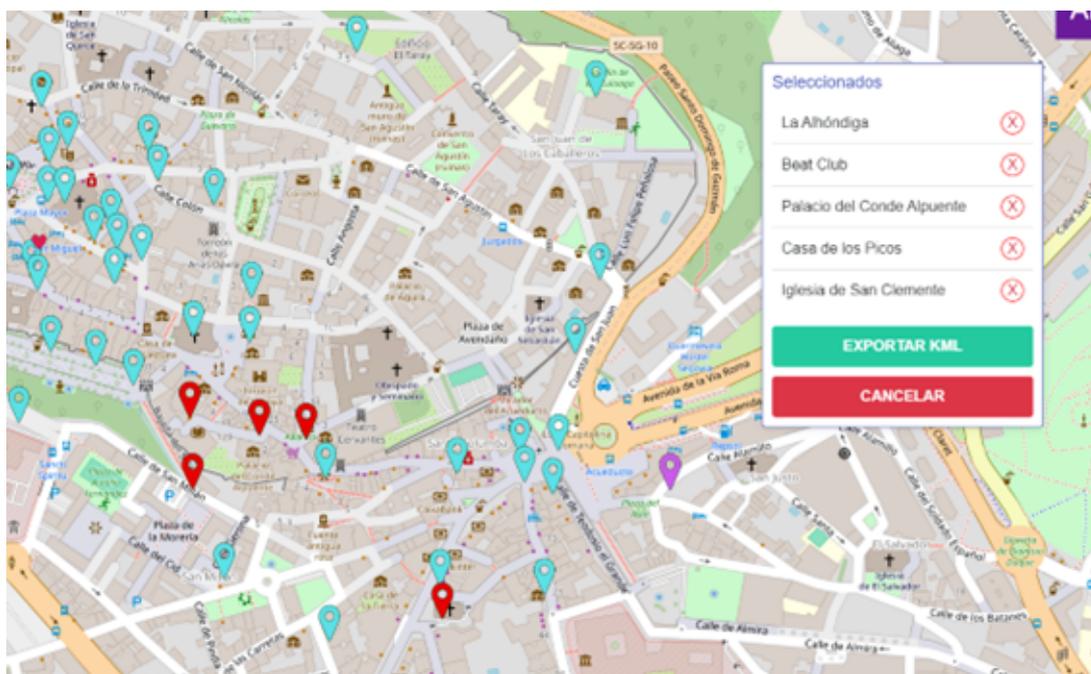


Figura 8.11: Captura de la aplicación Web: Exportar a KML

Otras funcionalidades

- **Mis sitios:** Esta página muestra solo tus puntos de interés.

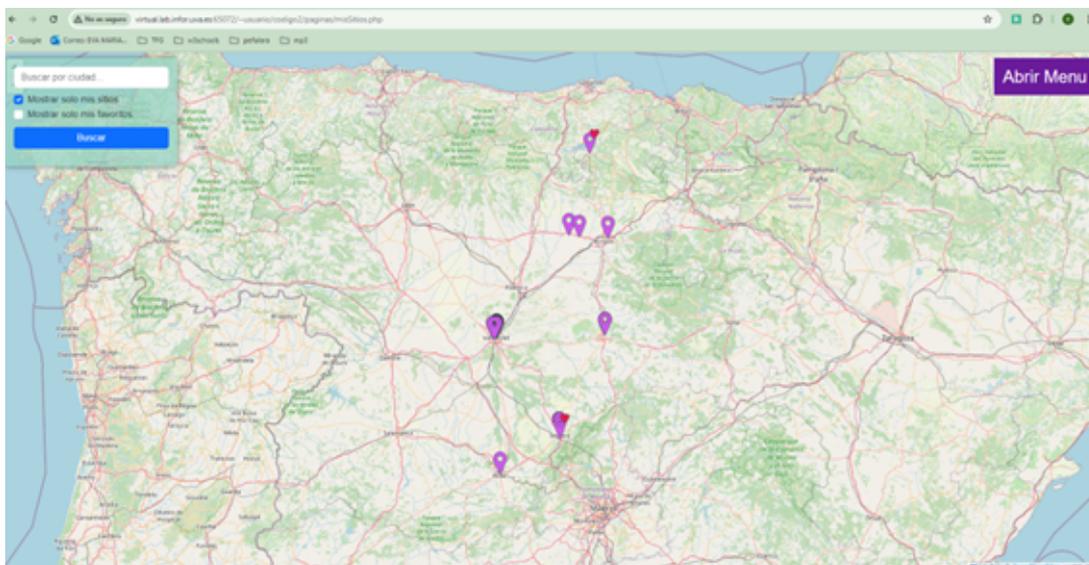


Figura 8.12: Captura de la aplicación Web: Mis sitios

- **Subir KML:** Usa esta página para subir un archivo KML y cargar los datos en la aplicación.

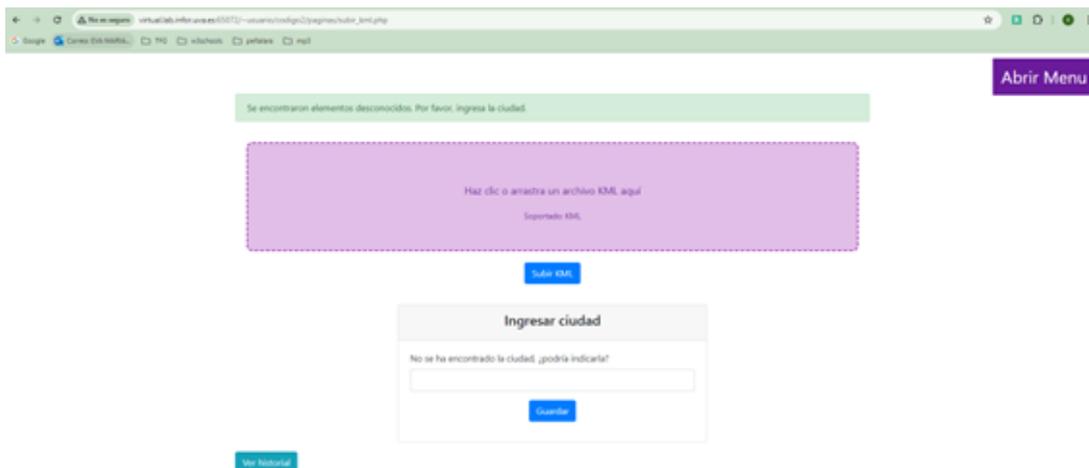


Figura 8.13: Captura de la aplicación Web: Subir KML

8.1.3. Usuario registrado sin puntos de interés propios

Diferencias

- **Comentarios y favoritos:** Puedes añadir comentarios y marcar puntos de interés como favoritos.
- **Modificar puntos de interés:** No puedes modificar ni eliminar el sitio de interés si no eres el usuario que lo creó.

8.2. Manual de administración

Este manual está diseñado para guiar al administrador de la plataforma en la gestión avanzada de la aplicación.

8.2.1. Introducción

El administrador tiene acceso completo a todas las funcionalidades y datos de la aplicación, incluyendo la gestión de usuarios, puntos de interés, comentarios, imágenes, y rutas.

8.2.2. Funcionalidades del administrador

Gestión de Usuarios

- **Ver usuarios:** El administrador puede ver la lista completa de usuarios registrados en la plataforma.
- **Modificar/eliminar usuarios:** El administrador tiene la capacidad de modificar la información de cualquier usuario y eliminar cuentas si es necesario.

Base de Datos

- **Acceso directo:** Dependiendo de las necesidades, el administrador puede tener acceso directo a la base de datos para realizar tareas avanzadas como consultas, modificaciones y mantenimiento.

Seguridad y mantenimiento

- **Monitoreo:** El administrador debe monitorear regularmente la actividad en la plataforma para asegurar el buen funcionamiento y la seguridad de los datos.
- **Actualizaciones:** Es responsabilidad del administrador realizar actualizaciones de software y mantener la plataforma segura contra posibles vulnerabilidades.

Parte IV

Apéndices

Apéndice A

Anexos

A.1. Información complementaria

A.1.1. Definición de siglas y abreviaturas

A continuación se presenta una lista de siglas y abreviaturas utilizadas en esta memoria:

Sigla	Significado
HTML	<i>HyperText Markup Language</i>
CSS	<i>Cascading Style Sheets</i>
PHP	<i>Hypertext Preprocessor</i>
API	<i>Application Programming Interface</i>
MySQL	<i>My Structured Query Language</i>
KML	<i>Keyhole Markup Language</i>
WCAG	<i>Web Content Accessibility Guidelines</i>
CI	<i>Continuous Integration</i>
CD	<i>Continuous Deployment</i>
XSS	<i>Cross-Site Scripting</i>

Cuadro A.1: Tabla de siglas y abreviaturas.

Bibliografía y Webgrafía

- [1] Programación 365. *Programación orientada a objetos vs estructurada*. Accedido: 2024-07-14. URL: <https://programacion365.com/programacion-orientada-a-objetos-vs-estructurada/>.
- [2] Len Bass, Paul Clements y Rick Kazman. *Software Architecture in Practice*. 3rd. Upper Saddle River, NJ, USA: Addison-Wesley Professional, 2012. ISBN: 978-0321815736.
- [3] COMPARASOFTWARE. *¿Qué es y Cómo trabajar en el Modelo Incremental?* Accedido: 2024-03-25. 2024. URL: <https://blog.comparasoftware.com/que-es-el-modelo-incremental/>.
- [4] Perla Velasco Elizondo. *Arquitectura de Software: Conceptos y Ciclo de Desarrollo*. https://www.researchgate.net/profile/Perla-Velasco-Elizondo/publication/281137715_Arquitectura_de_Software_Conceptos_y_Ciclo_de_Desarrollo/links/57144e1408aeebe07c0641ab/Arquitectura-de-Software-Conceptos-y-Ciclo-de-Desarrollo.pdf. Accedido: 2024-07-02. 2015.
- [5] Portal Oficial de Turismo de España. *Portal Oficial de Turismo de España*. Accedido: 2024-05-02. 2024. URL: <https://www.spain.info/es/>.
- [6] Instituto Nacional de Estadística (INE). *Estadísticas del turismo en España*. https://www.ine.es/dyngs/INEbase/es/categoria.htm?c=Estadistica_P&cid=1254735576863. Accedido: 2024-07-02. 2019.
- [7] Outdooractive GmbH. *Outdooractive - Hiking, Cycling, MTB, Running*. Accedido: 2024-05-02. 2024. URL: <https://www.outdooractive.com/en/>.
- [8] Agustín J González y Orientados a Objeto. “Ingeniería de software: Metodologías”. En: *Visitado el 10/04/2024* 8 (2007).
- [9] Google. *Google Maps API Documentation*. Accedido: 2024-07-02. 2024. URL: <https://developers.google.com/maps/documentation>.
- [10] Guru99. *Modelo incremental en SDLC: uso, ventajas y desventajas*. Accedido: 2024-04-15. 2024. URL: <https://www.guru99.com/es/what-is-incremental-model-in-sdlc-advantages-disadvantages.html>.
- [11] H. Küçükaydin, E. Karasakal y G. Yıldız. “Route planning in tourist destinations using optimization techniques”. En: *Tourism Management* 56 (2016), págs. 28-43.

- [12] Craig Larman. *Agile and Iterative Development: A Manager's Guide*. Addison-Wesley, 2004.
- [13] Leaflet. *Leaflet API Documentation*. Accedido: 2024-07-02. 2024. URL: <https://leafletjs.com/reference-1.7.1.html>.
- [14] P. A. Longley et al. *Geospatial Analysis: A Comprehensive Guide*. 5th. Winchelsea, UK: The Winchelsea Press, 2015.
- [15] OpenWebinars. *Arquitectura de Software: Qué es y qué tipos existen*. <https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>. Accedido: 2024-07-02. 2024.
- [16] Philip L. Pearce. *Tourist Behaviour: Themes and Conceptual Schemes*. Clevedon, UK: Channel View Publications, 2005.
- [17] PostGIS. *PostGIS Documentation*. Accedido: 2024-07-02. 2024. URL: <https://postgis.net/documentation/>.
- [18] Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. McGraw-Hill, 2010.
- [19] RETO. *Pruebas de Caja Negra vs Pruebas de Caja Blanca: Diferencias*. Accedido: 2024-07-12. 2024. URL: <https://reto.com.mx/pruebas-de-caja-negra-vs-pruebas-de-caja-blanca-difere/>.
- [20] OBS Business School. *Características y fases del modelo incremental*. Accedido: 2024-04-10. 2024. URL: <https://www.obsbusiness.school/blog/caracteristicas-y-fases-del-modelo-incremental/>.
- [21] SciELO. *Aplicación de la metodología incremental en el desarrollo de ...*. Accedido: 2024-04-05. 2024. URL: https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-07642016000200014.
- [22] Wikiloc Outdoor SL. *Wikiloc - Trails of the World*. Accedido: 2024-05-02. 2024. URL: <https://www.wikiloc.com/>.
- [23] Ian Sommerville. *Software Engineering*. Pearson Education, 2011.
- [24] TripAdvisor. *TripAdvisor*. Accedido: 2024-07-02. 2024. URL: <https://www.tripadvisor.com/>.
- [25] Santander Universidades. *Metodologías de desarrollo de software*. Accedido: 2024-07-14. 2020. URL: <https://www.santanderopenacademy.com/es/blog/metodologias-desarrollo-software.html>.
- [26] Wikipedia. *Black-box testing*. Accedido: 2024-07-12. 2024. URL: https://en.wikipedia.org/wiki/Black-box_testing.
- [27] Wikipedia contributors. *Georeferencing*. Accedido: 2024-07-02. 2024. URL: <https://en.wikipedia.org/wiki/Georeferencing>.
- [28] Wikipedia contributors. *Point of Interest*. Accedido: 2024-07-02. 2024. URL: https://en.wikipedia.org/wiki/Point_of_interest.

- [29] Wikipedia contributors. *Route*. Accedido: 2024-07-02. 2024. URL: <https://en.wikipedia.org/wiki/Route>.
- [30] Yelp. *Yelp*. Accedido: 2024-07-02. 2024. URL: <https://www.yelp.com/>.