

Universidad de Valladolid

E.T.S. DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

APLICACIÓN DE TURISMO DE VALLADOLID

AUTOR: Raquel González Suárez

TUTORES: Cesar Vaca Rodríguez

Mónica Martín García

Agradecimientos:

*A mis padres, por darme la oportunidad de llegar hasta aquí
y por su constante apoyo.*

*A mi hermano y Javi, por apoyarme y soportar mis malos
momentos.*

*A mis compañeros y amigos, por todos los buenos
momentos vividos durante estos años.*

RESUMEN

El objetivo de este proyecto es el desarrollo de una aplicación Android que proporcione información sobre las principales atracciones turísticas de la ciudad de Valladolid.

La aplicación permitirá al usuario interactuar con la aplicación para adquirir la información que éste precise y de esta forma facilitarle la visita turística a la ciudad.

Esta aplicación funciona en dispositivos con el sistema operativo Android desde la versión 3 hasta la versión 5, y última por el momento. Está desarrollada para dispositivos de varios tamaños.

Su uso será público, pudiendo ser utilizado por cualquier usuario que esté interesado en información turística de la ciudad.

Índice

1. INTRODUCCIÓN	1
1.1. Visión general	3
1.2. Objetivos	3
1.3. Motivaciones personales	4
1.5 Estructura del documento	4
2. CONTEXTO TECNOLÓGICO	6
2.1. Introducción	8
2.2. Contexto de Desarrollo	8
2.2.1. <i>Android</i>	8
2.2.1. <i>Servicios Web REST</i>	10
2.3. Lenguajes de programación	12
2.3.1. <i>Java</i>	12
2.3.2 <i>XML</i>	12
2.3.3. <i>JSON</i>	13
2.3.4. <i>Xpath</i>	13
2.3.5. <i>UML</i>	13
2.4. Herramientas.....	13
2.4.1. <i>Android SDK</i>	13
2.4.2. <i>Android Studio</i>	14
2.4.3. <i>Eclipse Web Tools Platforms</i>	14
2.4.4. <i>MySQL WorkBench</i>	14
2.4.5. <i>Astah Community</i>	15
2.4.7. <i>Dropbox</i>	15
3. GESTIÓN DEL PROYECTO.....	16
3.1. Introducción	18
3.2. Modelo de Proceso	18
3.3. Gestión de Riesgos	19
3.4. Planificación	21

3.5. Recursos	24
3.5.1. Recursos hardware	24
3.5.2. Recursos software	25
3.5.3. Recursos humanos	25
3.6. Costes y presupuesto	25
3.6.1. Presupuesto final	26
4. ANÁLISIS DEL SISTEMA	27
4.1 Introducción	29
4.2 Especificación de requisitos	29
4.2.1. Requisitos funcionales	29
4.2.2. Requisitos no funcionales	30
4.3. Modelado de Casos de Uso	31
4.3.1. Identificación de los actores	31
4.3.2. Casos de Uso	31
4.4. Matriz relación requisitos-casos de uso	41
4.5. Modelo de dominio	41
4.6. Prototipos	42
5. DISEÑO DEL SISTEMA	46
5.1. Introducción	48
5.2. Decisiones de diseño	48
5.2.1. Topología del sistema	49
5.2.3. Persistencia de datos	51
5.3. Patrones de Diseño	52
5.3.1. MVC	52
5.4. Interfaz de usuario	54
5.4.1. Adaptación al tamaño de pantalla	54
5.5. Diagrama de paquetes	56
5.6. Diagramas de secuencia de diseño	57
5.7. Diagrama de clases de diseño	71
6. IMPLEMENTACIÓN	76
6.1. Introducción	77

6.2. Librerías y APIS utilizadas	77
6.2.2. Servicio Web	77
6.2.3. Aplicación Android.....	78
6.3. Soporte en varios idiomas.....	79
6.4. Seguridad en Android.....	79
7. PRUEBAS	81
7.1. Introducción	82
7.2. Casos de prueba	82
7.3. Matriz Casos de uso- Casos de prueba	88
8. CONCLUSIONES Y TRABAJOS FUTUROS.....	89
8.1. Conclusiones.....	90
8.2. Trabajo futuros.....	90
ANEXOS.....	92
A: Manual de Usuario.....	93
A.1. Lugares de interés y rutas	94
A.2. Mapas	98
A.3. Otra información de interés	99
A.4. Redes sociales	100
A.5. Preferencias	101
B: Guía de Instalación.....	104
C: Glosario de Términos	105
D: Código fuente del proyecto	106
Bibliografía	107
Referencias Web.....	107



1. INTRODUCCIÓN





1.1. Visión general

Actualmente la posibilidad de acceder de forma rápida a la información desde cualquier dispositivo y desde cualquier lugar tiene especial importancia en la vida cotidiana de la mayoría de las personas. Con el auge de los Smartphones y las Tablets, que poco a poco han ido desplazando a los ordenadores en muchas de sus tareas, el acceso a la información de forma interactiva está a la orden del día.

Debido al uso generalizado de estos dispositivos, el número de aplicaciones para los mismos ha aumentado vertiginosamente, pudiendo decir que existen aplicaciones en el mercado para casi cualquier objetivo que nos podamos imaginar. No podemos negar que el uso de aplicaciones móviles está muy presente en la sociedad actual, lo que ha hecho que se convierta en un mercado realmente llamativo para muchos desarrolladores.

El sector turístico también se ha hecho eco de esta revolución tecnológica de tal forma que muchos turistas actuales hacen uso de aplicaciones móviles que proporcionan información al usuario acerca de un país, ciudad, pueblo, etc con la finalidad de hacer más fácil y versátil las visitas turísticas.

Debido a todo esto, y teniendo en cuenta la gran popularidad de la que goza el Sistema operativo Android en la mayoría de Smartphones y tablets actuales, el presente documento expone el desarrollo de una aplicación de Turismo de la ciudad de Valladolid, que funcionará bajo el sistema operativo Android. Esta aplicación ofrecerá información de utilidad como monumentos, museos u otro tipo de lugares, que el usuario podrá utilizar para complementar su viaje.

Además de facilitar información, esta aplicación puede servir de escaparate a los potenciales visitantes de la ciudad, ya que expone de manera llamativa las opciones que la ciudad puede ofrecer a un turista.

1.2. Objetivos

Como se comenta en el apartado anterior, el proyecto expone el desarrollo de una aplicación móvil para el sistema operativo Android que proporciona información de interés y facilita la visita turística a la ciudad de Valladolid.

Para el desarrollo satisfactorio de este proyecto se han establecido varios objetivos que deben cumplirse. Son los siguientes:

- Realizar el estudio previo de la aplicación móvil a desarrollar mediante el análisis de la misma y la búsqueda de requisitos, tanto funcionales como no funcionales, que permitan tener una idea clara de lo que va a ser la funcionalidad en conjunto de la aplicación.
- Estudio de otras aplicaciones del mismo ámbito atendiendo a sus funcionalidades para mejorar las prestaciones, experiencia y usabilidad proporcionada al usuario en la aplicación a desarrollar.



- Realizar un diseño inicial y visual de la aplicación Android uniendo todas las funcionalidades encontradas durante el análisis de la misma.
- Implementación de la aplicación móvil teniendo en cuenta las fases anteriores.
- Capacidad de la aplicación para conectarse a un servidor que le proporcione datos.

1.3. Motivaciones personales

El Trabajo de Fin de Grado ofrece la posibilidad de consolidar todos los conocimientos adquiridos durante la carrera y poner en práctica la formación adquirida.

Este proyecto en concreto permite mejorar los conocimientos Android en desarrollo de aplicaciones adquiridos durante la carrera y trabajar con nuevas tecnologías. También es importante destacar los conocimientos adquiridos en el uso de servicios web utilizadas para intercambiar información. Un tema desconocido durante la carrera y del que se han podido obtener suficientes conocimientos para aplicarlos a proyecto en cuestión.

Otra gran motivación a la hora de desarrollar esta aplicación es ser conscientes de que puede servir realmente de utilidad para el turismo de la ciudad de Valladolid.

Además de las nombradas, podemos citar como motivación adicional, poder afianzar los conocimientos adquiridos en el lenguaje Java y realizar un proyecto desde cero llevando a cabo cada uno de los pasos necesarios para lograr un desarrollo con éxito.

1.5 Estructura del documento

En este apartado se expone la estructura del presente documento. Compuesto por ocho capítulos en los que se describe de manera detallada el proyecto en cuestión. El documento se completa con varios anexos finales que contienen información adicional que complementa y ayuda a mejorar la comprensión del texto.

Los contenidos presentados son los siguientes:

- **Capítulo 1. Introducción:** en el cual nos encontramos, se presenta el contexto y la finalidad del presente documento y un breve resumen del estado del mercado en este tipo de aplicaciones.
- **Capítulo 2. Contexto tecnológico:** donde se presenta una breve descripción de los conceptos tecnológicos sobre los que se apoya el proyecto, así como las diferentes tecnologías, herramientas y lenguajes utilizados para el desarrollo del mismo.



- **Capítulo 3. Gestión del proyecto:** en este capítulo se presenta la metodología utilizada para la realización del proyecto. También se expondrá la planificación seguida en el desarrollo de éste. Además se realizará un análisis de riesgos y costes.
- **Capítulo 4. Análisis del sistema:** donde se detallará la fase de análisis. En él se expondrán los requisitos de la aplicación y se hará un estudio en profundidad del análisis del sistema.
- **Capítulo 5. Diseño:** en este apartado se detalla la fase de diseño del desarrollo. Todas las decisiones tomadas en este capítulo serán implementadas y detalladas en el capítulo siguiente.
- **Capítulo 6. Implementación:** se comenta de forma breve como se ha llevado a cabo la implementación del sistema. Se hará hincapié en algunos detalles que tengan la suficiente relevancia como para que el lector deba conocerlas.
- **Capítulo 7. Pruebas:** en este capítulo se exponen todas las pruebas llevadas a cabo para comprobar que la solución se ha desarrollado completamente y cumple la totalidad de los requisitos expuestos en el capítulo tres.
- **Capítulo 8. Conclusiones y trabajos futuros:** en este último apartado se describe las conclusiones obtenidas durante la realización del proyecto y la evaluación final, así como las posibles mejoras que puedan efectuarse en el futuro.
- **Anexos:** que son los siguientes:
 - ✓ Manual de usuario: breve manual de usuario donde se explica cómo utilizar la aplicación.
 - ✓ Guía de instalación: explicación sobre cómo debe realizarse la instalación de la aplicación en un dispositivo.
 - ✓ Glosario de términos: términos y acrónimos utilizados en este documento.
 - ✓ Bibliografía utilizada: referencias bibliográficas de las que se ha hecho uso para la realización del presente documento.



2. CONTEXTO TECNOLÓGICO



2.1. Introducción

En este capítulo se expondrá de manera más o menos detallada el contexto de nuestro proyecto, es decir las tecnologías existentes junto con las utilizadas para el desarrollo del sistema. También se hará referencia a las herramientas que se han utilizado.

2.2. Contexto de Desarrollo

A continuación se explicará más detalladamente las dos tecnologías más destacadas en el contexto de nuestro proyecto.

2.2.1. Android

La aplicación móvil del proyecto está desarrollada para el sistema operativo Android, a continuación se expondrán las características más relevantes de este sistema operativo.

2.2.1.1. Características fundamentales

Android es un sistema operativo basado en el núcleo de Linux. Inicialmente fue desarrollado por Android Inc., empresa comprada por Google en el año 2005. Android fue presentado oficialmente en el año 2007.

Es un sistema de código abierto bajo la licencia de Apache. Fue diseñado especialmente para dispositivos táctiles como smartphones o tabletas. Estas dos características, junto con la gran comunidad de desarrolladores de aplicaciones existente han facilitado mucho su auge en el mercado. Actualmente los móviles que llevan integrado este sistema suponen casi más de la mitad del total. En la siguiente figura se muestra una gráfica de la cuota de mercado de este sistema actualmente:

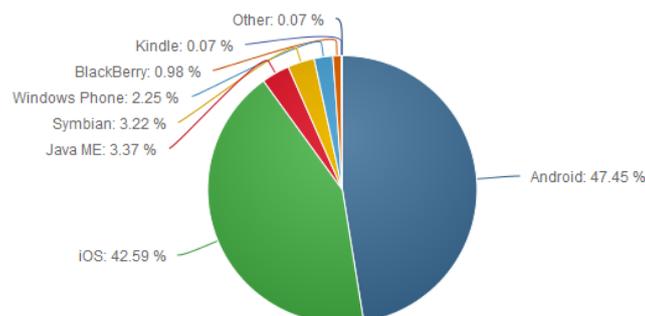


Ilustración 1 Cuota de mercado (año 2015)



Las principales características de este sistema son las siguientes:

- Soporte para Java: las aplicaciones Java corren en una máquina virtual denominada Dalvik.
- Multitarea: soporta la ejecución de aplicaciones en segundo plano.
- Google Play: es un catálogo de aplicaciones que proporciona el propio sistema operativo para la descarga de aplicaciones Android.
- Entorno de desarrollo: actualmente el entorno oficial de desarrollo se denomina Android Studio, con análisis para depuración de memoria y análisis de rendimiento.
- Conectividad: soporta múltiples tecnologías de conectividad. Podemos decir que sigue una filosofía de dispositivo siempre conectado a Internet.
- Almacenamiento:
- Diseño multipantalla: el diseño es adaptable a pantallas de diferente tamaño y resolución.

2.2.1.2. Arquitectura

La arquitectura Android está formada por varias capas que son las siguientes, de abajo hacia arriba:

- **Kernel de Linux:** como ya hemos dicho Android es un sistema basado en Linux. Esta capa actúa como capa de abstracción entre el hardware y las demás capas. También se encarga de la gestión de diversos recursos del sistema, como puede ser la memoria. El desarrollador no accede directamente a esta capa, pero sí mediante las librerías que se encuentran en el siguiente nivel.
- **Librerías:** estas librerías están escritas en lenguaje C y compiladas para la arquitectura específica del sistema. Proporcionan funcionalidad a las tareas que se repiten con frecuencia. Entre estas nos podemos encontrar algunas como el módulo OpenGL (encargado del procesamiento gráfico), SQLite (motor de base de datos) o SSL.
- **Android runtime:** es el entorno de ejecución, aunque realmente no se considera una capa en sí ya que está formada por librerías. El componente más importante es la máquina virtual Dalvik, una variante de la máquina virtual de Java donde se ejecutan las aplicaciones Android. Las aplicaciones escritas en Java se compilan con la extensión *.dex*, propia de esta máquina. Es importante destacar que no podemos ejecutar aplicaciones Java en Dalvik.
- **Framework de aplicaciones:** esta capa está formada por las clases y servicios que utilizarán nuestras aplicaciones. Proporciona una plataforma de desarrollo orientada a la reutilización de componentes.
- **Aplicaciones:** esta capa está compuesta por todas las aplicaciones del dispositivo. Como ya hemos dicho la mayoría escritas en lenguaje Java, pero también podemos encontrarnos aplicaciones nativas escritas en lenguaje C.

En la siguiente figura podemos ver las capas descritas:

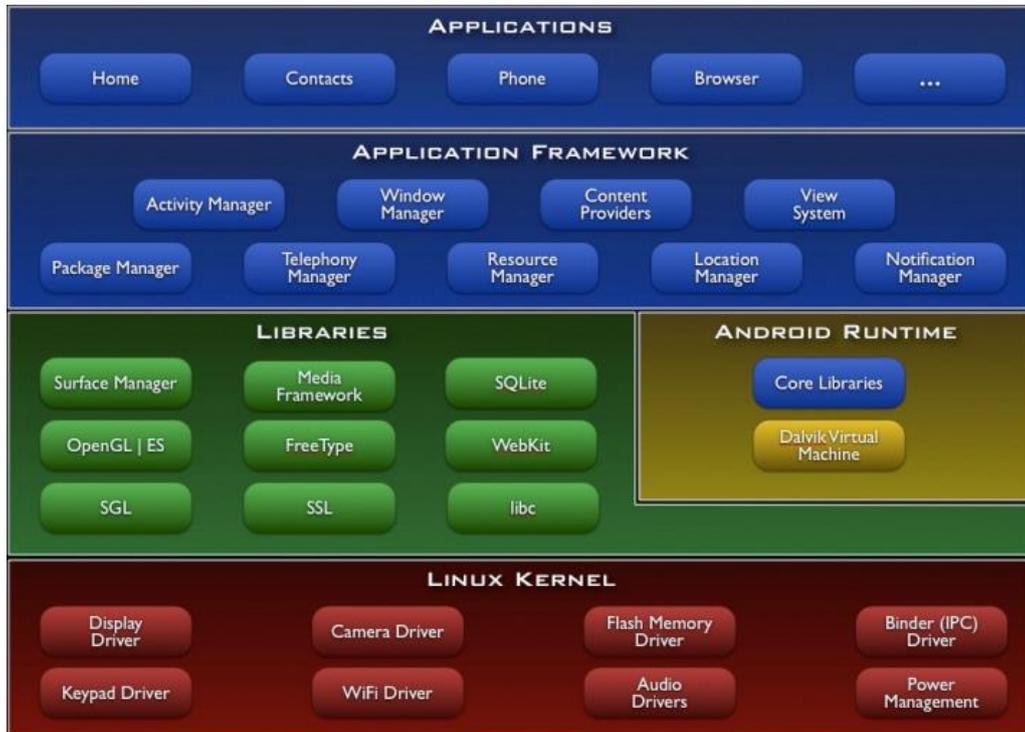


Ilustración 2 Arquitectura del sistema operativo Android

2.2.1. Servicios Web REST

Es necesario una comunicación entre la aplicación móvil y la base de datos donde se aloja la información. Para este propósito se hará uso de un servicio web Restful o REST.

Un servicio web es un conjunto de tecnologías utilizadas para el intercambio de datos entre aplicaciones que utiliza los protocolos web. Los servicios web permiten el intercambio de información independientemente de las propiedades o de las plataformas sobre las que estén instaladas. Existen varios tipos de sistemas web tales como RPC, SOAP o el estándar elegido por nosotros, REST o RESTful.

REST (*Representational State Transfer*) es una arquitectura sencilla que se ejecuta sobre protocolos web tales como HTTP. REST posee unas características básicas:

- **Arquitectura cliente-servidor:** separación clara entre los dos agentes que participan en el intercambio de información.

- **Sin estado:** ni el cliente ni el servidor necesitan recordar ningún estado de la comunicación, cada mensaje tiene toda la información necesaria para comprender la petición.
- **Accesibilidad:** REST utiliza URLs y el protocolo HTTP, por lo que resulta fácil de manipular.

REST es una arquitectura orientada a recursos. Este principio de diseño establece una asociación uno-a-uno entre las operaciones de crear, leer, actualizar y borrar y los métodos HTTP. Estas operaciones serán las utilizadas para acceder a los recursos y son las siguientes:

- *POST* para crear un recurso en el servidor.
- *GET* para obtener un recurso.
- *PUT* para cambiar el estado de un recurso o actualizarlo.
- *DELETE* para eliminar un recurso.

Cada recurso debe tener definido:

- Una URI
- La representación de la respuesta que soporta, como puede ser JSON o XML.
- La operación asociada, entre las que se han descrito.

En la siguiente figura podemos ver el proceso de una petición REST:

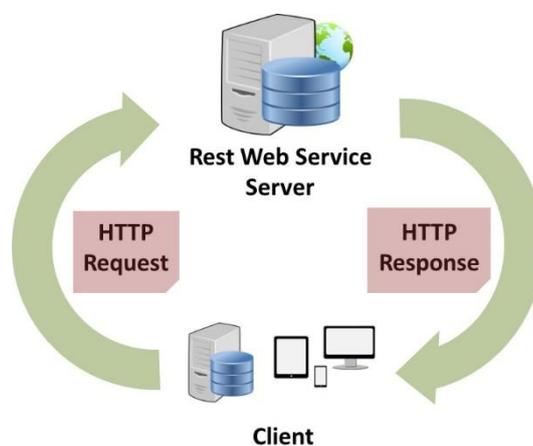


Ilustración 3 Petición REST



2.3. Lenguajes de programación

Para el desarrollo de nuestra aplicación Android hemos utilizado el lenguaje de programación Java y XML. Además para recuperar y almacenar los datos que manejará la aplicación se ha utilizado JSON.

El servicio web también estará escrito en el lenguaje Java.

A continuación se explicará con más detalle cada uno de estos lenguajes.

2.3.1. Java

Lenguaje de programación orientado a objetos (OOP). Este lenguaje fue desarrollado por la empresa Sun Microsystems, más tarde adquirida por Oracle. El lenguaje Java está estructurado por paquetes, espacios de nombres de Java compuestos por clases.

Si nos centramos en nuestro sistema podemos destacar las siguientes características:

- Orientado a objetos: con todos los conceptos que este paradigma posee: herencia, polimorfismo, etc.
- Simple: Java posee una sintaxis simple y de fácil aprendizaje.
- Gran número de bibliotecas disponibles: las cuales permiten realizar prácticamente cualquier tipo de aplicación.
- Distribuido: Java nos proporciona múltiples clases para la comunicación con servidores o clientes remotos.
- Robusto: diseñado para el desarrollo con alta fiabilidad. Proporciona numerosas comprobaciones en tiempo de compilación y ejecución. Una de las más importantes es el sistema de recolección de basura que se encarga de liberar memoria inutilizada sin necesidad de declararlo explícitamente.
- Multihilo: soporta la creación de múltiples hilos sincronizados. Para el caso que nos ocupa un hilo puede encargarse de las comunicaciones remotas mientras otro se encarga de la interacción con el usuario.

2.3.2 XML

XML (eXtensible Markup Language), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C). Permite representar información de forma estructurada. Es un lenguaje de fácil escritura y procesamiento. Sus principales características son las siguientes:

- Basado en texto.
- Es extensible, añadiendo nuevas etiquetas.
- Está orientado a contenidos.
- Es flexible.



En nuestro caso este lenguaje será el utilizado para el diseño de las vistas de la aplicación, que serán posteriormente procesadas en Java.

También XML estará presente en la base de datos, puesto que ésta almacenará cierta información en un campo XML del que se deberán extraer los datos, esto se explicará detenidamente en el capítulo 5.

2.3.3. JSON

No podemos considerarlo un lenguaje propiamente dicho, sino un formato de texto para el intercambio de información. JSON (*JavaScript Object Notation*) nació como alternativa a XML para la representación de información. Puede ser leído por múltiples lenguajes y de manera rápida, lo que le ha facilitado el uso generalizado en un gran número de aplicaciones actuales.

2.3.4. Xpath

Es un lenguaje para la construcción de expresiones que permitan acceder a la información de un documento XML. Será utilizado para el acceso a información en la base de datos cuando el campo al que se acceda sea un campo XML.

2.3.5. UML

UML o Lenguaje de Modelado Unificado, es el lenguaje utilizado para la fase de análisis y diseño del sistema. Podemos decir que más que un lenguaje se trata de una serie de normas y estándares gráficos para visualizar, especificar, construir y documentar un sistema.

Hacemos uso de UML para describir el modelo de nuestro sistema.

2.4. Herramientas

A continuación se detallarán las herramientas utilizadas para el desarrollo del proyecto.

2.4.1. Android SDK

Android Software Development Kit es el conjunto de herramientas de desarrollo que nos proporciona Android para la realización de aplicaciones. Está desarrollado por Google y entre las herramientas que nos ofrece se encuentran un depurador de código, librerías de compatibilidad de versiones, un emulador de dispositivos Android (ADV Manager) y diversas herramientas para la interacción con otro tipo de dispositivos.



2.4.2. Android Studio.

Es el entorno de desarrollo integrado (IDE) oficial de Android. Está basado en IntelliJ IDEA. Es gratis y se encuentra bajo la licencia Apache 2.0.

Como ya hemos dicho está desarrollado para la implementación de aplicaciones Android, por tanto para su uso necesitamos tener instalada la SDK de Android.

Android Studio utiliza Gradle para la compilación, empaquetado y despliegue de nuestra aplicación. Nos proporciona una herramienta para la generación de ficheros “apk”, extensión de las aplicaciones propias de Android.

Además nos proporciona herramientas para el diseño de interfaces de aplicaciones más avanzadas que otros IDEs de desarrollo.

2.4.3. Eclipse Web Tools Platforms

Se trata del entorno de desarrollo de eclipse con el plugin de desarrollo de aplicaciones web integrado (WTP).

Eclipse es un IDE desarrollado originalmente por IBM y actualmente por la Fundación Eclipse. Actualmente se encuentra bajo la licencia Eclipse Public License. Está compuesto por un conjunto de herramientas de programación de código abierto multiplataforma.

La plataforma se estructura bajo el concepto de plug estructurados que se pueden añadir mediante Eclipse MarketPlace. El eclipse utilizado lleva integrado el plugin Web Platform Tools que nos proporciona herramientas necesarias para el desarrollo de aplicaciones web.

2.4.4. MySQL WorkBench

MySQL Workbench es un software creado por la empresa Sun Microsystems, esta herramienta permite el diseño de bases de Entidad-Relación, creación y mantenimiento para el sistema de base de datos MySQL.

Gracias a MySQL WorkBench podemos conectarnos de manera remota a una base de datos MySQL y realizar las operaciones necesarias sobre ésta.



2.4.5. Astah Community.

Editor UML para la realización de diagramas de diseño y de análisis. La versión Community es de licencia gratuita y proporciona todas las herramientas necesarias en nuestro caso.

Se trata de una herramienta intuitiva y de fácil manejo, algo que facilita el desarrollo de los diagramas presentes en este documento.

2.4.7. Dropbox

Dropbox se trata de una herramienta de almacenamiento y compartición de archivos en la nube. Mediante un usuario y una contraseña podemos almacenar nuestros ficheros en Internet. Resulta de gran utilidad como copia de seguridad.



3. GESTIÓN DEL PROYECTO





3.1. Introducción

En este capítulo se detallarán los aspectos más importantes de la planificación y gestión del proyecto. Se hará mención al modelo de proceso elegido para el desarrollo del mismo, así como un análisis de los posibles riesgos a tener en cuenta y la planificación del mismo.

3.2. Modelo de Proceso

La metodología para el desarrollo del software es un modo sistemático de realizar y gestionar un proyecto para llevarlo a cabo con altas posibilidades de éxito. De esta forma el proyecto se divide en pequeñas fases o ciclos de vida. Un ciclo de vida define las etapas de un proyecto de desarrollo de software para transformar la información que proporciona el usuario en un producto final que se ajuste a sus necesidades.

Es importante definir de una forma clara las etapas principales que componen un ciclo de vida:

- **Análisis de Requisitos:** Se definirán los requisitos del sistema software a partir de consultas con los clientes y usuarios del sistema final. Asociada a esta fase se desarrollará un documento de análisis del sistema.
- **Diseño:** se establecerá la arquitectura completa del sistema. En esta fase se establecen las clases de diseño y el modelo de despliegue del sistema.
- **Implementación:** se llevará a cabo la implementación del sistema en el lenguaje o lenguajes de programación establecidos. Esta fase se detalla más adelante en el documento de implementación del sistema.
- **Pruebas:** realización de pruebas sobre la implementación realizada del sistema, de forma que los requisitos establecidos se cumplan correctamente.
- **Implantación del sistema:** instalación del sistema al cliente.

En el presente proyecto se sigue un modelo de ciclo de vida evolutivo en espiral. Con este modelo los requisitos del usuario pueden ir variando en cualquier momento, debido a que inicialmente es difícil obtener un documento con los requisitos totalmente definidos. Con este modelo, el proyecto se adapta de mejor forma al usuario ya que afronta los nuevos requisitos que surjan mediante una iteración de ciclos requisitos-desarrollo-evaluación.

Nuestro modelo estará guiado por el desarrollo exploratorio, es decir la evolución del sistema se enfocará en explorar con el usuario los requisitos hasta llegar a un sistema final. El sistema evoluciona a medida que se añaden nuevas características.



Esta forma de desarrollo del proyecto posee ciertas ventajas:

- La especificación puede desarrollarse de forma creciente.
- Tanto el usuario como el desarrollador logran un mayor entendimiento del sistema.

3.3. Gestión de Riesgos

En este apartado se describen las posibles situaciones adversas que pueden darse durante el desarrollo del proyecto y pueden afectar a la planificación y desarrollo del mismo.

Para cada situación de riesgo se establecerá un modo de actuación para corregirlo. El objetivo que se persigue es implantar una filosofía de prevención, identificando los posibles fallos y adoptando medidas correctivas cuando sea posible para poder prevenir los fallos.

Todos los riesgos posibles identificados se analizan teniendo en cuenta los siguientes aspectos:

- **Riesgo:** Nombre y descripción breve del riesgo a tratar.
- **Probabilidad:** Es la probabilidad de ocurrencia del fallo. Se evalúa en una escala desde una probabilidad de ocurrencia baja, hasta una muy alta o catastrófica.
- **Impacto o gravedad:** descripción de los posibles efectos sobre el proyecto y su magnitud. Se valora en una escala desde insignificante hasta un impacto catastrófico.
- **Riesgo:** resultado de multiplicar el impacto por la probabilidad de ocurrencia.
- **Estrategia de prevención:** Acciones o medidas a llevar a cabo en el proyecto para evitar que el riesgo aparezca o en caso de que este ocurra, minimizar su efecto en la medida de lo posible. Estas medidas son aplicadas antes de que el riesgo se transforme en un hecho.
- **Plan de contingencia:** Acciones o medidas a llevar a cabo en el proyecto cuando el riesgo ha ocurrido.

La siguiente tabla detalla los riesgos que pueden afectar al desarrollo del proyecto, así como la probabilidad de que ocurran, la magnitud del efecto que conllevaría y la estrategia que pondremos en marcha para disminuir la probabilidad de dichos riesgos.



Riesgo nº	1
Nombre	Planificación optimista
Descripción	Planificación inicial no acorde con la realidad puede retrasar ciertas tareas del proyecto.
Probabilidad	Media
Impacto	Medio
Riesgo	Medio
Acción correctora	Realización de una planificación lo más realista posible.
Plan de contingencia	Reorganización del trabajo.

Riesgo nº	2
Nombre	Pérdida de información
Descripción	Debido a una pérdida de información se pierde parte del proyecto
Probabilidad	Baja
Impacto	Alto
Riesgo	Medio
Acción correctora	Mantener copias de seguridad.
Plan de contingencia	Realizar de nuevo el trabajo perdido

Riesgo nº	3
Nombre	Avería de equipos
Descripción	Algún equipo utilizado durante el desarrollo sufre una avería que lo deja inutilizable.
Probabilidad	Baja
Impacto	Media
Riesgo	Medio
Acción correctora	Tener en cuenta en la planificación
Plan de contingencia	Recambio de equipo por otro en buenas condiciones.

Riesgo nº	4
Nombre	Enfermedades o viajes imprevistos
Descripción	Enfermedad o viaje obligado no previsto en la planificación
Probabilidad	Media
Impacto	Medio
Riesgo	Medio
Acción correctora	Realiza una planificación teniendo en cuenta este riesgo.
Plan de contingencia	Reajustar la planificación inicial



Riesgo nº	5
Nombre	Cambios imprevistos en requisitos
Descripción	Los requisitos pueden variar a lo largo del tiempo por cambios de funcionalidad, pero un gran cambio puede retrasar otras tareas.
Probabilidad	Media
Impacto	Alto
Riesgo	Medio
Acción correctora	Mejora de la planificación temporal para posibles cambios en los requisitos.
Plan de contingencia	Realizar el trabajo que impliquen los nuevos cambios.

Riesgo nº	6
Nombre	Integración de componentes
Descripción	La integración de las partes del sistema resulta dificultosa y retrasa otras tareas del proyecto.
Probabilidad	Media
Impacto	Alto
Riesgo	Medio
Acción correctora	Establecer correctamente las partes que deben integrarse entre sí y de qué forma antes de llevarse a cabo.
Plan de contingencia	Modificación de los componentes para poder integrarlos correctamente.

3.4. Planificación

En este apartado se presenta la planificación temporal del proyecto en cuestión. La planificación inicial que se presenta a continuación ha sufrido algunas variaciones a lo largo del desarrollo del proyecto debido a algunos riesgos citados en el apartado anterior, que aunque se tuvieron en cuenta en la planificación hicieron variarla en algunos casos.

La planificación temporal del proyecto se desarrolla a lo largo de 300 horas de trabajo, con una fecha de finalización de proyecto estipulada:



		Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1		Fase inicial	28 días	lun 22/12/14	jue 29/01/15	
2		Investigación y documentación	3 días	lun 22/12/14	sáb 27/12/14	
3		Tecnologías a utilizar	4 días	dom 28/12/14	mié 31/12/14	2
4		Instalación del entorno de desarrollo	1 día	vie 02/01/15	vie 02/01/15	3
5		Planificación temporal y gestión	5 días	lun 05/01/15	vie 09/01/15	4
6		Análisis del sistema	5 días	lun 12/01/15	vie 16/01/15	5
7		Diseño del sistema	10 días	sáb 17/01/15	jue 29/01/15	6
8		Reunion fin de ciclo	0 días	vie 16/01/15	vie 16/01/15	6
9		Implementación	70 días	vie 30/01/15	jue 07/05/15	
10		Servicio Web	10 días	vie 30/01/15	jue 12/02/15	7
11		Aplicación Móvil	60 días	vie 13/02/15	jue 07/05/15	7
12		Reunión Fin de ciclo	0 días	jue 07/05/15	jue 07/05/15	
13		Pruebas	10 días	mar 12/05/15	jue 21/05/15	11
14		Pruebas del sistema completo	3 días	mar 12/05/15	jue 14/05/15	9
15		Revisión del ciclo	5 días	dom 17/05/15	jue 21/05/15	14
16		Reunion fin de ciclo	0 días	jue 14/05/15	jue 14/05/15	14
17		Cierre de proyecto	15 días	vie 22/05/15	mié 10/06/15	
18		Documentación del proyecto	12 días	vie 22/05/15	vie 05/06/15	15
19		Revisión de documentación	3 días	lun 08/06/15	mié 10/06/15	18
20		Reunion fin de ciclo	0 días	mié 10/06/15	mié 10/06/15	

Ilustración 4 Planificación del proyecto

Teniendo en cuenta que las horas de trabajo al día se establecen en una media 3, nuestro proyecto tendrá una duración de 118 días laborables, es decir 354 horas de trabajo totales. Se ha realizado una planificación que no sea demasiado optimista para así poder abordar posibles imprevistos que puedan surgir durante el desarrollo del proyecto.

A continuación se muestra la planificación temporal del proyecto en una línea de tiempo:

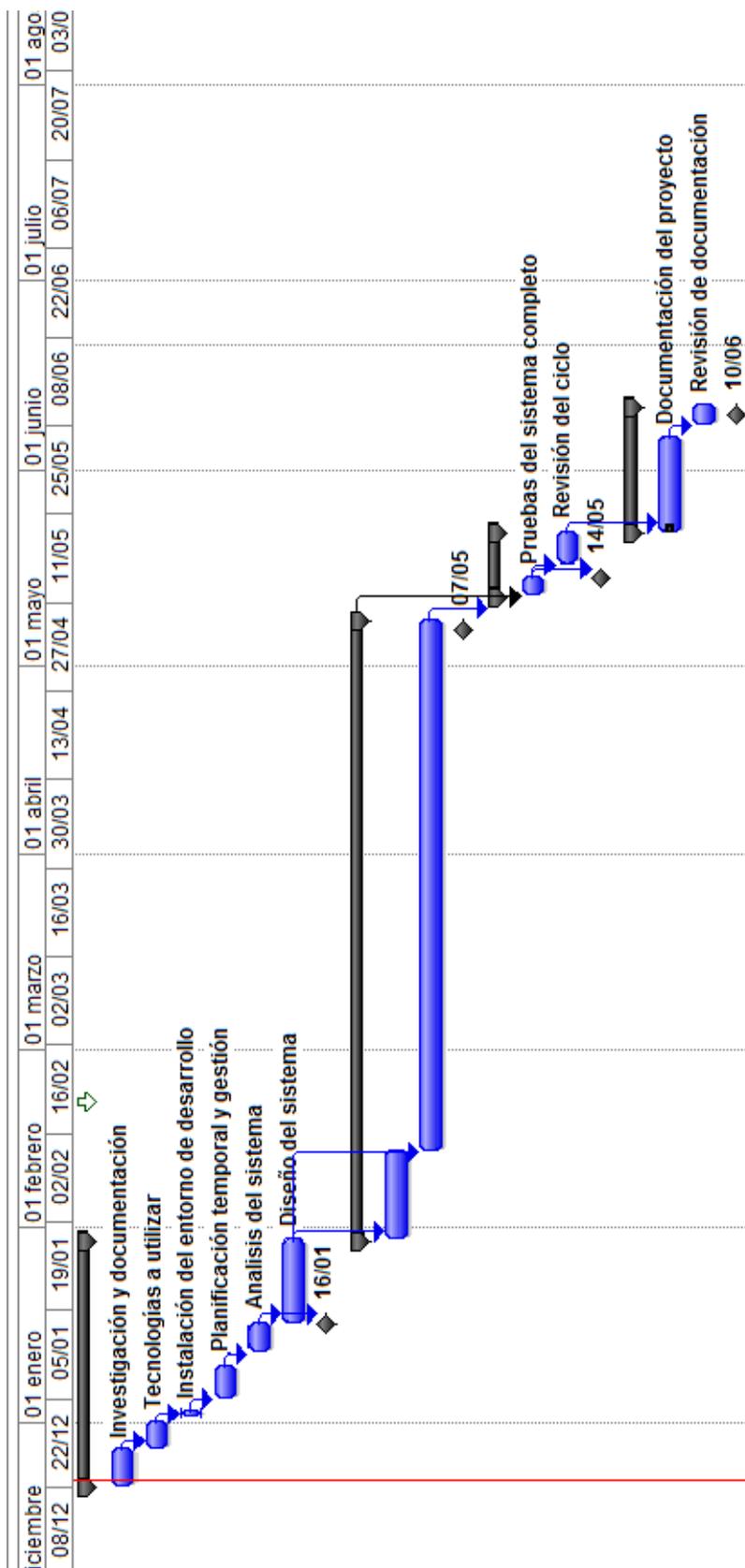


Ilustración 5 Diagrama Gantt, planificación temporal



Como se puede ver en la figura anterior, el desarrollo del proyecto se ha dividido en varias fases, que se han descrito en el modelo de proceso del sistema. Se tendrá muy en cuenta las fechas de finalización de cada etapa, pues al ser un sistema evolutivo pueden ser de gran importancia si aparecen desviaciones en la planificación inicial.

3.5. Recursos

A continuación se describen los recursos necesarios para el desarrollo del proyecto.

3.5.1. Recursos hardware

Para el desarrollo del proyecto se utilizarán los siguientes equipos:

- Portátil Asus K3S, con procesador Intel i7 (2.5GHz, 3MB (L3) Cache) y 8GB de Memoria RAM DDR3.
- Servidor web para publicar los componentes web del proyecto.

Además de los equipos de desarrollo se utilizarán los siguientes dispositivos para realizar las pruebas de la aplicación móvil.

Tablet Samsung Galaxy Tab 3 8''
Android 4.4 (KitKat)
Procesador Dual Core 1.5 GHz
1,5 GB RAM

Tablet Samsun Galaxy Tab 3 10.1''
Android 4.4 (KitKat)
Procesador Dual Core 1.6 GHz
1 GB RAM

Smartphone Samsung Galaxy Grand Neo
Android 4.4 (KitKat)
Procesador Quad Core 1.2 GHz
1 GB RAM

El primer dispositivo será el utilizado durante todo el desarrollo, mientras que los otros dos se han utilizado para pruebas muy específicas y puntuales, por lo que su coste no se incluirá en la estimación.



3.5.2. Recursos software

Para el desarrollo del proyecto se utilizarán los siguientes recursos software o herramientas, todas de licencia gratuita como se ha comentado anteriormente:

Herramienta	Propósito
Windows 8 Professional	Sistema operativo
Astah Community	Análisis y diseño del sistema
Android Studio	Implementación
Eclipse IDE	Implementación
MySQL WorkBench	Implementación

3.5.3. Recursos humanos

Para el desarrollo del sistema serán necesarios los perfiles mostrados en la siguiente tabla:

Recurso	Propósito
Analista	Análisis y diseño del sistema
Programador	Implementación del sistema

La asignación de recursos no será constante a lo largo de todo el proyecto sino que cada uno realizará su trabajo en el momento que le corresponda, estipulado en la planificación inicial.

3.6. Costes y presupuesto

A continuación se describen los costes habituales que un proyecto de desarrollo de software ocasiona. Posteriormente se presentará el presupuesto de la aplicación desarrollada para poder estimar un precio de cara a los usuarios.

Para realizar dicha valoración de los costes se tienen en cuenta los siguientes factores:

- Los costes del hardware y software utilizado, incluyendo su mantenimiento y amortización.
- Horas de trabajo, donde se incluyen los gastos derivados de los recursos humanos utilizados, como nivel de profesionalidad o seguridad social del individuo.
- Costes derivados: como gastos de luz, internet o desplazamientos.



Como se ha comentado en el capítulo dos, todas las herramientas utilizadas para el desarrollo son de licencia gratuita, por lo que éstas no supondrán ningún coste en el proyecto al tratarse de software libre.

Para la facturación de horas de trabajo se han establecido las siguientes tarifas:

- Analista: 25 euros/hora.
- Programador: 15 euros/hora.

En la siguiente tabla se puede ver la estimación de costes teniendo en cuenta todos los aspectos citados:

Concepto	Cantidad	Precio
Herramientas de desarrollo	6	0
Herramientas hardware		
Ordenador portátil	1	50
Samsung Galaxy s3 7''	1	40
	<i>Subtotal</i>	<i>90</i>
Horas de trabajo		
Analista y diseñador	144 h	3600
Programador	210 h	3150
	<i>Subtotal</i>	<i>6750</i>
Costes derivados		60
	TOTAL	6810

El coste total del proyecto se establece en la cantidad de 6810 euros.

3.6.1. Presupuesto final

Teniendo en cuentas los gastos estimados en el apartado anterior y que la aplicación estará totalmente libre de publicidad, se establece el siguiente presupuesto final para el desarrollo del proyecto. Se han estimado unos beneficios del 30%, por lo que el precio final del proyecto ascendería a 8853 euros totales, sin incluir mantenimiento de la aplicación.

Este presupuesto incluiría todos los gastos totales el primer año, sin incluir el mantenimiento de la aplicación.



4. ANÁLISIS DEL SISTEMA





4.1 Introducción

En este capítulo se expondrán cuáles son las características y funciones que debe cumplir el sistema. Para facilitar esta tarea se ha realizado una recogida de requisitos y su posterior descomposición en casos de uso.

4.2 Especificación de requisitos

La especificación de requisitos es una fase fundamental en el desarrollo de todo sistema, pues permite indicar detalladamente la funcionalidad que debe cubrir el sistema y las características que debe poseer. A continuación se muestran los requisitos del sistema en cuestión.

4.2.1. Requisitos funcionales

- RF1. El Sistema deberá ser capaz de descargarse los datos almacenados en una base de datos remota referentes a :
 - Monumentos
 - Museos
 - Rutas.

- RF2. La descarga de datos se realizará tanto de manera automática, como de forma manual cuando el usuario lo requiera.

- RF3. La aplicación constará de los siguientes apartados:
 - Lista de lugares de interés
 - Lista de rutas
 - Buscador
 - Mapa de lugares
 - Información de interés
 - Redes sociales
 - Preferencias
 - Otros contenidos

- RF4. El sistema deberá mostrar una lista de los monumentos y lugares obtenidos mediante la sincronización de datos.

- RF5. En el apartado de lugares de interés, el sistema permitirá al usuario seleccionar el tipo de datos a mostrar, pudiendo ser estos de dos tipos: lugares y monumentos.



- RF6. El sistema mostrará información detallada relativa a un monumento o lugar en particular.
- RF7. El sistema permitirá al usuario visualizar en un mapa la ruta desde su posición actual hasta la dirección del monumento o lugar seleccionado.
- RF8. El sistema permitirá realizar llamadas de un lugar seleccionado cuando éste disponga de un número de teléfono y el dispositivo así lo permita.
- RF9. El sistema deberá mostrar una lista de las rutas obtenidas mediante la sincronización de datos.
- RF10. El sistema permitirá al usuario seleccionar el tipo de rutas a mostrar.
- RF11. El sistema mostrará información detallada relativa a una ruta en particular.
- RF12. El sistema permitirá la búsqueda de monumentos y lugares.
- RF13. El sistema debe permitir realizar un filtrado más o menos detallado de la búsqueda a realizar.
- RF14. El sistema debe ser capaz de ubicar en un mapa el dispositivo que se esté usando, siempre que éste tenga habilitados los sistemas de localización mediante GPS.
- RF15. El sistema debe mostrar todos los monumentos y lugares ubicados en un mapa.
- RF16. El sistema debe permitir al usuario seleccionar los tipos de datos a mostrar en el mapa.
- RF17. El sistema debe permitir al usuario seleccionar un lugar del mapa y proporcionarle información más detallada acerca de éste.
- RF18. La aplicación se podrá ejecutar en dos idiomas: español e inglés.
- RF19. El sistema permitirá al usuario seleccionar el idioma que éste desee en el apartado preferencias.

4.2.2. Requisitos no funcionales

- RNF1. La aplicación deberá ser soportada por dispositivos móviles que funcionen con Android 3.0 (Honeycomb) en adelante.
- RNF2. La aplicación deberá adaptarse a dispositivos con pantallas de tamaños desde 3 hasta 10 pulgadas.



- RNF3. La conexión a la base de datos se realizará mediante un servicio web.
- RNF4. El servicio web será implementado por la arquitectura REST.
- RNF5. El dispositivo deberá soportar la conexión a internet.
- RNF6. La aplicación será extensible.
- RNF7. La aplicación deberá ser implementada en el lenguaje de programación JAVA.
- RNF8. La aplicación deberá ser intuitiva y fácil de manejar.
- RNF9. La aplicación deberá ser fiable y robusta.

4.3. Modelado de Casos de Uso

En este apartado se tratará de definir cómo debe comportarse nuestro sistema. Para ellos se definen los usuario que utilizarán el sistema y posteriormente se define el comportamiento del mismo.

Existen cuatro componentes básicos en el modelado de casos de uso:

- Límite del sistema.
- Actores.
- Casos de uso
- Relaciones.

En los siguientes apartados se realizará el estudio completo de casos de uso, incluyendo los diagramas y componentes necesarios.

4.3.1. Identificación de los actores

Un actor es la especificación de una entidad externa que interactúa con un sistema.

En nuestro sistema nos encontramos con un único actor, al que denominaremos Usuario. Éste representa cualquier persona que acceda a la aplicación y haga uso de esta.

4.3.2. Casos de Uso

Los casos de uso son la especificación de la secuencia de acciones que un sistema realizar para interactuar con actores externos, es decir define los pasos que el usuario debe seguir para completar cierta función del sistema.

4.3.2.1. Diagrama de Casos de uso

Mediante los diagramas de caso de uso se pueden ver las funcionalidades que ofrece el sistema y que los usuarios pueden realizar. En este diagrama se muestra el límite del sistema, representado.

En la figura se puede ver dicho diagrama:

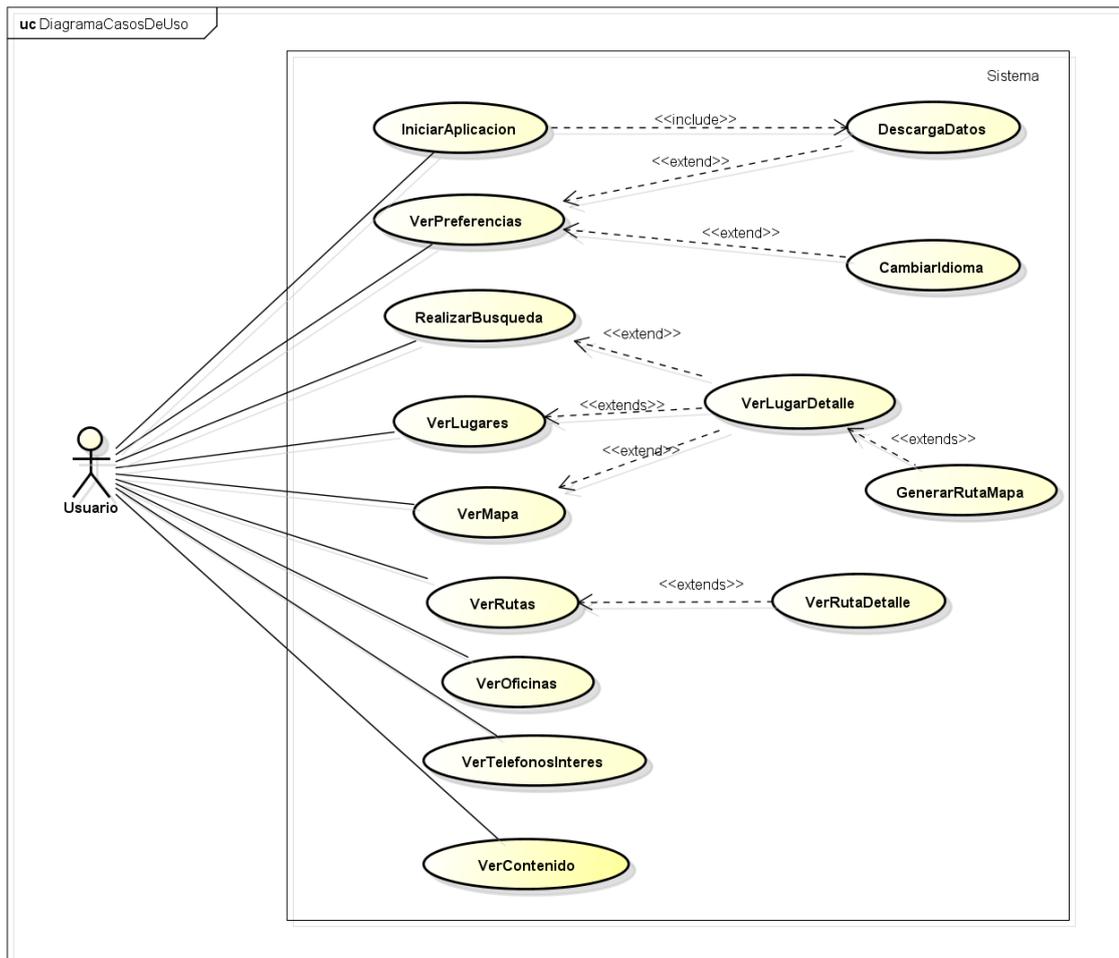


Ilustración 6 Diagrama de casos de uso



4.3.2.2. Descripción de Casos de Uso

A continuación se describirán los casos de uso del diagrama anterior. En cada caso se incluirá el diagrama de secuencia asociado.

ID: CU_1	Caso de Uso: IniciarAplicacion	
Breve descripción:	El usuario desea iniciar la aplicación de Turismo de Valladolid	
Actores principales:	Usuario	
Precondiciones:	No existen precondiciones	
Flujo principal:	Paso	Acción
	1	El caso de uso comienza cuando el usuario accede al sistema.
	2	El sistema comprueba la conexión a internet del dispositivo. 2.1. Si no existe conexión, el sistema avisa al usuario.
	3	<<include>> DescargaDatos
Postcondiciones:	El sistema tiene almacenados los datos actualizados	
Flujos alternativos:	Si no existe conexión a internet, el sistema avisa al usuario.	

ID: CU_2	Caso de Uso: VerPreferencias	
Breve descripción:	El usuario accede a las preferencias para modificar ciertas opciones de la aplicación.	
Actores principales:	Usuario	
Precondiciones:	No existen precondiciones	
Flujo principal:	Paso	Acción
	1	El caso de uso comienza cuando el usuario desea acceder a las preferencias del sistema.
	2	El sistema muestra las preferencias de la aplicación.
	3	El usuario puede modificar uno de los apartados disponibles.
	4	El caso de uso finaliza cuando el usuario abandona las preferencias del sistema.
Postcondiciones:	No existen postcondiciones	
Flujos alternativos:	No hay flujos alternativos posibles.	



ID: CU_3	Caso de Uso: RealizarBúsqueda	
Breve descripción:	El usuario accede a la búsqueda de lugares para poder filtrar los lugares que desea visualizar.	
Actores principales:	Usuario	
Precondiciones:	No existen precondiciones	
Flujo principal:	Paso	Acción
	<ol style="list-style-type: none">1 El caso de uso comienza cuando el usuario accede a la búsqueda de lugares.2 El sistema muestra la vista correspondiente a la pantalla de búsqueda.3 El usuario introduce el texto y los filtros de búsqueda que desee.4 El sistema busca la información almacenada en el sistema que coincida con los filtros introducidos.5 El sistema muestra al usuario los resultados de la búsqueda6 El usuario puede visualizar todos los resultados.7 El caso de uso finaliza cuando el usuario abandona la búsqueda del sistema.	
Postcondiciones:	No existen postcondiciones	
Flujos alternativos:	El usuario seleccionar un elemento de la lista	

ID: CU_4	Caso de Uso: VerMapa	
Breve descripción:	El usuario accede al mapa para obtener información sobre la ubicación de los monumentos y museos de la ciudad en el mapa.	
Actores principales:	Usuario	
Precondiciones:	El usuario se encuentra en el menú principal	
Flujo principal:	Paso	Acción
	<ol style="list-style-type: none">1 El caso de uso comienza cuando el usuario accede al mapa de la aplicación.2 El sistema comprueba que el dispositivo tiene habilitada la localización GPS.<ol style="list-style-type: none">2.1. En caso negativo, muestra un aviso al usuario.2.2. En caso positivo, muestra la pantalla de mapas de la aplicación.3 El sistema introduce los monumentos y museos en el mapa.4 El usuario puede moverse por el mapa y seleccionar un lugar que desee.	



	7 El caso de uso finaliza cuando el usuario abandona los mapas de la aplicación.
Postcondiciones:	No existen postcondiciones
Flujos alternativos:	El usuario selecciona un lugar del mapa

ID: CU_5	Caso de Uso: VerRutas	
Breve descripción:	El usuario accede al apartado de rutas para visualizar un listado con las rutas existentes.	
Actores principales:	Usuario	
Precondiciones:	No existen precondiciones	
Flujo principal:	Paso	Acción
	<ol style="list-style-type: none"> 1 El caso de uso comienza cuando el usuario desea visualizar las rutas. 2 El sistema muestra una vista con los tipos de rutas existentes. 3 El usuario selecciona el tipo de ruta que desea visualizar. 4 El sistema muestra las rutas relativas al tipo que ha seleccionado el usuario. 5 Si el usuario desea consultar otro tipo de ruta, pulsa el botón "Atrás" del menú superior de la aplicación 6 El caso de uso finaliza cuando el usuario abandona el apartado de rutas o bien seleccionar una ruta. 	
Postcondiciones:	No existen postcondiciones	
Flujos alternativos:	El usuario selecciona una ruta de la lista.	



ID: CU_6	Caso de Uso: VerLugares	
Breve descripción:	El usuario accede al apartado de lugares para visualizar un listado con los monumentos y museos.	
Actores principales:	Usuario	
Precondiciones:	No existen precondiciones	
Flujo principal:	Paso	Acción
	<ol style="list-style-type: none">1 El caso de uso comienza cuando el usuario desea visualizar la lista de lugares del sistema.2 El sistema muestra una vista con los tipos de lugares existentes.3 El usuario selecciona el tipo de lugar que desea visualizar.4 El sistema muestra los lugares existentes relativos al tipo de lugar seleccionado.5 Si el usuario desea consultar otro tipo de lugar, pulsa el botón "Atrás" del menú superior de la aplicación6 El caso de uso finaliza cuando el usuario abandona el listado de lugares o bien selecciona un lugar.	
Postcondiciones:	No existen postcondiciones	
Flujos alternativos:	El usuario selecciona un lugar de la lista.	

ID: CU_7	Caso de Uso: VerOficinas	
Breve descripción:	El usuario accede al apartado de oficinas para ver las oficinas de Turismo existentes en Valladolid.	
Actores principales:	Usuario	
Precondiciones:	No existen precondiciones	
Flujo principal:	Paso	Acción
	<ol style="list-style-type: none">1 El caso de uso comienza cuando el usuario desea ver el listado de Oficinas de Turismo.2 El sistema muestra la vista correspondiente a las oficinas de turismo.3 El usuario puede desplazarse por la pantalla para ver la información relativa a las oficinas.4 El caso de uso finaliza cuando el usuario de oficinas de turismo.	
Postcondiciones:	No existen postcondiciones.	
Flujos alternativos:	No hay flujos alternativos posibles.	



ID: CU_8	Caso de Uso: VerTelefonosInterés	
Breve descripción:	El usuario accede al apartado de teléfonos de interés para obtener información sobre números de teléfono del Ayuntamiento de Valladolid.	
Actores principales:	Usuario	
Precondiciones:	El usuario se encuentra en el menú principal	
Flujo principal:	Paso	Acción
	<ol style="list-style-type: none"> 1 El caso de uso comienza cuando el usuario desea visualizar la lista de teléfonos. 2 El sistema muestra una vista con un listado de teléfonos y su información asociada. 3 El usuario puede seleccionar un teléfono. 4 El sistema comenzará a realizar una llamada al número de teléfono seleccionado. 6 El caso de uso finaliza cuando el usuario abandona la pantalla de teléfonos y sale al menú principal. 	
Postcondiciones:	No existen postcondiciones	
Flujos alternativos:	No hay flujos alternativos posibles.	

ID: CU_9	Caso de Uso: VerContenido	
Breve descripción:	Un usuario de la aplicación desea obtener información turística adicional de la ciudad de Valladolid.	
Actores principales:	Usuario	
Precondiciones:	El dispositivo tiene conexión a internet.	
Flujo principal:	Paso	Acción
	<ol style="list-style-type: none"> 1 El caso de uso comienza cuando el usuario desea visualizar otro tipo de contenidos. 2 El usuario seleccionado el tipo de contenido. 3 El sistema recoge el contenido seleccionado y realiza la descarga de información. 4 El sistema muestra la información solicitada al usuario. 6 El caso de uso finaliza cuando el usuario abandona la pantalla de contenidos. 	
Postcondiciones:	No existen postcondiciones	
Flujos alternativos:	No hay flujos alternativos posibles.	



ID: CU_10	Caso de Uso: DescargaDatos	
Breve descripción:	Un usuario de la aplicación desea realizar la sincronización de los datos y almacenarlos en la aplicación.	
Actores principales:	Usuario	
Precondiciones:	El dispositivo tiene conexión a internet.	
Flujo principal:	Paso	Acción
	<ol style="list-style-type: none">1 El caso de uso comienza cuando el usuario inicia el sistema o bien desea realizar una sincronización de los datos.2 El sistema muestra un aviso de que se está realizando la descarga de datos.3 El sistema completa la descarga de datos.4 El caso de uso finaliza cuando el sistema almacena los datos en un fichero y elimina el aviso de descarga de datos.	
Postcondiciones:	Los datos almacenados en el sistema se encuentran actualizados.	
Flujos alternativos:	<ul style="list-style-type: none">- Si no existe conexión a internet, el sistema avisa al usuario.- La conexión a internet se corta durante la descarga y el caso de uso finaliza.	

ID: CU_11	Caso de Uso: CambiarIdioma	
Breve descripción:	El usuario puede seleccionar el idioma en el que se ejecutará y se mostraran los datos de la aplicación.	
Actores principales:	Usuario	
Precondiciones:	El usuario se encuentra en la pantalla de preferencias	
Flujo principal:	Paso	Acción
	<ol style="list-style-type: none">1 El caso de uso comienza cuando el usuario desea seleccionar un idioma en la aplicación.2 El sistema muestra un listado con los idiomas disponibles de la aplicación.3 El usuario seleccionar el idioma que desee de la lista.4 El caso de uso finaliza cuando el sistema recoge la opción seleccionada y la almacena como preferencia.	
Postcondiciones:	El sistema ha actualizado las preferencias.	
Flujos alternativos:	No hay flujos alternativos posibles.	



ID: CU_12	Caso de Uso: VerRutaDetalle	
Breve descripción:	El usuario puede seleccionar una ruta de la lista que se le presenta para obtener más información sobre ésta.	
Actores principales:	Usuario	
Precondiciones:	El usuario se encuentra en la pantalla rutas	
Flujo principal:	Paso	Acción
	1	El caso de uso comienza cuando el usuario selecciona una ruta de la lista.
	2	El sistema busca la información sobre esa ruta y se la muestra al usuario.
	3	El usuario puede visualizar los datos referentes a la ruta seleccionada.
	4	El caso de uso finaliza cuando el usuario sale del detalle de la ruta.
Postcondiciones:	El sistema muestra la lista de rutas para el mismo tipo	
Flujos alternativos:	No hay flujos alternativos posibles.	

ID: CU_13	Caso de Uso: VerLugarDetalle	
Breve descripción:	El usuario puede seleccionar un lugar de la lista que se le presenta para obtener más información sobre éste.	
Actores principales:	Usuario	
Precondiciones:	El usuario se encuentra en la pantalla de preferencias	
Flujo principal:	Paso	Acción
	1	El caso de uso comienza cuando el usuario selecciona un lugar de la lista.
	2	El sistema muestra la información sobre ese lugar y se la muestra al usuario.
	3	El usuario puede visualizar los detalles relativos a ese lugar seleccionado.
	5	El caso de uso finaliza cuando el usuario abandona la vista con los detalles del lugar.
Postcondiciones:	El sistema muestra la lista de lugares	
Flujos alternativos:	No hay flujos alternativos posibles.	



ID: CU_14	Caso de Uso: GenerarRutaMapa	
Breve descripción:	El usuario puede generar una ruta en el mapa de un lugar seleccionado desde su ubicación actual hasta la ubicación del lugar.	
Actores principales:	Usuario	
Precondiciones:	El usuario se encuentra visualizando los detalles de lugar seleccionado.	
Flujo principal:	Paso	Acción
	1 El caso de uso comienza cuando el usuario desea generar una ruta de un lugar. 2 El sistema comprueba que la localización GPS se encuentra activada. 2.1. En caso negativo muestra un aviso al usuario. 3 El sistema busca los datos referentes a ese lugar y los muestra en el mapa. 5 El usuario puede visualizar una ruta desde su posición hasta la del lugar seleccionado. 6 El usuario abandona los mapas de la aplicación.	
Postcondiciones:	No existen postcondiciones	
Flujos alternativos:	No hay flujos alternativos posibles.	



4.4. Matriz relación requisitos-casos de uso

En la siguiente tabla se muestra la relación entre los requisitos funcionales del sistema y los casos de uso definidos. Como se puede apreciar los casos de uso cubren la totalidad de los requisitos funcionales del sistema:

	RF1	RF2	RF3	RF4	RF5	RF6	RF7	RF8	RF9	RF10	RF11	RF12	RF13	RF14	RF15	RF16	RF17	RF18	RF19
CU_1	X	X																	
CU_2			X																
CU_3			X									X	X						
CU_4			X	X	X														
CU_5			X											X	X	X			
CU_6			X						X	X									
CU_7			X																
CU_8			X																
CU_9			X																
CU_10	X	X																	
CU_11																		X	X
CU_12						X	X										X		
CU_13											X								
CU_14								X											

4.5. Modelo de dominio

El modelo de dominio proporciona una representación visual de los conceptos principales de la aplicación y de las relaciones entre éstos. El principal objetivo del modelo de dominio es esclarecer conceptos y facilitar así la comunicación.

En la siguiente figura se muestra de manera esquemática el conjunto de clases y conceptos que se tendrá en cuenta para la fase de diseño y posterior implementación del software.

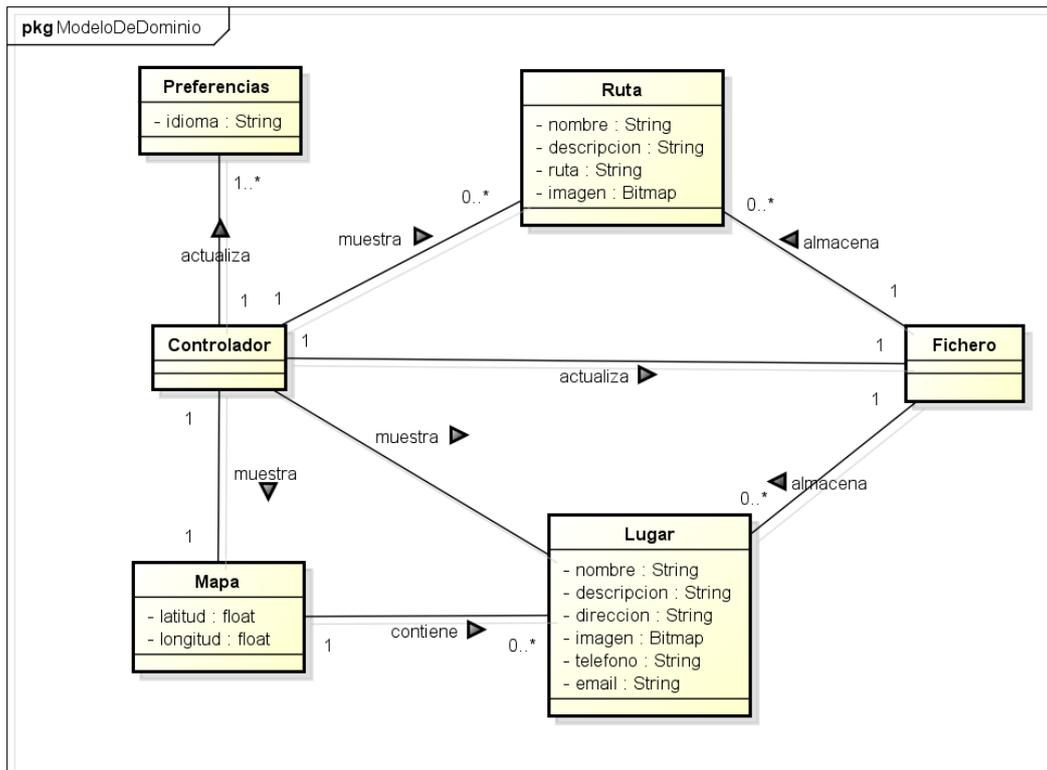


Ilustración 7 Modelo de dominio

4.6. Prototipos

Con el fin de facilitar la fase de diseño e implementación, se han realizado varios prototipos de las partes más relevantes de la aplicación.

1. **Pantalla principal:** en la pantalla principal existirán dos menús, un menú principal con las principales opciones de la aplicación y un segundo menú lateral deslizable donde el usuario podrá ver más funcionalidades que ofrece la aplicación.

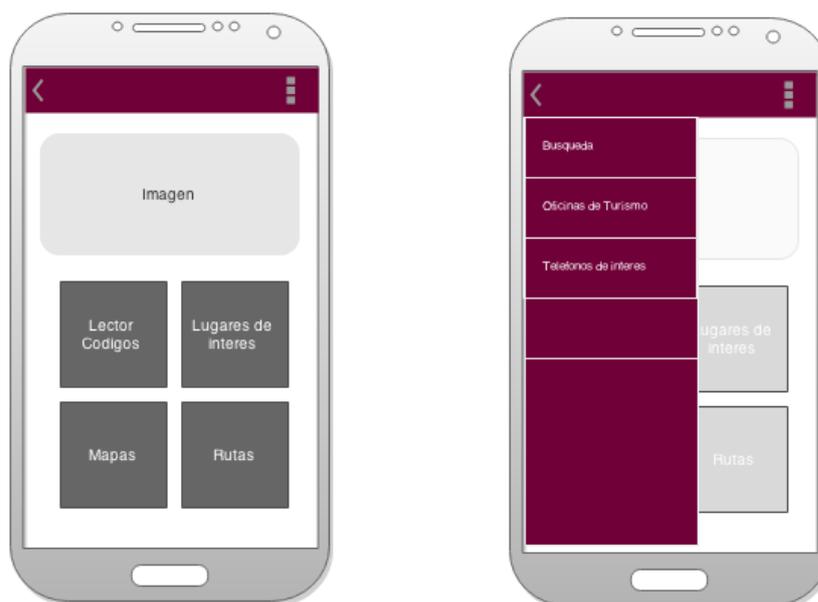


Ilustración 8 Menu principal

- 2. Lista de lugares y rutas:** tanto las lista de lugares y rutas se mostrarán junto con una imagen representativa en caso de que exista y su título.

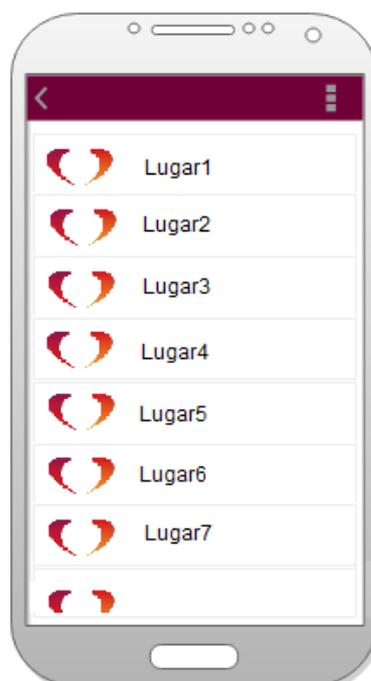


Ilustración 9 Listado de lugares

- 3. Detalle de lugares y rutas:** en el detalle de un lugar o una ruta de mostrará la imagen de éste, en caso de que exista; e información importante acerca de éste.

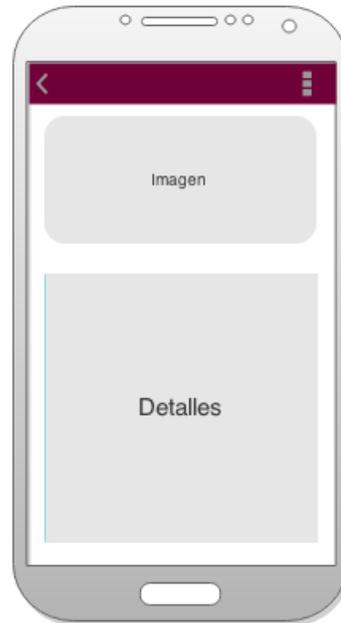


Ilustración 10 Detalle de lugar

- 4. Búsqueda de información:** la pantalla de búsqueda será simple e intuitiva para que el usuario pueda acceder de manera rápida a la información que le interesa. Podrá introducir ciertos filtros para facilitar la labor de búsqueda.



Ilustración 11 Búsqueda

5. **Mapa de los lugares:** en el mapa el usuario podrá ver los monumentos y museos ubicados en un plano y pulsar sobre ellos.

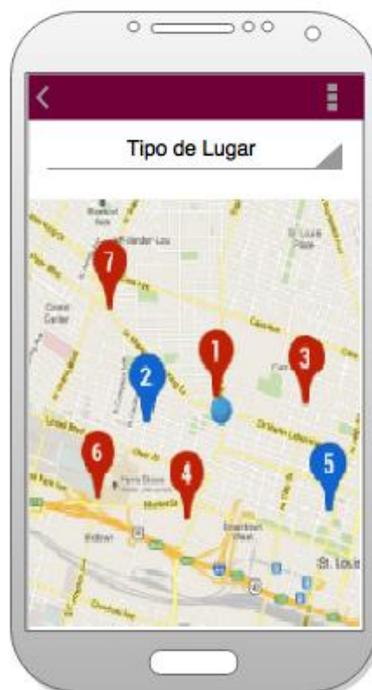


Ilustración 12 Mapas



5. DISEÑO DEL SISTEMA



5.1. Introducción

A continuación se detallarán las decisiones de diseño de la aplicación, es decir la arquitectura física del sistema, la persistencia de datos, el uso de diversos patrones y la interfaz gráfica de usuario.

5.2. Decisiones de diseño

La organización del sistema refleja las decisiones tomadas para estructurar el sistema. Se empleará un modelo Cliente-Servidor.

En la siguiente figura podemos ver la distribución física de las partes de la aplicación:

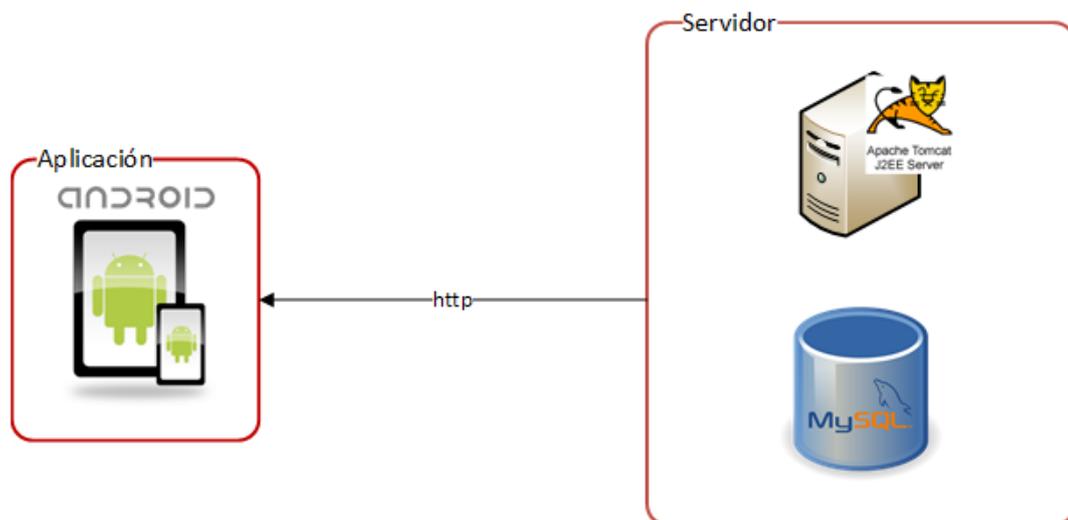


Ilustración 13 Diseño físico del sistema

El sistema se divide en dos componentes bien diferenciados que se comunicarán entre sí:

- El servidor, que albergará una base de datos MySQL y un servidor web. Será el encargado de recoger la información de la base de datos y responder a las peticiones que el cliente realice para proporcionarle esta información.
- La aplicación Android de Turismo, cliente del sistema. La aplicación será la encargada de realizar peticiones al servidor para obtener los datos necesarios, que posteriormente mostrará al usuario.



5.2.1. Topología del sistema

Como se ha descrito en el apartado anterior, el sistema se divide en dos componentes: una aplicación Android como cliente y un servidor. Ambos deben comunicarse entre sí y transferirse información.

A continuación vamos a exponer con más detalle las decisiones de diseño tomadas.

En la siguiente figura se muestra la arquitectura física del sistema:

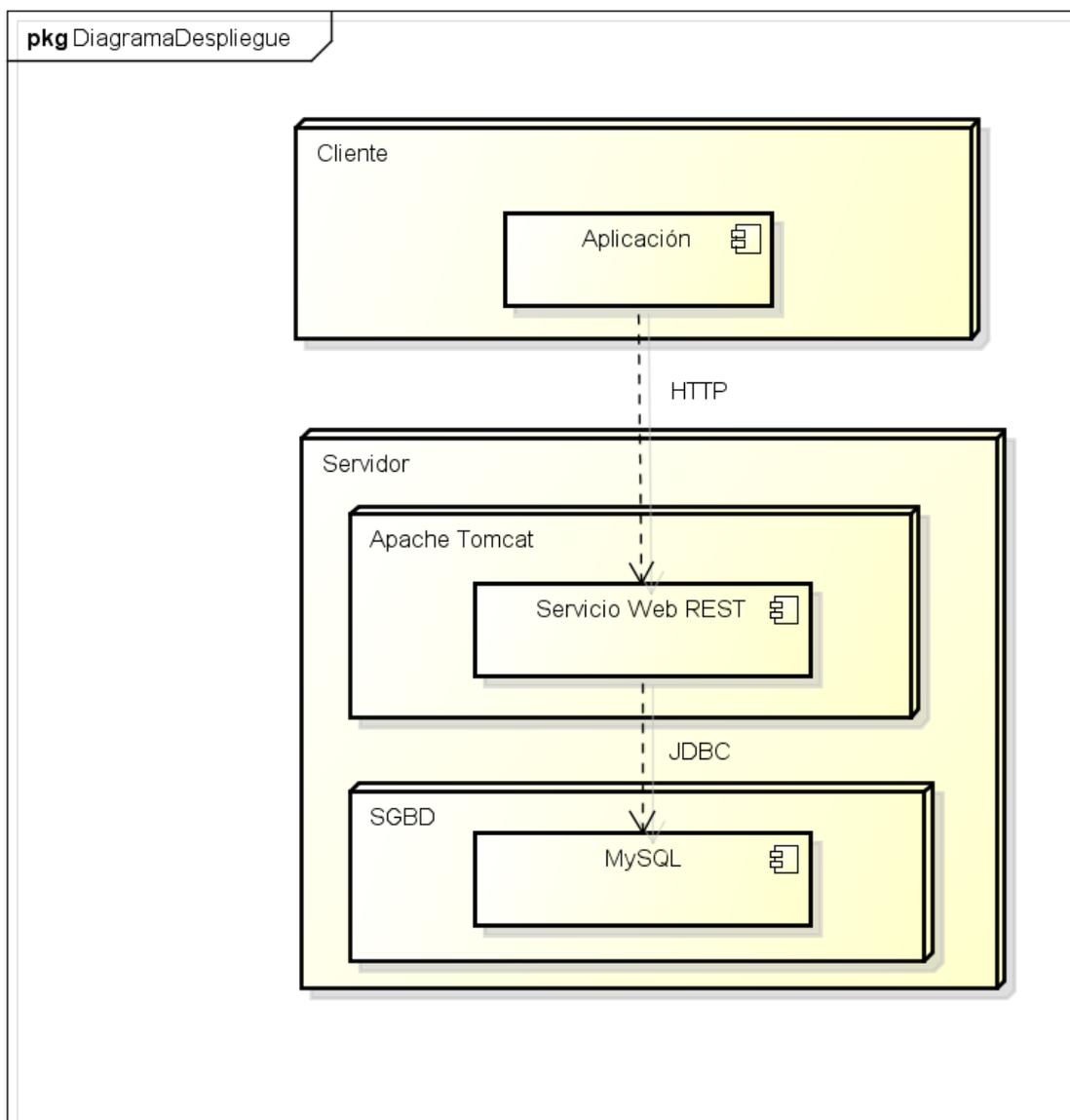


Ilustración 14 Topología del sistema



La arquitectura lógica del sistema está dividida en tres capas:

- En el cliente nos encontramos con la aplicación Android que será la encargada de procesar la información y mostrársela al usuario.
- En el lado del servidor nos encontramos con los otros dos niveles:
 - *El servidor web, donde estará alojado el servicio web que se comunicará con la base de datos.* El servidor será un Apache Tomcat, un servidor con soporte para servlets y JSPs. A pesar de que Apache Tomcat no es un servidor de aplicaciones puede utilizarse como servidor web por sí mismo, por lo que no habrá problemas a la hora de publicar nuestro servicio web.
Nuestro servicio web, tipo REST, será el encargado de proporcionar la información necesaria a la aplicación móvil. Éste se comunicará por un lado con la base de datos mediante JDBC (Java Database Connectivity) y por otro lado con la aplicación Android mediante el protocolo HTTP.
 - El sistema de gestión de base datos, encargada del almacenamiento de los datos. Se detallará más adelante.

5.2.2. Comunicación entre aplicación y base datos.

La comunicación entre ambas partes del sistema es esencial para el caso que nos ocupa, puesto que el cliente no puede funcionar correctamente sin los datos que le proporciona el servidor, por esta razón este apartado.

Esta comunicación se realizará través de un servicio web que hará de intermediario de éstos para obtener los datos requeridos.

Este servicio web será el encargado de proporcionar la información requerida a la aplicación Android. El servicio web será tipo REST, y la comunicación solo se realizará en una dirección, es decir solo se consumirán datos del servidor sin insertar ni modificar información existente.

Los servicios REST a menudo hacen uso de estándares basados en texto para el intercambio de los datos, los más comunes son: XML (*Extensible Markup Language*) y JSON (*JavaScript Object Notation*). Java proporciona librerías para manipular ambos estándares tales como SAX (*Simple API for XML*) o DOM (*Modelo de Objetos de Documentos*) para el caso del primero, y JSON.simple o GSON para el caso del segundo.

La lectura de ficheros JSON y su posterior conversión en objetos de nuestra aplicación se realiza de una manera mucho más rápida que mediante el tradicional XML, una ventaja importante para el caso que nos ocupa debido al manejo de grandes cantidades de datos. Por lo que el servicio web nos devolverá la información por medio de un fichero JSON.

El servicio web estará implementado en el lenguaje de programación Java debido al tipo de servidor sobre el que estará publicado. Para ello utilizaremos el framework Jersey.

Project Jersey es un framework de código libre para el desarrollo de servicios web basados en REST. Hace uso de la API JAX-RS (*Java API for RESTful Web Services*) que proporciona soporte para la creación de servicios web.



5.2.3. Persistencia de datos.

Como se ha hecho referencia anteriormente, la persistencia del sistema será gestionada por el Sistema Gestor de Bases de Datos de MySQL. La Base de Datos almacena todos los datos que nuestra aplicación Android mostrará al usuario. Estos datos se almacenarán en un servidor remoto al que la aplicación deberá conectarse.

También se deberá tener en cuenta que Android utilizará otros tipos de mecanismos alternativos para almacenar datos propios.

5.2.3.1. Datos a almacenar

A continuación se indican los datos que deben almacenarse en la aplicación Android:

Datos de turismo:

Se deberá almacenar toda la información de datos turísticos de Valladolid con datos detallados, tales como:

- Monumentos.
- Lugares y museos.
- Rutas.
- Otros datos de interés como: oficinas de turismo, teléfonos, etc.
- Otros contenidos.

Datos y parámetros de configuración:

Se deberá almacenar los datos de configuración que el usuario elija. En nuestro caso solo almacenaremos el idioma con el que se quiera ejecutar la aplicación.

5.2.3.2. Decisiones de almacenamiento

Android proporciona varios mecanismos de almacenamiento. A continuación hacemos referencia a tres de ellos que pueden ser de utilidad para el caso que nos ocupa:

- Base de datos: Android proporciona soporte para bases de datos mediante el motor SQLite, que permite almacenar estructuras en una base de datos privada de la aplicación.
- Preferencias: (*Shared Preferences*) Mecanismo liviano que permite almacenar y recuperar datos primitivos en la forma de pares clave/valor. Utilizado normalmente para almacenar los parámetros de configuración de una aplicación.
- Ficheros: Se pueden almacenar ficheros en la memoria interna del dispositivo o en un medio de almacenamiento removible como una tarjeta SD.

El sistema almacenará el idioma elegido por el usuario, que será con el que debe ejecutarse la aplicación. Para ello se ha decidido hacer uso del sistema de preferencias que proporciona Android ya mencionado.

Para el almacenamiento de datos relativos a monumentos, rutas, museos, etc, se ha decidido hacer uso de ficheros. Los datos recuperados mediante el servicio web se almacenarán en formato JSON en la carpeta propia de la aplicación. La velocidad de escritura y lectura puede ser un cuello de botella a la hora de ejecutar la aplicación, por lo que es un punto importante que debe tenerse en cuenta. Si hacemos una comparación entre los dos mecanismos SQLite y JSON no poseen grandes diferencias en lo que se refiere a la lectura de datos, pero si a la hora de la escritura. Otro punto importante a la hora de escoger el uso de ficheros JSON ha sido la facilidad de conversión de los datos en objetos Java de los que la aplicación Android hará uso.

5.3. Patrones de Diseño

Los patrones de diseño nos proporcionan soluciones generales para problemas concretos. Su uso es recomendable para el desarrollo de aplicaciones robustas y de fácil mantenimiento.

5.3.1. MVC

El desarrollo de aplicaciones Android sigue el patrón Modelo Vista Controlador (MVC). Este patrón separa los datos, la interfaz de usuario y la lógica de la aplicación en tres componentes que se relacionarán entre sí. El MVC permite de manera sencilla añadir nuevas funcionalidades futuras a la aplicación. Esencialmente el comportamiento del modelo se puede ver en la siguiente figura:

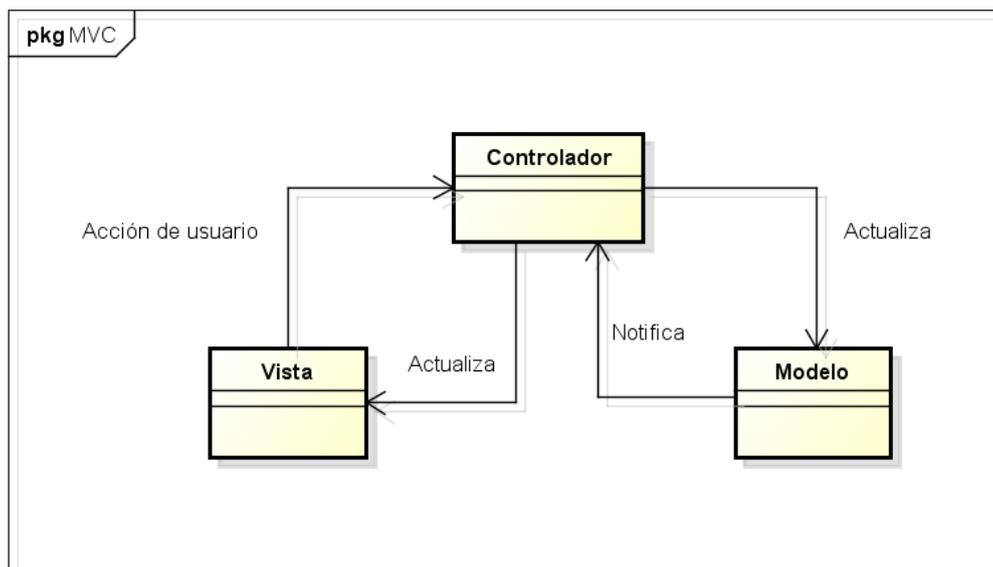


Ilustración 15 MVC

Nos encontramos con tres componentes con funciones bien diferenciadas entre sí:

- Modelo: son las representaciones de la información con la que trabaja la aplicación. Estas vistas implementan la lógica de la misma. El modelo se creará a partir de los ficheros JSON almacenados en la aplicación o los datos recuperados mediante el servicio web.
- Vista: es la interfaz con la interactúa el usuario. Las vistas se implementan mediante XML.
- Controlador: es el núcleo de la aplicación, interactúa tanto con el modelo como con la vista. Manejan la información y la despliegan para que la vista pueda mostrarla. Los controladores son propiamente las actividades creadas en Java.

El flujo de control general es el siguiente:

1. El usuario interactúa con la vista.
2. El controlador recibirá las solicitudes de acción del usuario y hará las gestiones oportunas.
3. El controlador actualiza el modelo de acuerdo a las acciones de usuario.
4. La vista esperará nuevas interacciones de usuario, comenzando de nuevo el ciclo.

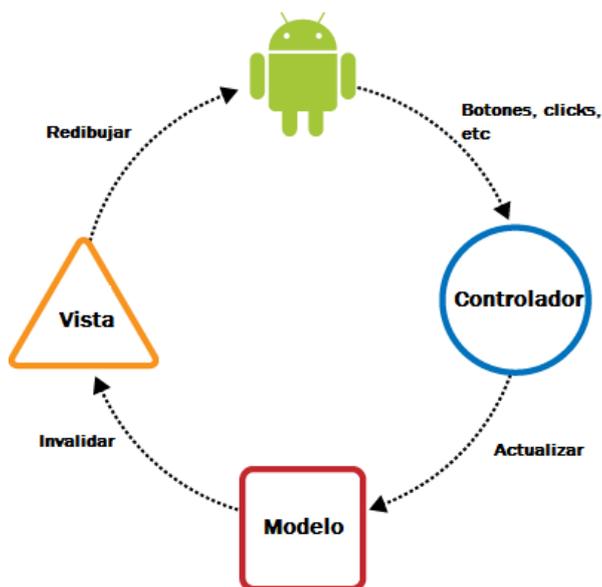


Ilustración 16 MVC Android



5.4. Interfaz de usuario.

La importancia de una interfaz de usuario de fácil manejo y atractiva al usuario hace que este punto cobre importancia en nuestro proyecto.

Para el diseño de la interfaz se han tenido en cuenta los siguientes puntos:

- Facilidad y comprensión: además de iconos se utilizará textos para informar al usuario de los elementos del menú principal. La interfaz y el diseño de navegación de la aplicación debe ser lo suficientemente sencilla para que el usuario sea capaz de acceder a todas las partes de la aplicación por sí mismo y con un número reducido de pasos.
- La navegación ha de ser intuitiva para el usuario. Para ello se utilizarán las líneas generales de diseño de aplicaciones Android, como un menú desplazable a la izquierda o un menú de opciones en la barra superior de la aplicación.
- Diseño homogéneo: evitar el abuso de colores, centrándose en una gama básica que identifique nuestra aplicación. El uso de iconos también es importante, por lo tanto se deberá cuidar que éstos sean del mismo tipo, aspecto y tamaño.
- Facilidad de interacción: cuando partes del sistema, así como botones, opciones de selección y controles similares requieran la intervención del usuario, éstos serán diseñados correctamente y de manera que el usuario sepa cómo utilizarlos.
- Diseño responsivo: el diseño de la aplicación deberá adaptarse a diferentes tamaños de pantallas. Este tema se tratará en mayor profundidad en el siguiente apartado.

5.4.1. Adaptación al tamaño de pantalla

Debido a la gran variedad de dispositivos Android existentes en el mercado actual, se ha considerado importante que la interfaz gráfica pueda ser capaz de adaptarse a todos los posibles de tamaño existentes. Android nos proporciona ciertas facilidades para lograr aplicaciones responsivas.

Para ello se deben diseñar diferentes layouts en función del tamaño de pantalla deseado. Estos layouts se almacenarán en carpetas con nombres predefinidos que dependen de ciertos atributos como pueden ser el tamaño o la densidad de la pantalla.

El propio Android será el encargado de utilizar el layout que mejor se ajuste a las características del dispositivo que está ejecutando la aplicación, de esta forma el coste de desarrollar una aplicación adaptada cualquier tamaño de pantalla en mínima.

La aplicación de turismo debe adaptarse a pantallas de dimensiones desde 3 hasta 10.1 pixeles, que son las tablets habituales que nos encontramos en el mercado. La adaptación de la aplicación a distintos tamaños de pantalla se tratará dividiendo los diseños de las vistas en tres carpetas. De acuerdo con la documentación oficial de Android, serán las siguientes:

- *layout-xlarge*: para pantallas desde 7 hasta 10.1 pulgadas.
- *layout-large*: para pantallas entre 4 y 7 pulgadas.
- *layout*: el resto de tamaños utilizarán las vistas de esta carpeta.

Además de esta división, las imágenes e iconos utilizados en la aplicación tendrán diferentes tamaños en función de la densidad de pantalla del dispositivo. Para generar los iconos se utilizará la herramienta *Android Assset Studio*. Este conjunto de herramientas basadas en web de código abierto permite generar archivos como gráficos que las aplicaciones Android utilizarán en su directorio de recursos (*res*). Los gráficos se almacenarán en los siguientes directorios:

- *drawable-mdpi*
- *drawable-xdpi*
- *drawable-xhdpi*
- *drawable-xxdpi*

En la siguiente figura se puede ver la división categorizada por diferentes tamaños de pantalla y densidades que ofrece la documentación oficial de Android:

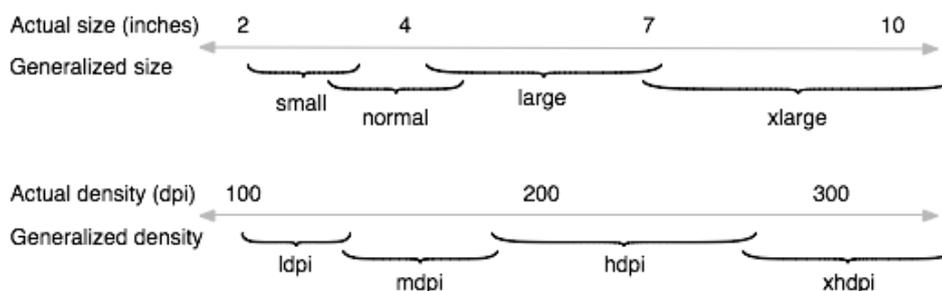


Ilustración 17

5.5. Diagrama de paquetes

En la siguiente figura se muestra el sistema dividido en paquetes con el objetivo de facilitar la comprensión del mismo.

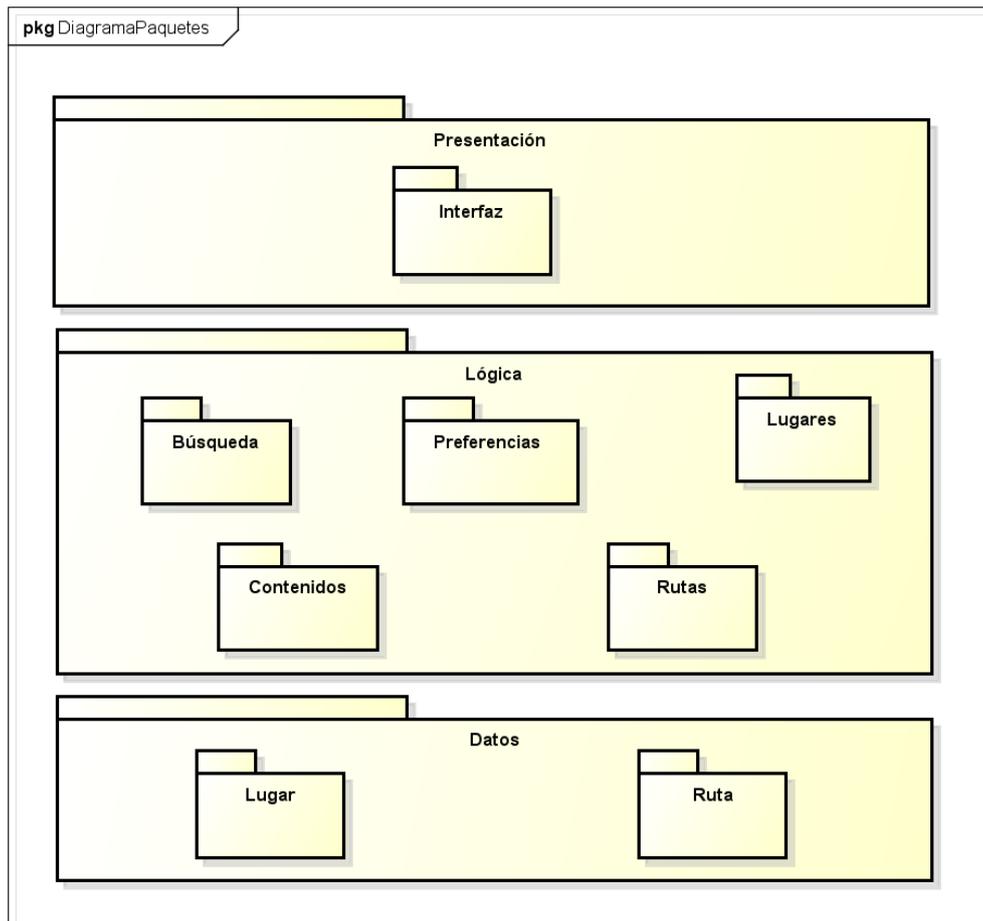


Ilustración 18 Diagrama de paquetes



5.6. Diagramas de secuencia de diseño

En este apartado se llevará a cabo la realización de los diagramas de secuencia del sistema. Se han realizado los siguientes:

Diagrama de secuencia	Figura
<i>IniciarAplicación</i>	19
<i>VerPreferencias</i>	20
<i>RealizarBúsqueda</i>	21
<i>VerMapa</i>	22
<i>VerRutas</i>	23
<i>VerLugares</i>	24
<i>VerOficinas</i>	25
<i>VerTeléfonosInteres</i>	26
<i>VerContenido</i>	27
<i>DescargaDatos</i>	28
<i>CambiarIdioma</i>	29
<i>VerLugarDetalle</i>	30
<i>VerRutaDetalle</i>	31
<i>GenerarRutaMapa</i>	32

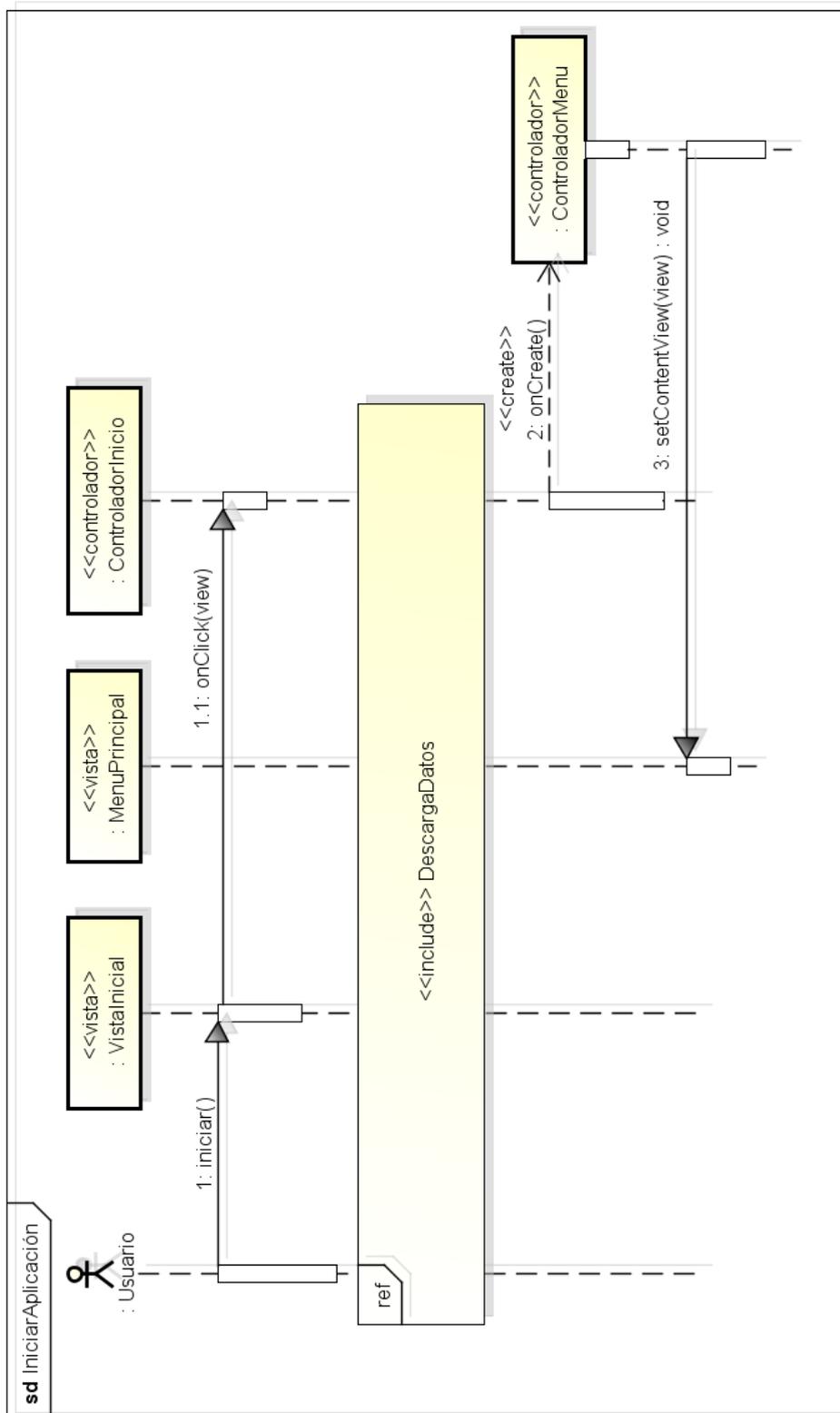


Ilustración 19 DSS IniciarAplicacion

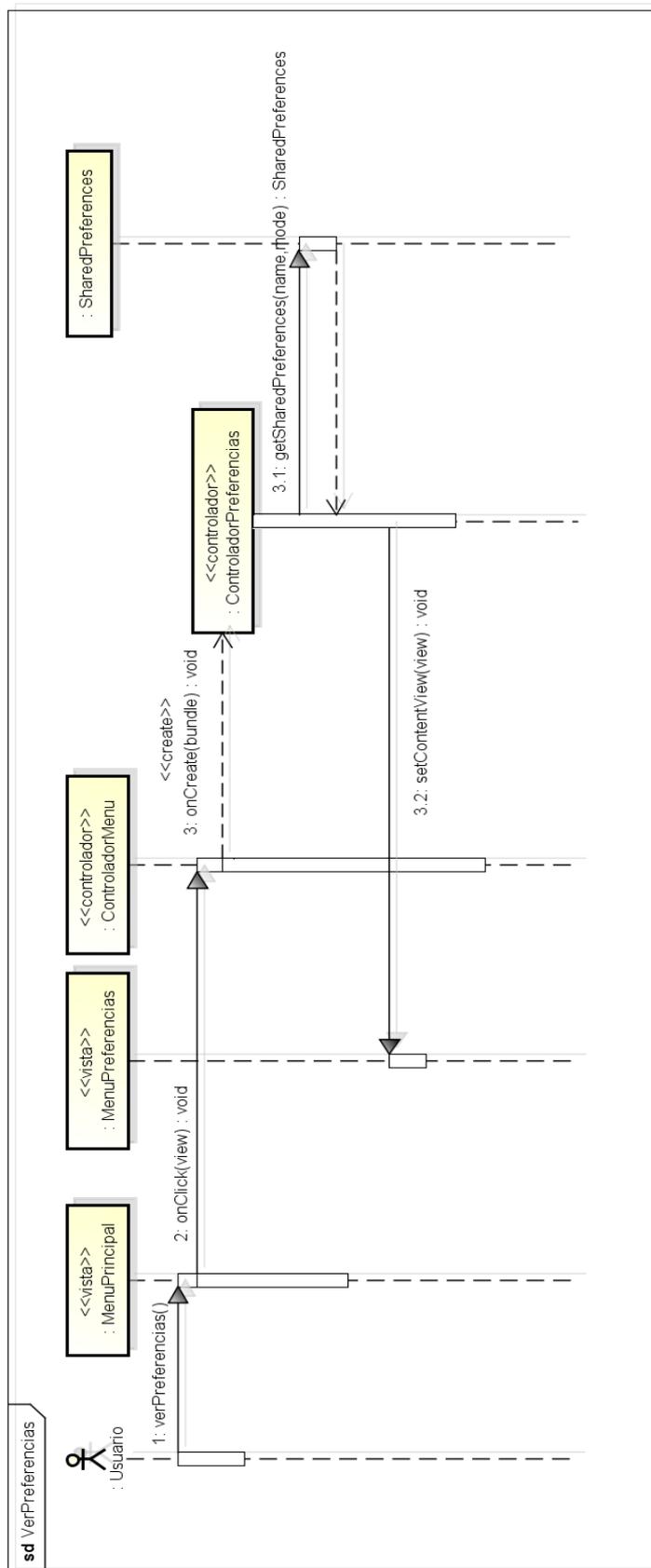


Ilustración 20 DSS VerPreferences

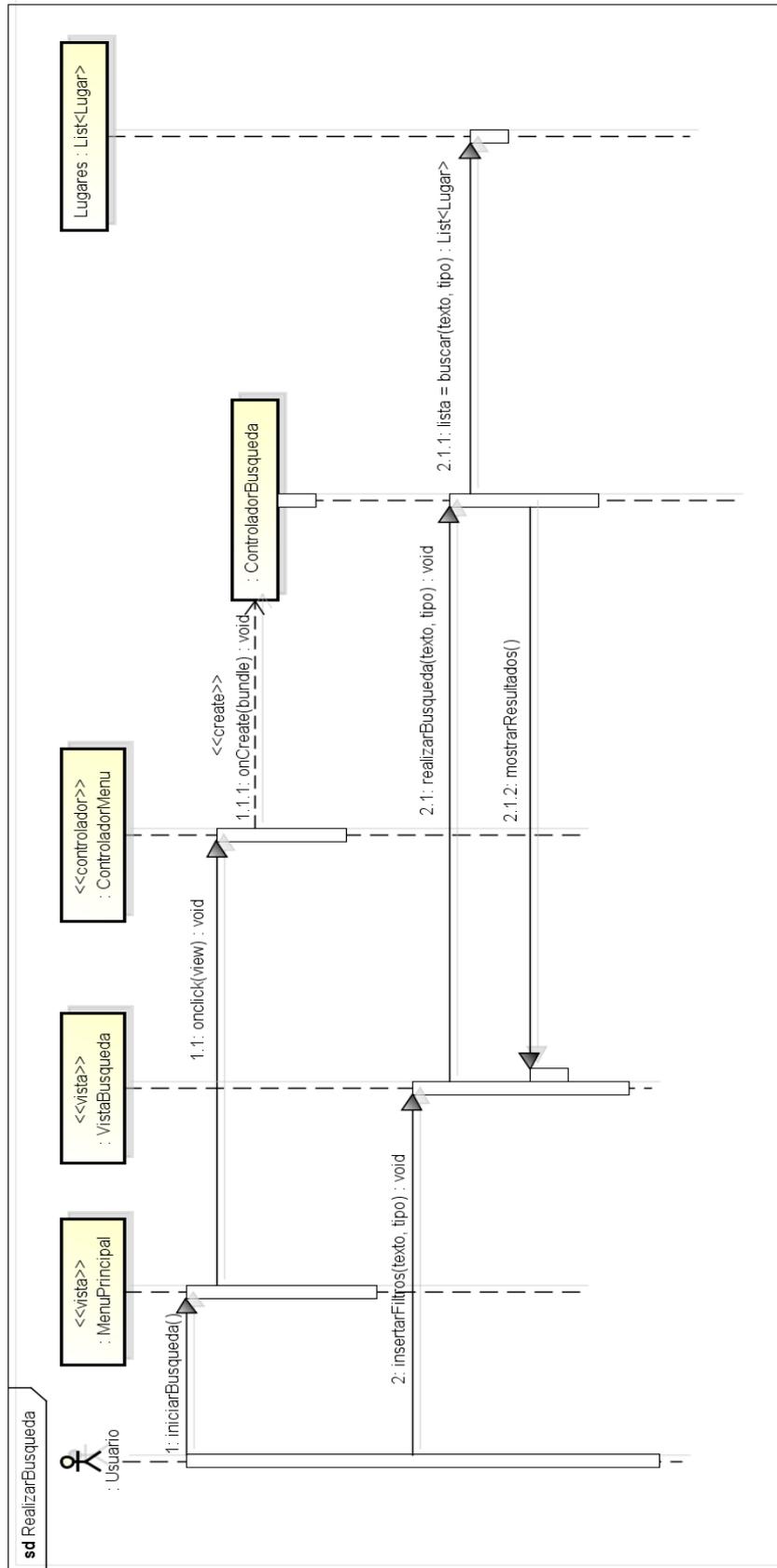


Ilustración 21 DSS RealizarBusqueda

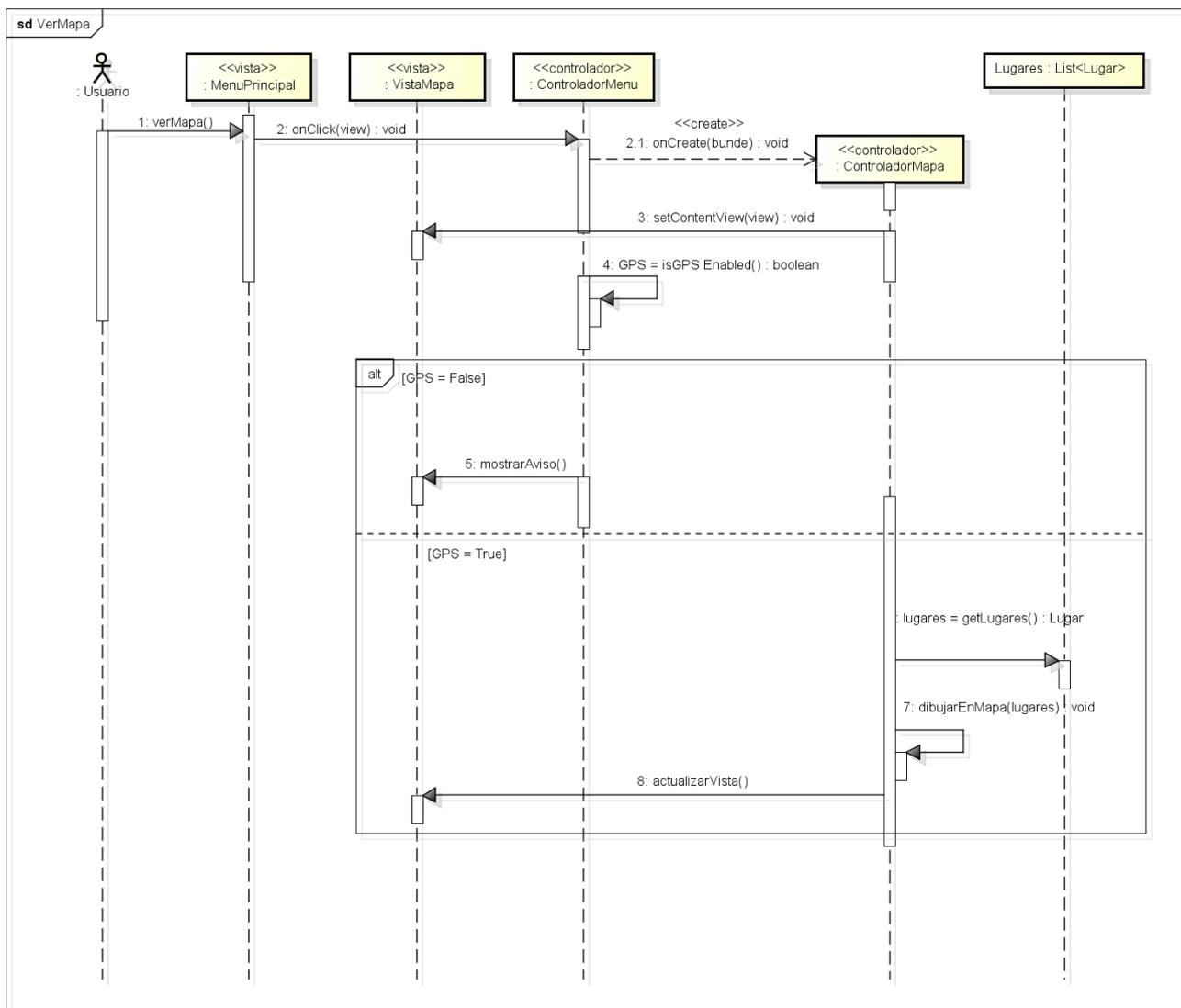


Ilustración 22 DSS Ver Mapa

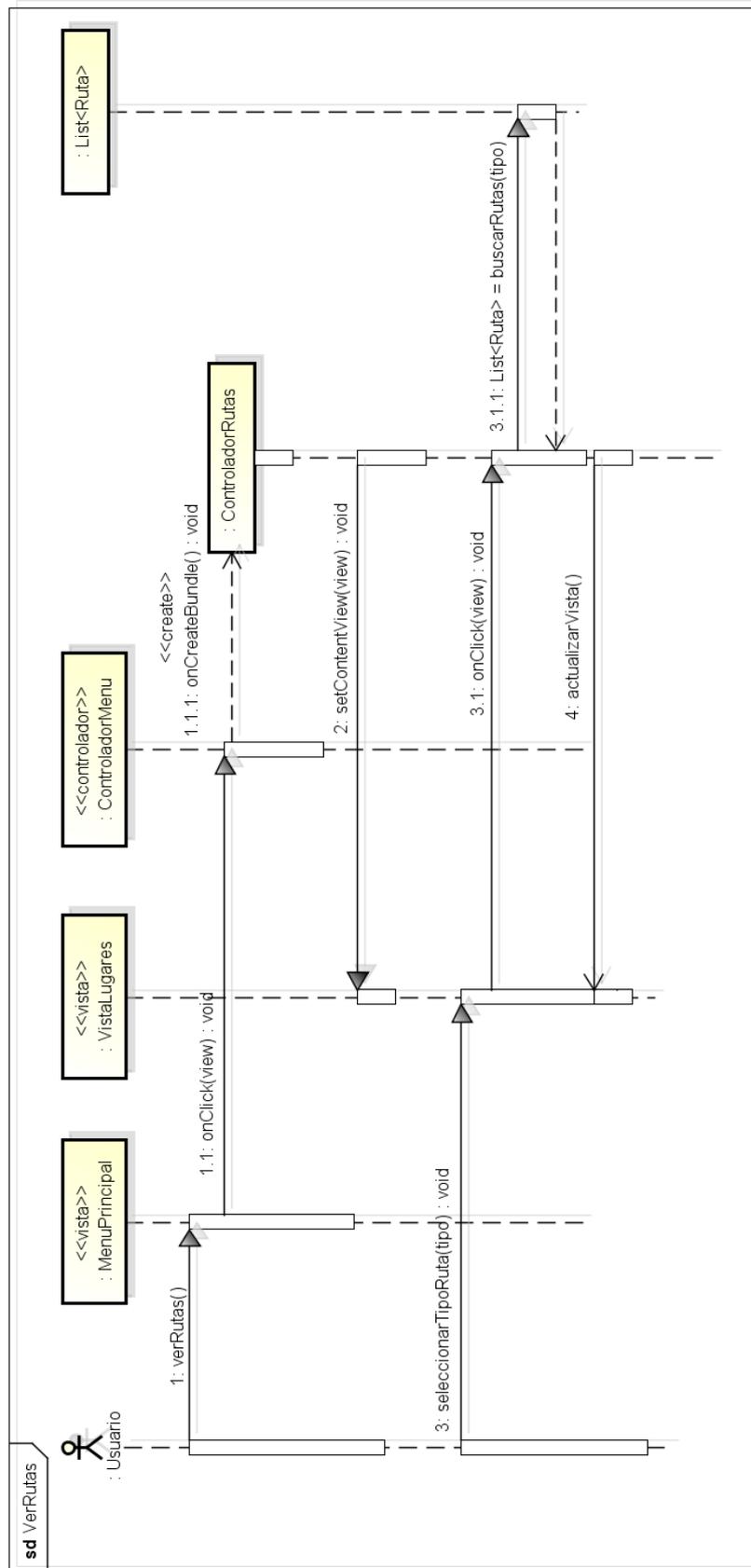


Ilustración 23 DSS VerRutas

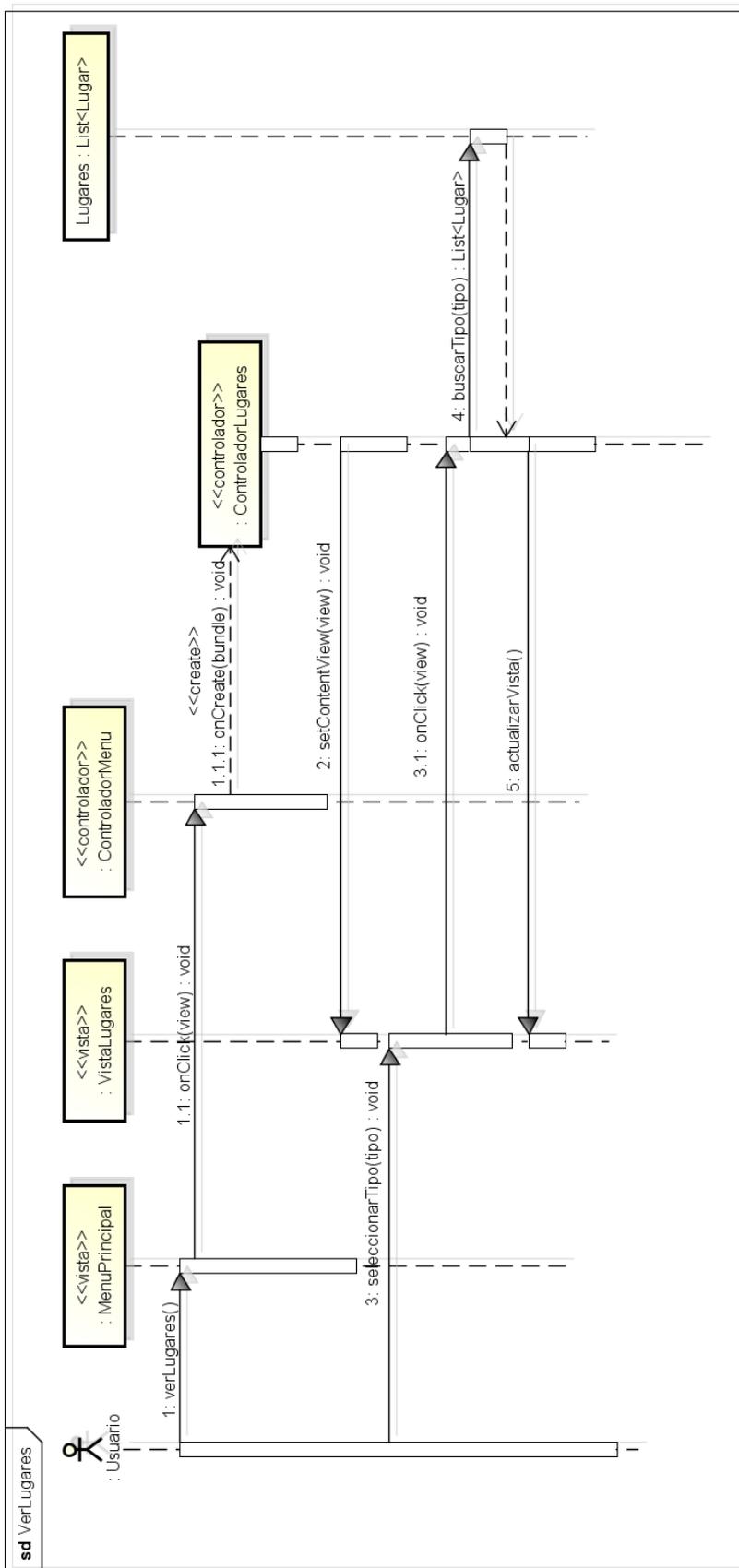


Ilustración 24 DSS VerLugares

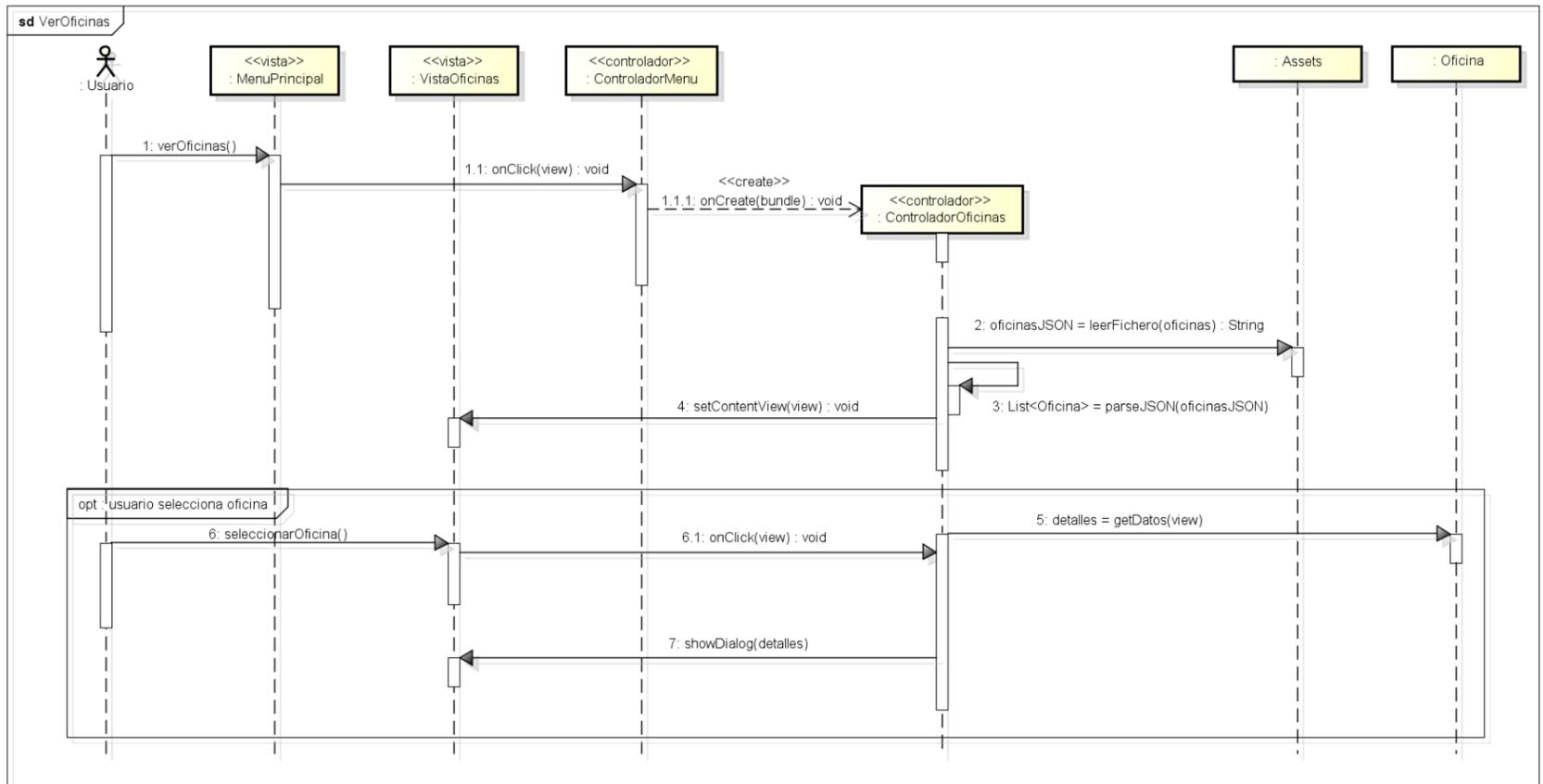


Ilustración 25 DSS VerOficinas

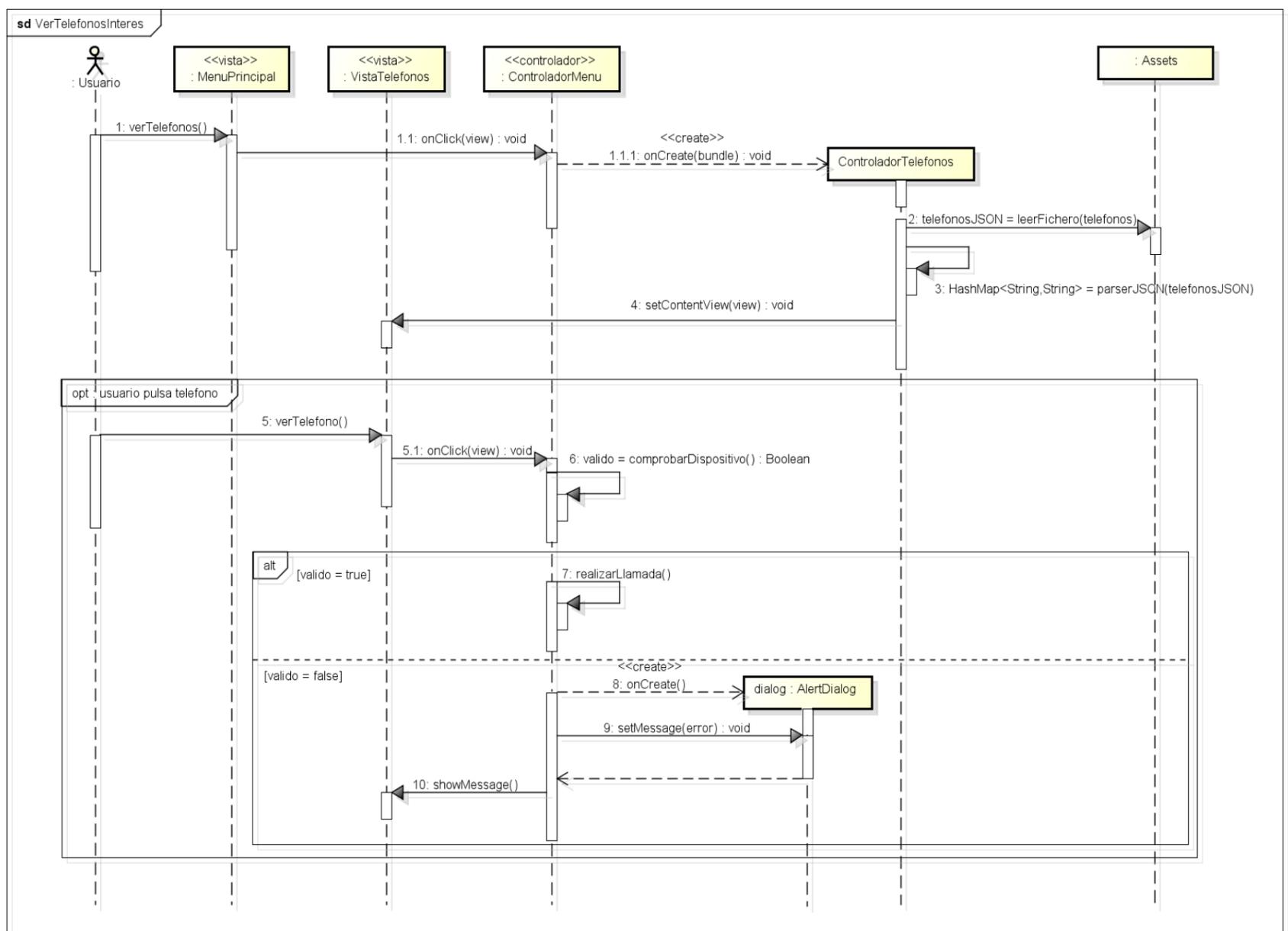


Ilustración 26 DSS VerTelefonosInteres

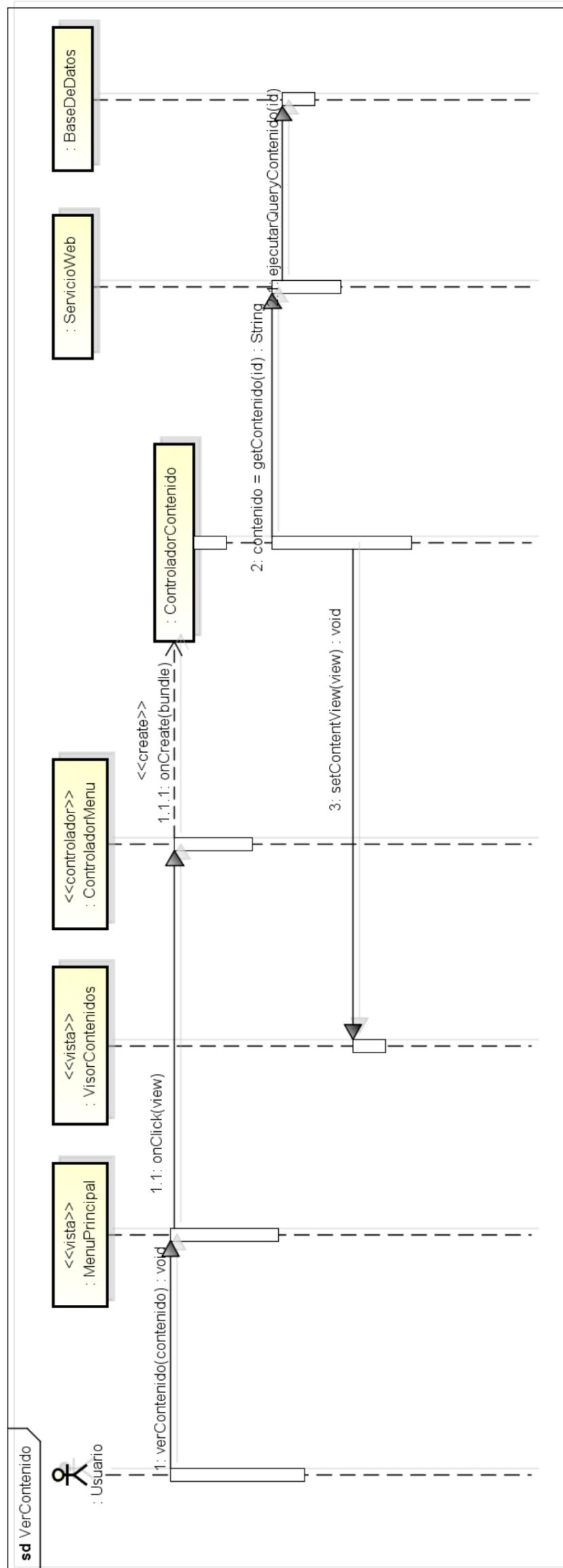


Ilustración 27 DSS VerContenido

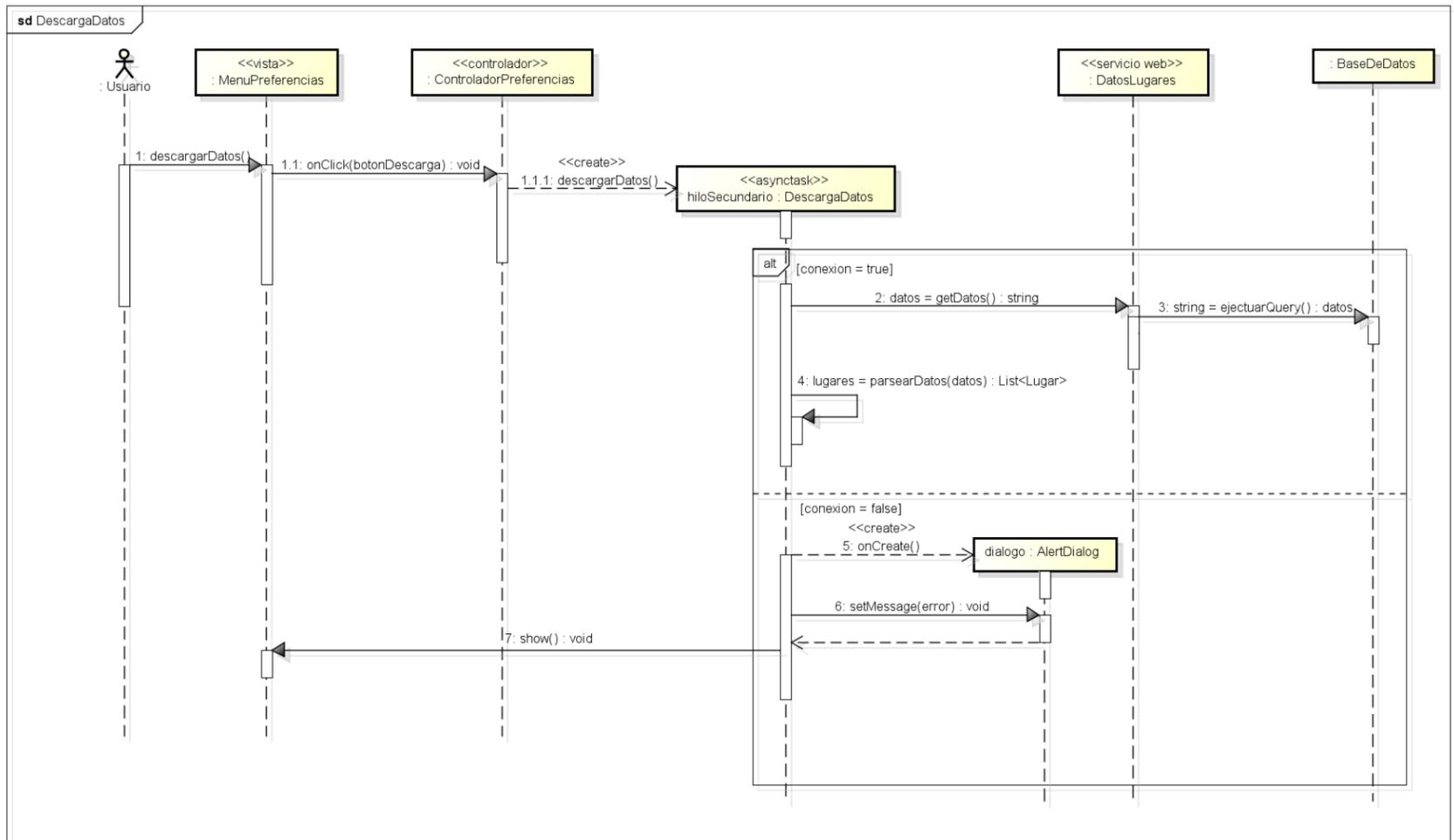


Ilustración 28 DSS DescargaDatos

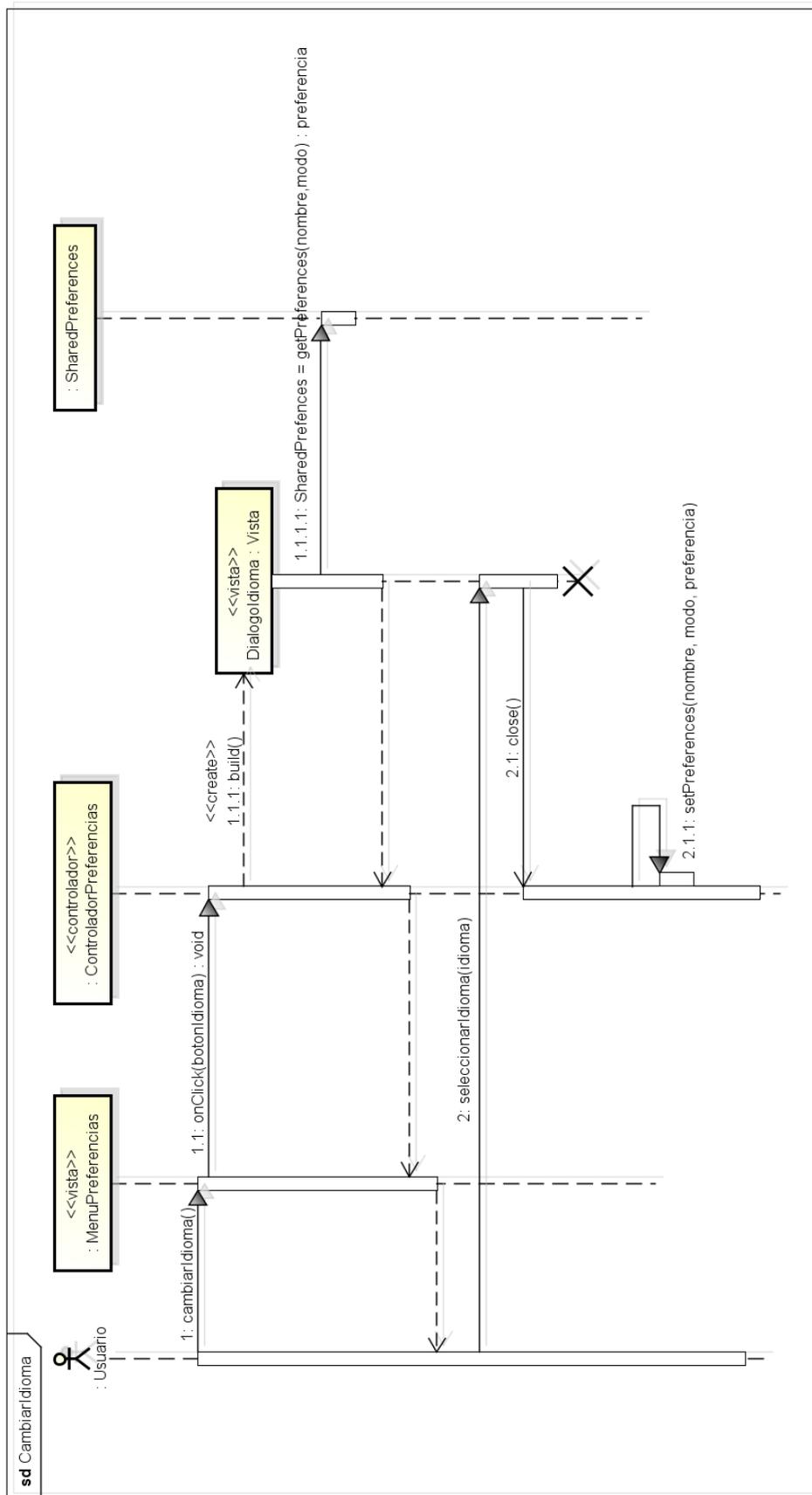


Ilustración 29 DSS CambiarIdioma

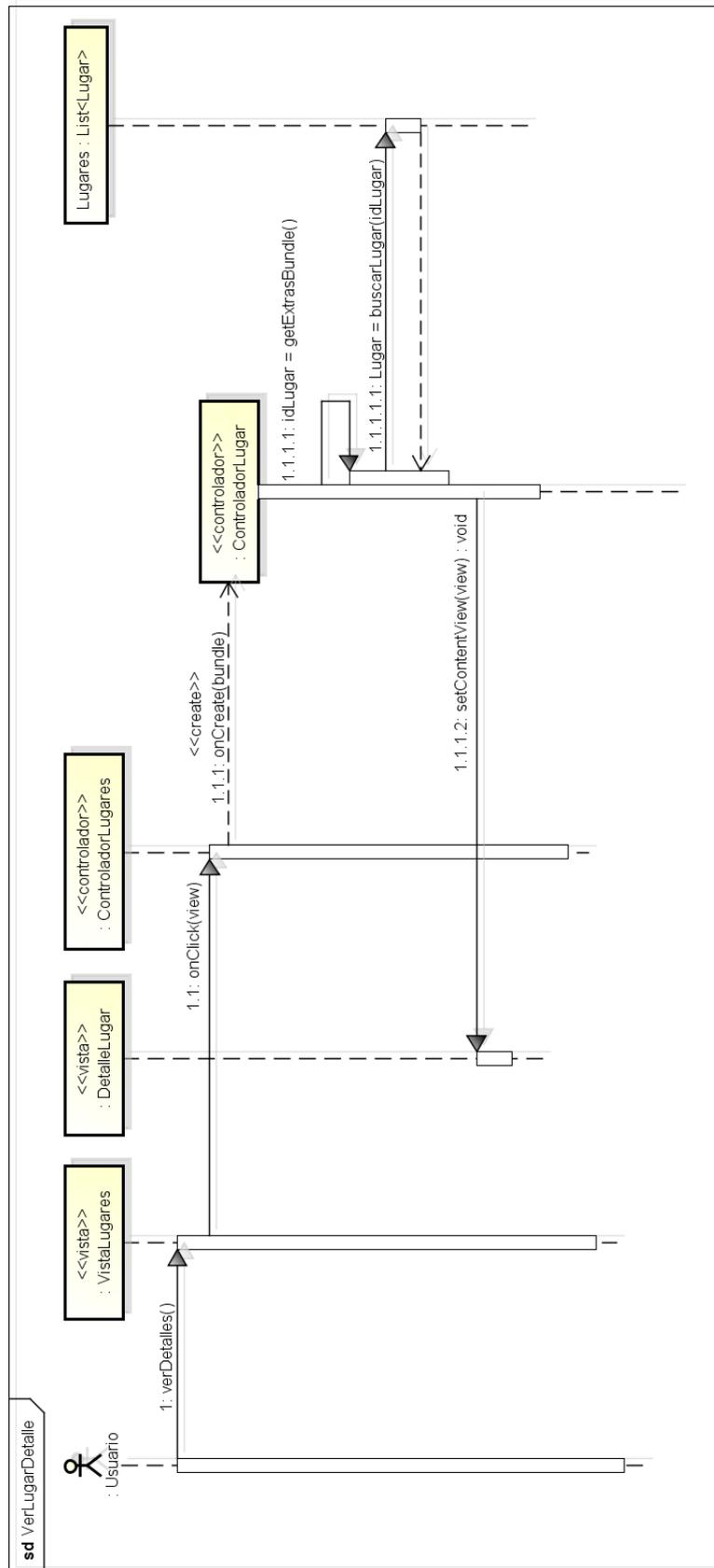


Ilustración 30 DSS VerLugarDetalle

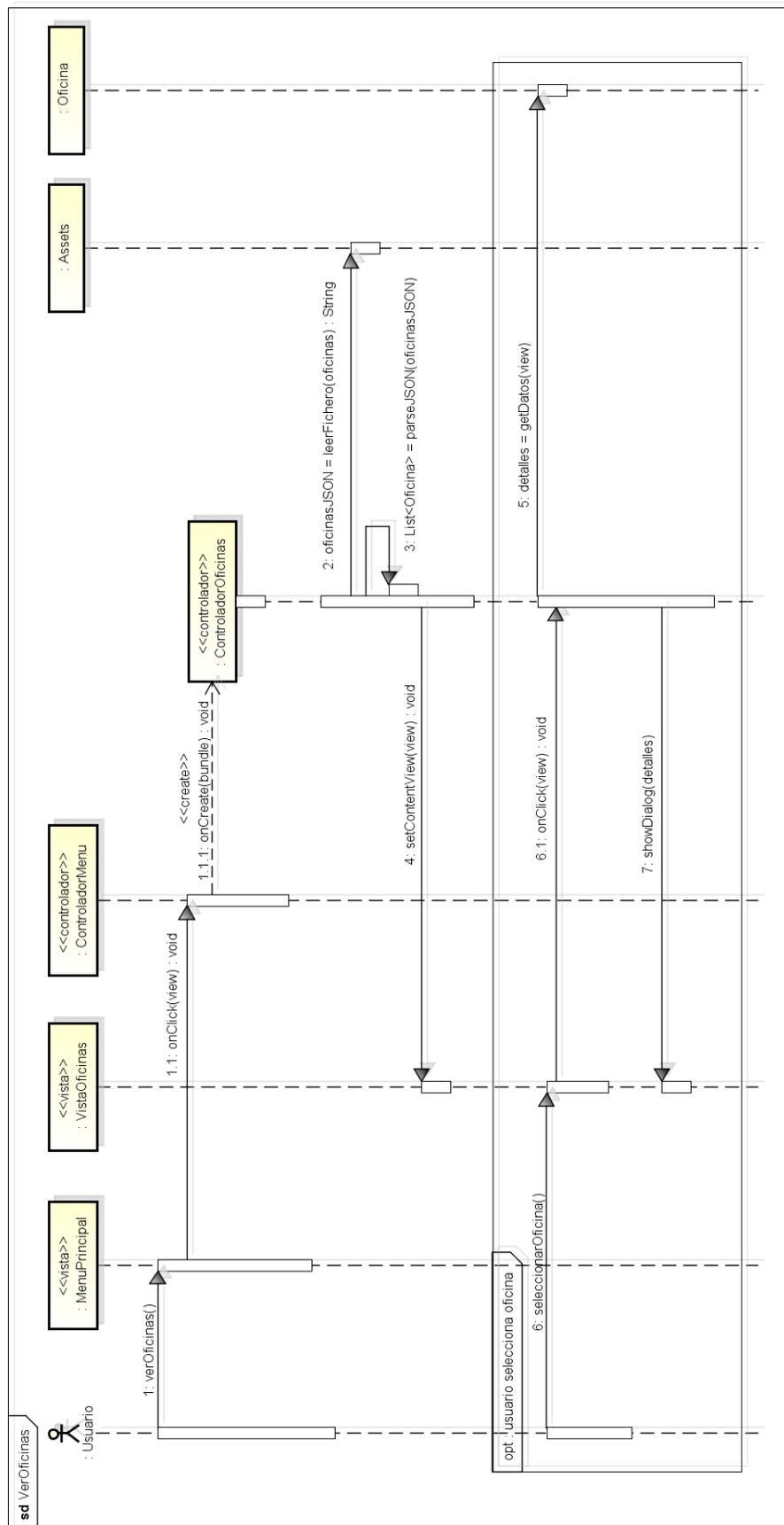


Ilustración 31 DSS VerRutaDetalle

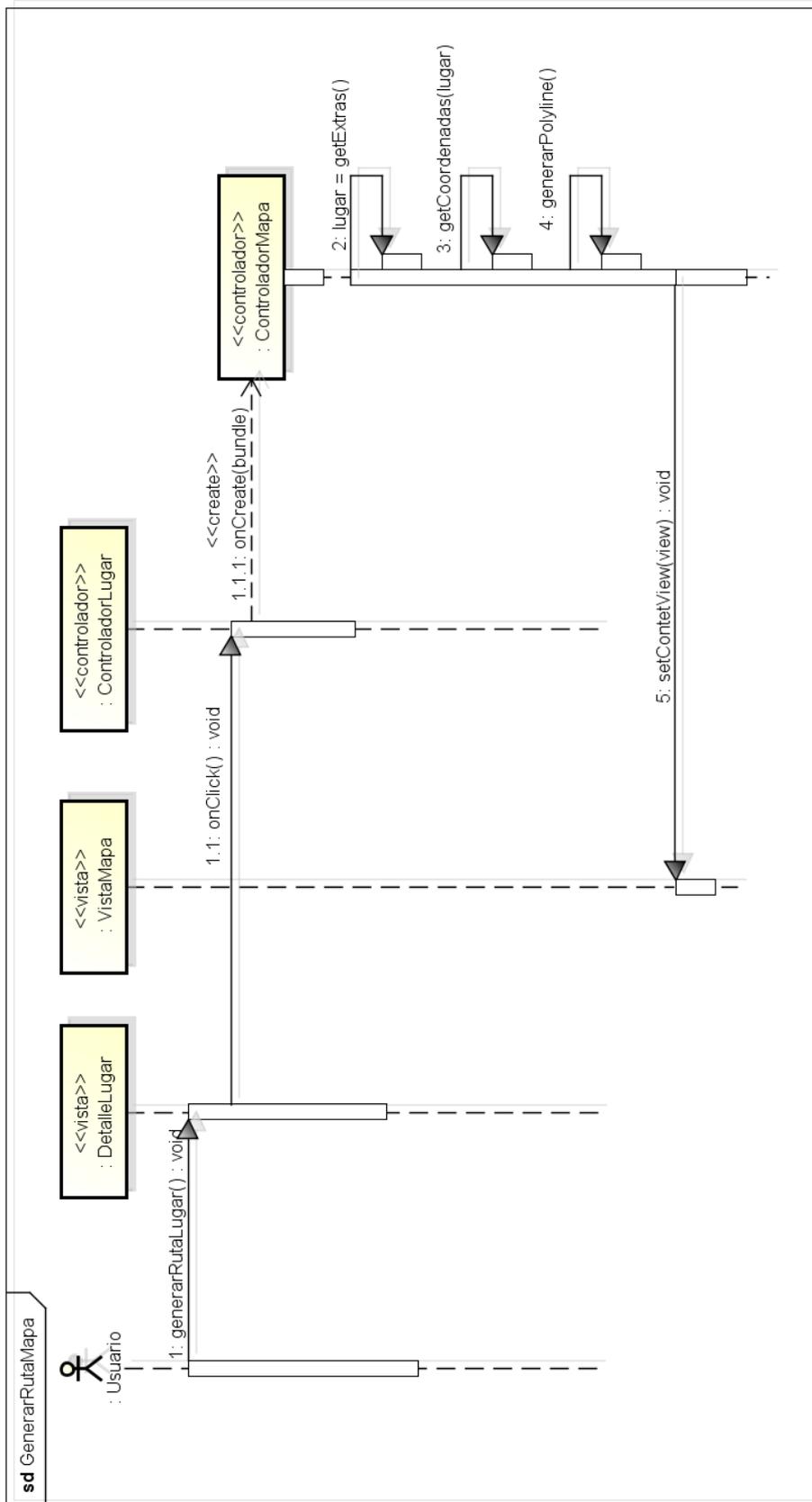


Ilustración 32 DSS GenerarRutaMapa



5.7. Diagrama de clases de diseño

En este apartado se mostrarán los diagramas de clases correspondientes a cada paquete en los que se ha dividido la aplicación.

Paquete	Figura
<i>Preferencias</i>	33
<i>Lugares</i>	34
<i>Contenidos</i>	35
<i>Rutas</i>	36

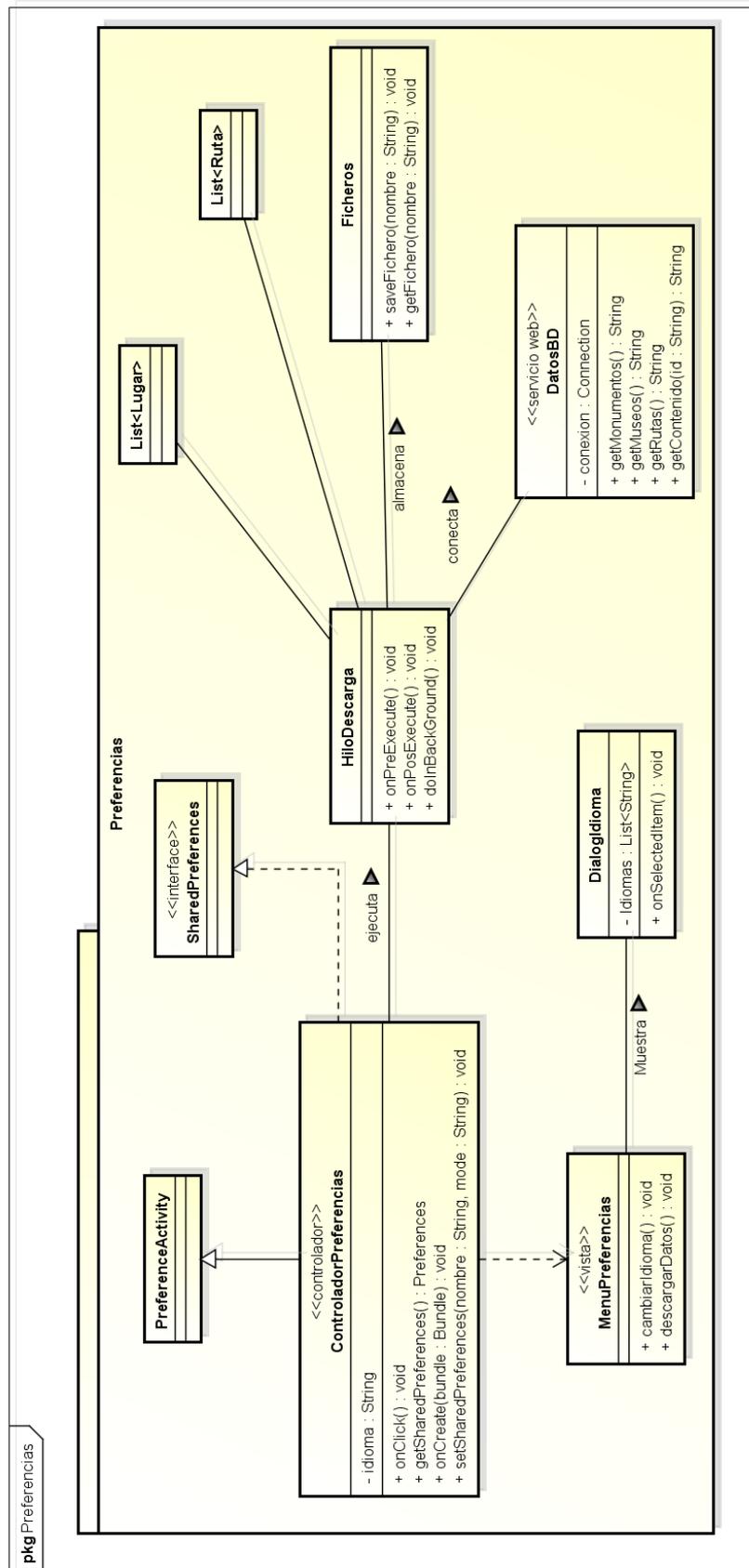


Ilustración 33 Paquete de Preferencias

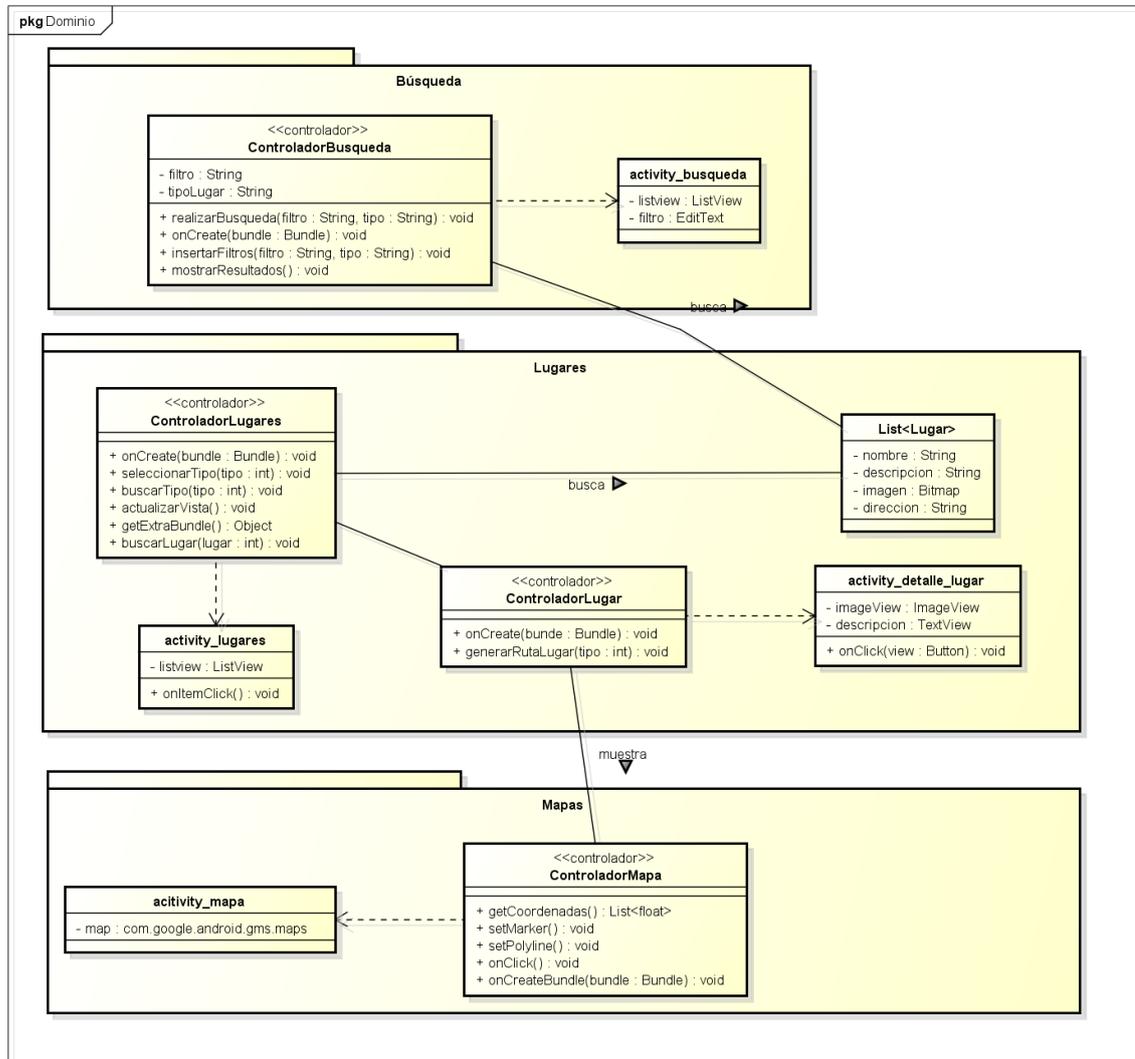


Ilustración 34 Paquete Lugares

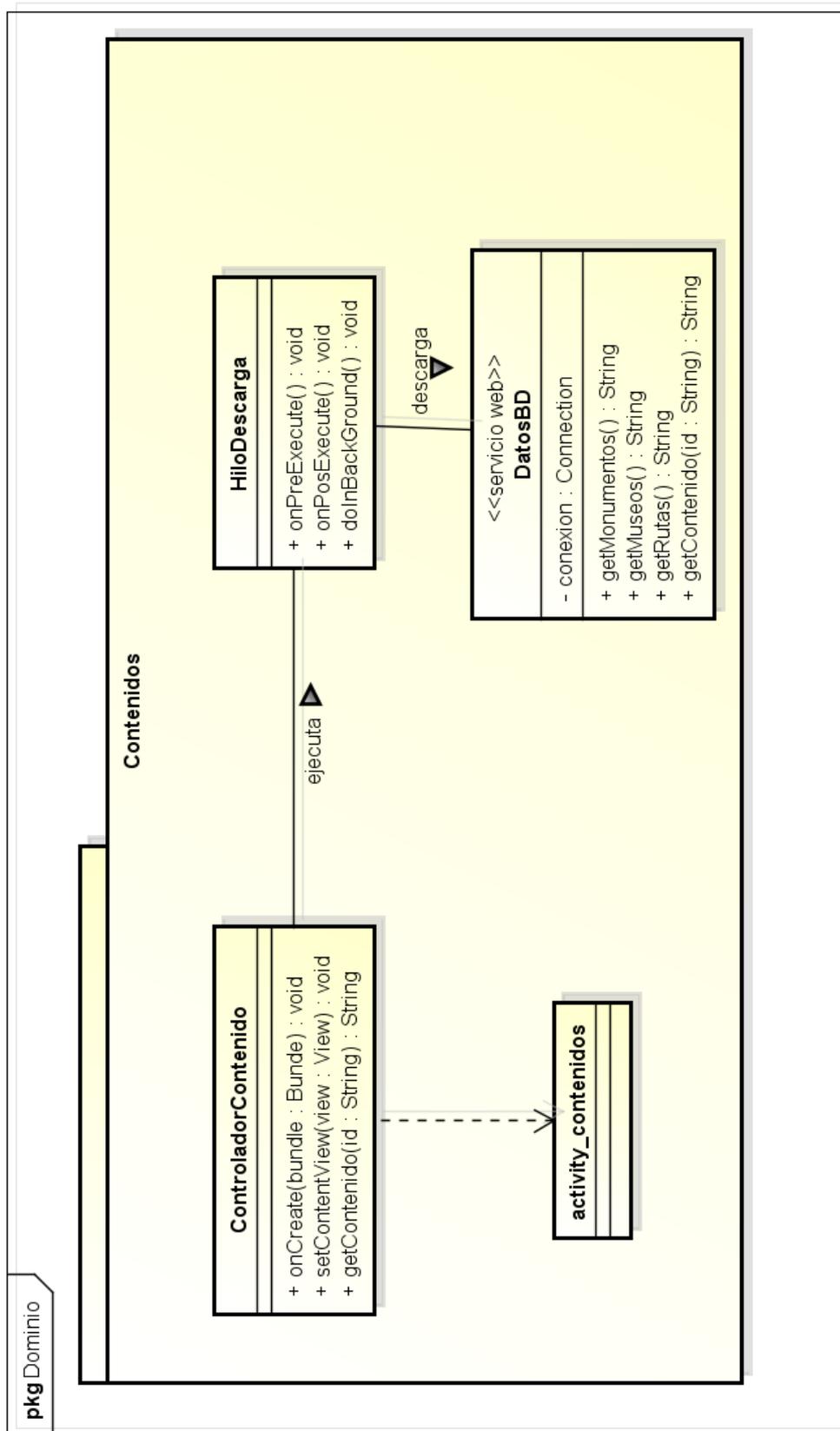


Ilustración 35 Paquete contenidos

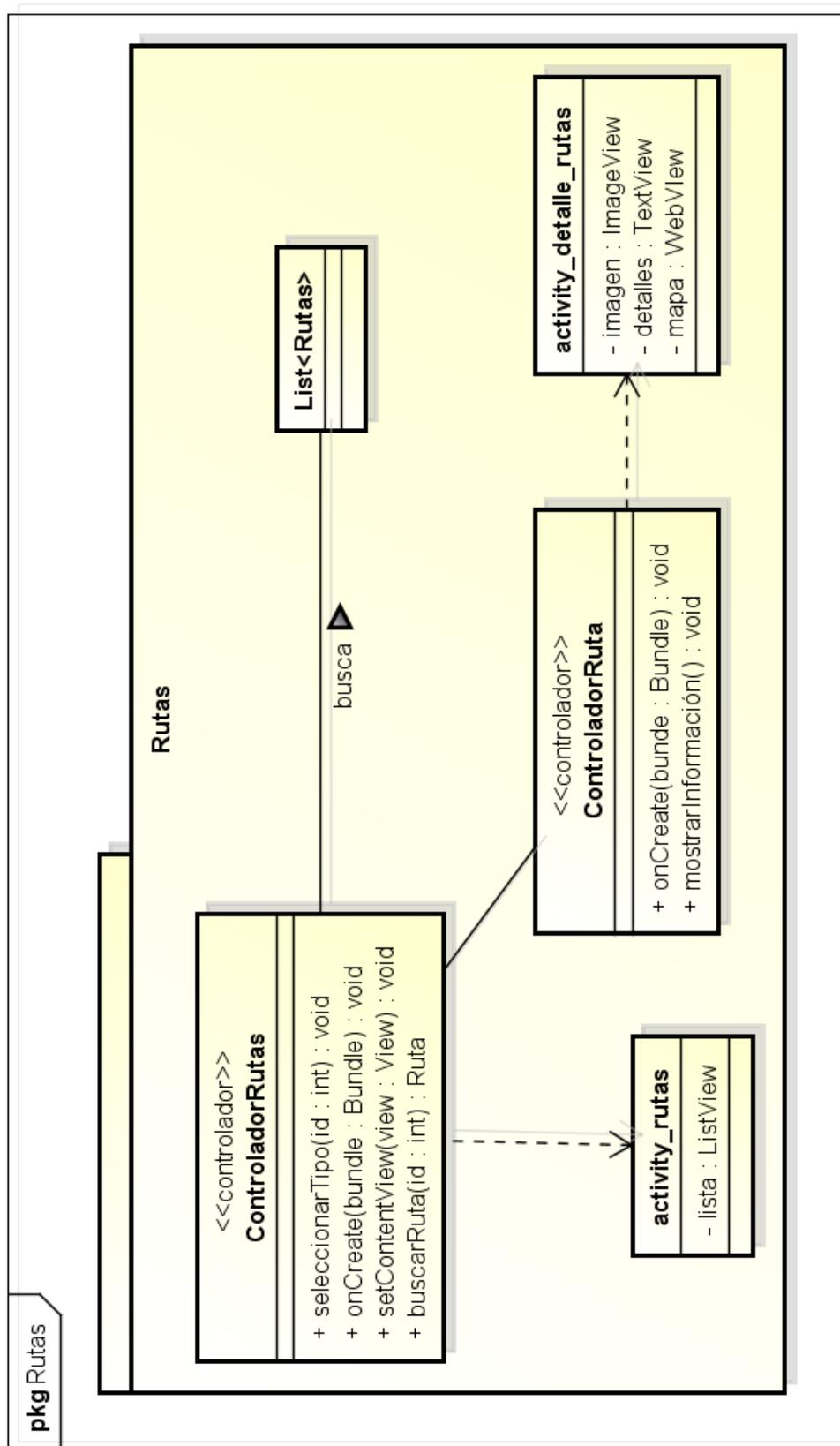


Ilustración 36 Paquete rutas



6. IMPLEMENTACIÓN



6.1. Introducción

En este capítulo se pretende concretar cómo se han llevado a cabo en el desarrollo del proyecto las decisiones de análisis y diseño descritas en los puntos anteriores.

Como se ha especificado en apartados anteriores, los lenguajes utilizados para el desarrollo del sistema son Java, XML y JSON. XML especialmente utilizado para el diseño de interfaz de la aplicación Android.

Además se especificará las diferentes APIs y bibliotecas utilizadas durante el desarrollo de éste que han facilitado esta tarea.

6.2. Librerías y APIS utilizadas

Durante el desarrollo del sistema se han utilizado diversas librerías y frameworks disponibles para facilitar el trabajo al alumno.

6.2.2. Servicio Web

El servicio web se ha implementado en el lenguaje Java. Para ello Java nos ofrece una API denominada JAX-RS, a la que se ha hecho mención en apartados anteriores. A partir de Java 6 no se requiere una configuración para hacer uso de esta API, lo que facilita su utilización.

Por medio de anotaciones en el código nos permite desarrollar servicios REST que accedan a un recurso web.

Para facilitar el desarrollo del servicio web se ha hecho uso del Framework Jersey, que nos proporciona soporte para el uso de las anotaciones. Además de por su facilidad de uso, Jersey ha sido elegido debido a que es nombrado como implementación de referencia por Oracle en su tutorial de Java EE 6.

El servicio web requerido por la aplicaciones solo hace uso de las operaciones GET (*@GET* como anotación). Éstas se encargaran de obtener información de la base de datos y devolverla al usuario en el formato JSON.

Debido al diseño de la base de datos, se ha utilizado también el lenguaje XPath, un lenguaje de procesamiento de XML utilizado para recoger información. Los campos necesarios de la base de datos se encuentran almacenados en formato XML, por lo que para llegar a los datos necesarios para la aplicación móvil se han construido expresiones XPath que se encargan de obtener los datos necesarios.



6.2.3. Aplicación Android

Como se ha comentado antes, para el desarrollo de la aplicación Android se ha utilizado exclusivamente Java, para el procesamiento y XML para el diseño de la interfaz gráfica.

Durante el desarrollo de la aplicación se han utilizado el desarrollo de algunas APIs y librerías que han facilitado el proceso del mismo:

- Google Maps API v2: librería oficial que ofrece Google para la integración y manipulación de sus mapas en aplicaciones Android. Mediante esta librería podemos añadir los mapas de Google a una actividad propia de Android y realizar los cambios oportunos, como añadir o personalizar puntos específicos en el mapa.
- AppCompatActivity: es la librería de compatibilidad de Android. Añadiendo esta librería a la aplicación se consigue compatibilidad de las versiones más antiguas del sistema operativo con la versión 5, actualmente la más reciente de las versiones de Android. De esta manera podemos utilizar nuevos controles añadidos en las últimas versiones, como la barra "Toolbar" o los "CardView" y utilizar la aplicación en dispositivos con versiones anteriores a Android L (API 21).
- GSON (Google Gson): es una librería de código abierto utilizada para serializar y deserializar objetos JSON, es decir convertirlos en objetos Java y viceversa. Esta librería se ha utilizado para convertir con mayor facilidad la información obtenida con el servicio web en formato JSON a objetos Java que la aplicación Android sea capaz de manejar.

6.2.3.1. Imágenes de la aplicación

El uso de imágenes en la aplicación Android cobra gran importancia ya que la hacen más atractiva e intuitiva al usuario. Por otro lado, no es aconsejable sobrecargar la memoria de la aplicación del dispositivo utilizando o almacenando gran cantidad de imágenes y de gran calidad, puesto que ralentiza la aplicación, pudiendo hacer incluso que ésta deje de funcionar en determinados momentos.

Para evitar este problema, nuestra aplicación no almacenará en la base de datos fotos en formato de imagen en ningún momento. Solo se almacenarán pequeñas imágenes en memoria durante la ejecución de la aplicación, como objetos Java.

Por otro lado se debe tener muy en cuenta que la aplicación debe ser fluida y realizar una descarga de datos lo más rápida posible. Por ello durante la sincronización de los datos, las imágenes se transmitirán en Base-64, un sistema de codificación que utiliza únicamente los caracteres ASCII para representar información. De esta forma la descarga de imágenes se realizará de manera más simple y rápida.



Al llegar a la aplicación, ésta se encargará de realizar la conversión de la cadena de caracteres en un objeto imagen Java, Bitmap.

6.3. Soporte en varios idiomas

Como se ha especificado en los requisitos funcionales, la aplicación Android debe tener soporte para varios idiomas. Android proporciona un mecanismo similar al aplicado para el soporte de varios tamaños de pantalla. En su carpeta de recursos (*res*), almacena las cadenas de texto que la aplicación utiliza durante la ejecución de la misma.

Para el caso de las cadenas de texto, éstas se almacenan en el fichero “strings.xml”, dentro de la carpeta “values”. Añadiendo un sufijo a esta carpeta se puede indicar el idioma en el que va a ser usado. Android se encargará automáticamente que obtener las cadenas de texto de la carpeta que le corresponda.

La herramienta utilizada para la implementación de la aplicación, Android Studio, facilita enormemente la tarea de seleccionar el idioma y otros atributos de los recursos de Android a la hora de crear un fichero XML propio de Android.

6.4. Seguridad en Android

La seguridad de un sistema es un aspecto importante en cualquier aplicación actual, y como tal se ha de tener en cuenta a la hora de desarrollar.

Android proporciona su propio esquema de seguridad, fundamentado en tres pilares principales:

- Es un sistema basado en Linux, por lo que aprovechará la seguridad que éste sistema incorpora evitando el acceso directo a hardware o la comunicación con otros recursos del sistema. Las aplicaciones Android se ejecutan de manera aislada en el sistema.
- La aplicación debe ser firmada por un certificado digital, identificando su autor, una clave y los años de validez de ésta. De esta forma se garantiza que el fichero de una aplicación no pueda ser modificado. Además éste certificado se puede enlazar cuando se usan las APIs de Google, como los mapas para evitar que una aplicación con otro certificado pueda hacer uso de éstos.
- Para acceder a determinadas partes del sistema es necesario proporcionar explícitamente estos permisos a la aplicación. Además el usuario de la aplicación podrá conocer que permisos se le han dado a la hora de instalar la aplicación.

Para esta última opción y el caso que nos ocupa se han habilitado los siguientes permisos:

- *android.permission.INTERNET* que permite a la aplicación abrir sockets.
- *android.permission.ACCESS_NETWORK_STATE*: permite el acceso al estado de la red.



- *android.permission.ACCESS_WIFI_STATE*: permite el acceso a información sobre la red wifi.
- *android.permission.WRITE_EXTERNAL_STORAGE*: permite el acceso al almacenamiento externo.
- *android.permission.CHANGE_WIFI_STATE*: permite cambiar el estado del wifi.
- *android.permission.ACCESS_COARSE_LOCATION*: permite el acceso a la ubicación basada en red.
- *android.permission.ACCESS_FINE_LOCATION*: permite el acceso a una ubicación aproximada.
- *android.permission.CALL_PHONE*: permite el acceso a las llamadas del teléfono.
- *android.permission.READ_PHONE_STATE*: permite el acceso a cierta información sobre el teléfono.



7. PRUEBAS



7.1. Introducción

En este capítulo se van a especificar las pruebas realizadas al sistema con el fin de asegurar que cumple todas las especificaciones marcadas en la fase de análisis. Mediante la fase de prueba se pretende conseguir poner de manifiesto los posibles defectos que pueda tener un sistema para poder solucionarlos.

Basándose en los casos de uso de análisis de especificarán los casos de prueba y se detallarán los resultados de éstos.

7.2. Casos de prueba

A continuación se expondrán los casos de prueba junto con el resultado obtenido. La obtención de los casos de prueba estará basada en los casos de uso obtenido durante el análisis del sistema.

Cada caso de prueba ha sido probado en dos dispositivos con características diferentes, ya que la aplicación debe funcionar tanto en móviles como en tablets. Se han utilizado los siguientes dispositivos:

- Samsung Galaxy Trend, con Android 4.0.
- Tablet Samsung Galaxy Tab, 8 pulgadas, con Android 4.4.

A continuación podemos ver las pruebas realizadas durante la primera batería de pruebas:

Identificador	CP_01
Objetivo	Iniciar aplicación
Descripción	El usuario inicia la aplicación.
Entrada	Icono de la aplicación
Salida	El sistema comienza a sincronizar y almacenar los datos actualizados.
Resultado obtenido	El resultado es el esperado

Identificador	CP_02
Objetivo	Iniciar aplicación sin conexión en el dispositivo
Descripción	El usuario inicia la aplicación.
Entrada	Icono de la aplicación.
Salida	El sistema avisa al usuario que no dispone de conexión.
Resultado obtenido	El resultado es el esperado



Identificador	CP_03
Objetivo	Ver el menú de preferencias
Descripción	Se pulsa el botón de preferencias de la aplicación para acceder a éstas.
Entrada	Botón de preferencias.
Salida	Se muestra un menú con las preferencias de la aplicación.
Resultado obtenido	El resultado es el esperado

Identificador	CP_04
Objetivo	Sincronización manual de los datos.
Descripción	Mediante un menú en preferencias se sincronizan de nuevos los datos a petición del usuario.
Entrada	Botón "Sincronizar datos".
Salida	El sistema sincroniza los datos y los almacena en caso de que sean distintos de los existentes.
Resultado obtenido	El resultado es el esperado

Identificador	CP_05
Objetivo	Sincronización manual de los datos sin conexión en el dispositivo.
Descripción	Mediante un menú en preferencias se sincronizan de nuevos los datos a petición del usuario.
Entrada	Botón "Sincronizar datos".
Salida	El sistema avisa al usuario de que no dispone de conexión a internet en el dispositivo
Resultado obtenido	El resultado es el esperado

Identificador	CP_06
Objetivo	Cambiar el idioma de la aplicación
Descripción	Mediante el botón de cambiar idioma en las preferencias el usuario cambia el idioma de español a inglés.
Entrada	Botón "Cambiar idioma"
Salida	El sistema muestra toda la información en inglés.
Resultado obtenido	El resultado es el esperado

Identificador	CP_07
Objetivo	Acceder a la pantalla de búsqueda
Descripción	Mediante la opción de "Búsqueda" en el menú lateral el usuario accede a esta pantalla.
Entrada	Botón "Búsqueda" en menú lateral
Salida	Se muestra la pantalla de búsqueda de la aplicación
Resultado obtenido	El resultado es el esperado



Identificador	CP_08
Objetivo	Realizar una búsqueda sin filtro por tipo
Descripción	En la pantalla de búsqueda realizar una búsqueda sin filtrar por tipo de lugar.
Entrada	Formulario de búsqueda
Salida	Lista de todos los monumentos y museos que coincidan con la palabra buscada
Resultado obtenido	<i>Si existen coincidencias el sistema muestra el listado, en caso contrario no muestra aviso.</i>

Identificador	CP_09
Objetivo	Realizar una búsqueda filtrando por tipo de lugar
Descripción	En la pantalla de búsqueda realizar una búsqueda filtrando por monumentos
Entrada	Formulario de búsqueda
Salida	Lista de todos los monumentos que coincidan con la palabra buscada
Resultado obtenido	<i>Si existen coincidencias el sistema muestra el listado, en caso contrario no muestra aviso.</i>

Identificador	CP_10
Objetivo	Seleccionar un lugar de la lista de resultados obtenidos al buscar
Descripción	Tras la búsqueda de un lugar, se selecciona uno en concreto para obtener más información sobre éste.
Entrada	Lista de resultados de una búsqueda
Salida	Se muestra toda la información relativa al lugar seleccionado.
Resultado obtenido	El resultado es el esperado

Identificador	CP_11
Objetivo	Ver pantalla de "Lugares de interés"
Descripción	Mediante el botón "Lugares de interés" el usuario accede a una pantalla con los tipos de lugares.
Entrada	Botón "Lugares de interés" del menú principal
Salida	Pantalla con los tipos de lugares de interés de la ciudad.
Resultado obtenido	El resultado es el esperado

Identificador	CP_12
Objetivo	Ver la lista de monumentos
Descripción	En el menú de "Lugares de interés", el usuario puede seleccionar Monumentos para ver la lista completa.
Entrada	Opción "Monumentos" del menú de "Lugares de interés"
Salida	Listado de todos los monumentos de la ciudad.
Resultado obtenido	El resultado es el esperado



Identificador	CP_13
Objetivo	Ver los detalles de un monumento
Descripción	Seleccionando un monumento de la lista completa, el usuario puede ver la información detallada de éste.
Entrada	Monumento elegido por el usuario de la lista de monumentos de la ciudad.
Salida	Detalles del monumento seleccionado.
Resultado obtenido	El resultado es el esperado

Identificador	CP_14
Objetivo	Obtener la ruta de un monumento seleccionado
Descripción	Mediante el botón de “Generar ruta” de un monumento seleccionado, el sistema muestra al usuario la ruta desde su posición actual hasta la posición del lugar.
Entrada	Botón “Generar ruta”.
Salida	Mapa con la ruta pintada.
Resultado obtenido	<i>Si el monumento no tiene coordenadas y solo dirección, la aplicación finaliza inesperadamente.</i>

Identificador	CP_15
Objetivo	Acceder a la pantalla de “Mapas”
Descripción	Mediante el botón de “Mapas” del menú principal el usuario accede a esta pantalla.
Entrada	Botón “Mapas”
Salida	Mapa de la ciudad con los diferentes lugares ubicado en él.
Resultado obtenido	El resultado es el esperado

Identificador	CP_16
Objetivo	Seleccionar un lugar del mapa
Descripción	Pulsando en un lugar del mapa se muestra un pop con información de éste.
Entrada	“Marker” del mapa
Salida	Ventana con información del lugar
Resultado obtenido	<i>La información no se adapta correctamente a la ventana.</i>

Identificador	CP_17
Objetivo	Seleccionar un tipo de lugar en el mapa
Descripción	Mediante un desplegable se puede seleccionar los tipos de lugares a mostrar.
Entrada	Desplegable lugares.
Salida	Mapa filtrado
Resultado obtenido	El resultado es el esperado



Identificador	CP_18
Objetivo	Ver los detalles de un lugar ubicado en el mapa
Descripción	Seleccionando un lugar del mapa el usuario accede a información detallada de este.
Entrada	Botón de un lugar ubicado en el mapa
Salida	Ventana con información detallada del lugar.
Resultado obtenido	El resultado es el esperado

Identificador	CP_19
Objetivo	Generar ruta de un lugar del mapa
Descripción	Seleccionando un lugar y el botón generar ruta, el mapa muestra la ruta desde la posición del usuario hasta la ubicación del lugar en el mapa.
Entrada	Botón generar ruta
Salida	Ruta dibujada en el mapa
Resultado obtenido	En el resultado es el esperado

Identificador	CP_20
Objetivo	Acceder a la pantalla de rutas
Descripción	Mediante el botón de "Rutas" del menú principal el usuario accede a esta pantalla.
Entrada	Botón "Ver Rutas"
Salida	Pantalla de rutas de la aplicación
Resultado obtenido	El resultado es el esperado.

Identificador	CP_21
Objetivo	Seleccionar un tipo de ruta
Descripción	En la pantalla de rutas el usuario puede seleccionar los tipos de ruta a visualizar.
Entrada	Menú de tipos de ruta
Salida	Listado de rutas filtrado por tipo.
Resultado obtenido	El resultado es el esperado.

Identificador	CP_22
Objetivo	Ver listado de rutas
Descripción	Seleccionando un tipo de ruta el usuario puede ver el listado de éstas.
Entrada	Botón de tipo de ruta
Salida	Listado de todas las rutas de ese tipo
Resultado obtenido	El resultado es el obtenido



Identificador	CP_23
Objetivo	Ver los detalles de una ruta seleccionada
Descripción	Seleccionando una ruta del listado el usuario puede ver los detalles relativos a ésta.
Entrada	Item de la lista de rutas
Salida	Detalles de la ruta
Resultado obtenido	<i>El mapa no se visualiza correctamente</i>

Identificador	CP_24
Objetivo	Acceder a la pantalla de "Oficinas de Turismo"
Descripción	Mediante el botón "Oficinas de Turismo" del menú lateral el sistema muestra las oficinas de turismo de la ciudad de Valladolid.
Entrada	Botón "Oficinas de Turismo"
Salida	Listado de oficinas
Resultado obtenido	El resultado es el esperado.

Identificador	CP_25
Objetivo	Acceder a la pantalla de "Teléfonos de interés"
Descripción	Mediante el botón "Teléfonos de interés" del menú lateral el usuario puede acceder a esta pantalla.
Entrada	Botón "Teléfonos de interés"
Salida	Listado de teléfonos
Resultado obtenido	El resultado es el esperado

Identificador	CP_26
Objetivo	Realizar la llamada de un teléfono
Descripción	Al pulsar sobre un teléfono, si el dispositivo lo permite, se podrá realizar la llamada
Entrada	Ítem de la lista de teléfonos.
Salida	Llamada
Resultado obtenido	<i>La llamada no se realiza.</i>

Identificador	CP_27
Objetivo	Ver contenido "Ocio nocturno"
Descripción	Mediante el botón "Ocio nocturno" del botón lateral el usuario accede a esta pantalla.
Entrada	Botón "Ocio nocturno"
Salida	Pantalla de ocio nocturno
Resultado obtenido	El resultado es el esperado.



Identificador	CP_28
Objetivo	Ver contenido "Tapas"
Descripción	Mediante el botón "Tapa" del botón lateral el usuario accede a esta pantalla.
Entrada	Botón "Tapas"
Salida	Pantalla de tapas
Resultado obtenido	El resultado es el esperado.

A partir de estas pruebas se corrigieron todos los errores encontrados y se realizó una nueva batería de pruebas con resultados satisfactorios.

7.3. Matriz Casos de uso- Casos de prueba

En la siguiente tabla se muestra la relación entre los casos de uso de análisis y los casos de prueba realizados:

	CU_1	CU_2	CU_3	CU_4	CU_5	CU_6	CU_7	CU_8	CU_9	CU_10	CU_11	CU_12	CU_13	CU_14
CP_01	X									X				
CP_02	X									X				
CP_03		X												
CP_04		X												
CP_05		X												
CP_06		X									X			
CP_07			X											
CP_08			X											
CP_09			X											
CP_10			X											
CP_11						X								
CP_12						X								
CP_13													X	
CP_14														X
CP_15				X										
CP_16				X										
CP_17				X										
CP_18				X										
CP_19				X										
CP_20					X									
CP_21					X									
CP_22					X									
CP_23					X									
CP_24												X		
CP_25								X						
CP_26								X						
CP_27									X					
CP_28										X				



8. CONCLUSIONES Y TRABAJOS FUTUROS



8.1. Conclusiones

En este apartado se detallan las conclusiones y logros obtenidos durante el desarrollo de este proyecto.

En primer lugar, se debe mencionar que el logro principal es el cumplimiento de los requisitos e ideas propuestas al principio del mismo. Muchas de ellas, siempre mejorables, pero que cumplen completamente todo lo establecido.

También se ha logrado mejorar los conocimientos de gestión de un proyecto, de principio a fin, algo que seguro será de utilidad para el futuro y desempeño de la carrera profesional.

Un logro importante que debe considerarse es la gestión y planificación del tiempo necesario para desarrollar el proyecto, ya que se ha tenido muy en cuenta durante todo el desarrollo del mismo.

Por último nombrar los conocimientos obtenidos en el lenguaje Android. Este lenguaje ya conocido por el alumno, se ha visto mejorado notablemente durante la implementación de la aplicación.

8.2. Trabajo futuros

Debido a la gran presencia de las aplicaciones móviles en el mundo de la tecnología en la actualidad y el tipo de aplicación desarrollada, podemos encontrarnos con multitud de mejoras que poder aplicar a este proyecto. A continuación se muestran algunas ideas que podrían ser aplicadas en un futuro.

- Lector de código BIDI en cada monumento: actualmente existe en cada monumento de Valladolid un número de teléfono al que el usuario puede llamar y recibir información acerca de éste. El número de teléfono podría ser sustituido por un código BIDI, más actual, que la aplicación fuera capaz de leer y reproducir la información.
- Añadir otro tipo de información, como hoteles o eventos de turismo.
- Implementación de la aplicación en otro sistema operativo.
- Mejora del rendimiento de la aplicación.





ANEXOS

A: Manual de Usuario

En este apartado se explica cómo utilizar la aplicación, de manera que el usuario pueda hacer uso de todas sus funcionalidades.

Tras haber instalado la aplicación en nuestro dispositivo, ésta se iniciará pulsando sobre el icono que se habrá creado en el menú del dispositivo, que es el siguiente:



Al iniciar la aplicación, se mostrará una pantalla de bienvenida y se comprobará que nuestro dispositivo posee conexión a internet. En caso contrario nos mostrará un aviso como el siguiente:

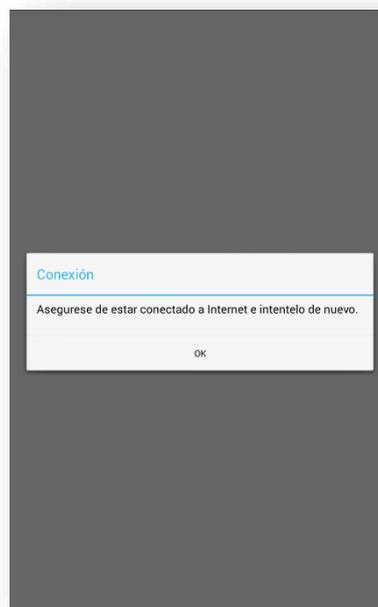
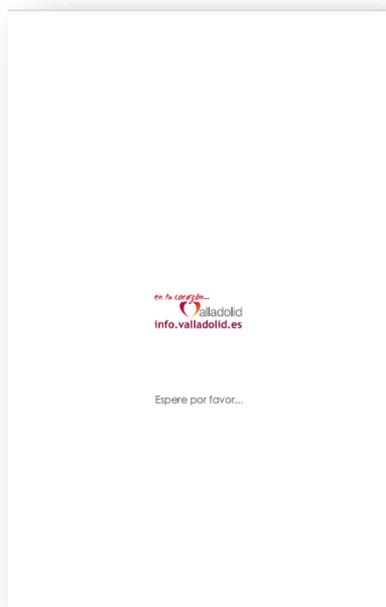


Ilustración 37 Pantalla de bienvenida

La aplicación presenta un menú principal, en el que podemos ver las principales características que ofrece, y un menú secundario que podemos desplegar desde la derecha:

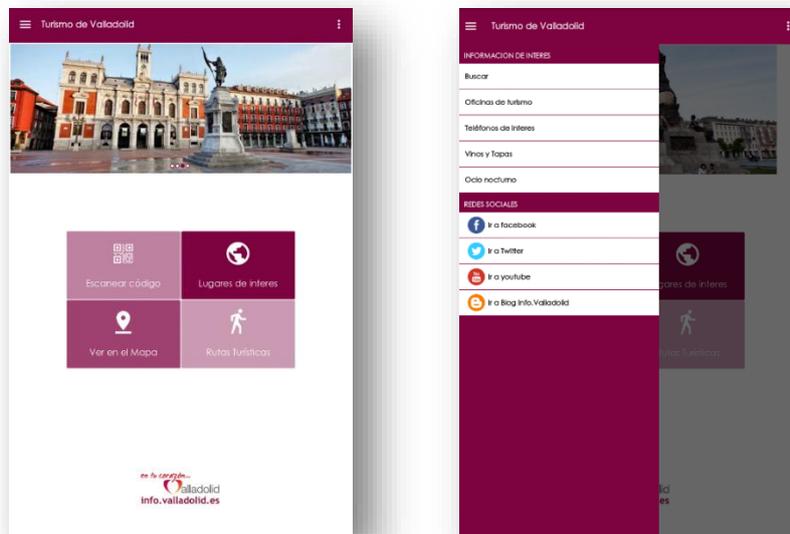


Ilustración 38 Menú de la aplicación

A.1. Lugares de interés y rutas

Puede consultar todos los lugares turísticos de interés de la ciudad de Valladolid filtrados por el tipo de atracción turística. Para ello, en el menú principal puede seleccionar una de las dos opciones siguientes:

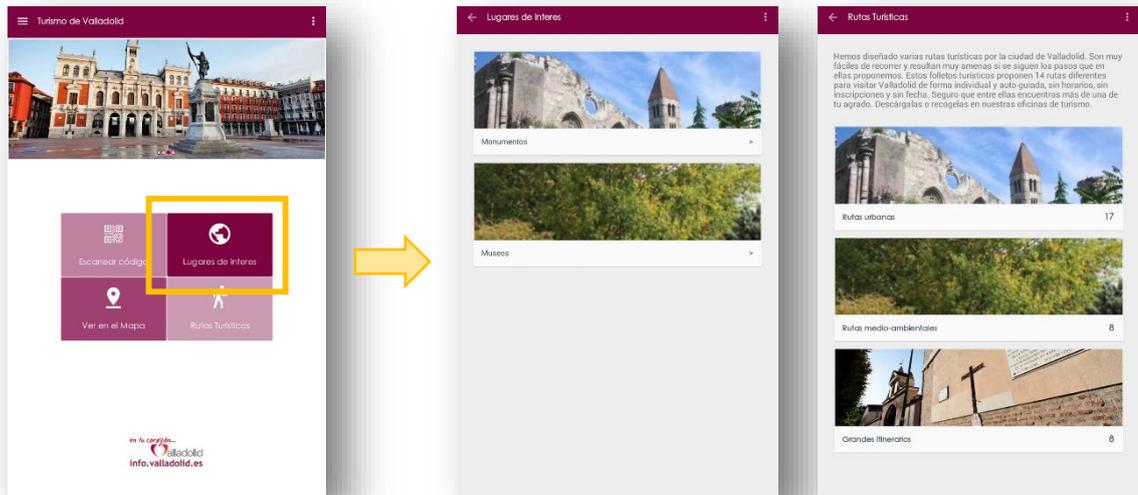


Ilustración 39 Lugares de interés

Si desea visualizar un listado con algún tipo de lugar o ruta, pulse sobre éste, y la aplicación le ofrecerá un listado como el siguiente:



Ilustración 40 Listado de lugares

Si desea obtener más información sobre un lugar en particular, pulse sobre él:



Casa de la India
La Casa de la India en España se constituyó en Valladolid el 17 de marzo de 2003 como fundación de carácter cultural, y con tres patronos: la República de la India, a través de la Embajada de la India en Madrid; el Ayuntamiento de Valladolid; y la Universidad de Valladolid. La sede se ubica en un edificio rehabilitado de principios del siglo XX, que se inauguró el 11 de noviembre de 2006. Es una plataforma innovadora creada para favorecer el diálogo entre la India y España y para promover el desarrollo de sus relaciones en los diferentes ámbitos.



Ilustración 41 Detalle de un lugar

La aplicación deberá mostrar toda la información relativa al sitio que usted haya seleccionado, tal como nombre, descripción, dirección, y una imagen en caso de que ésta exista. Si desea visualizar una ruta desde su posición actual hasta la dirección en la que se encuentra el lugar seleccionado, pulse sobre el botón señalado en la figura anterior, y podrá visualizar un mapa con una ruta dibujada que le servirá de guía, la siguiente figura muestra un ejemplo representativo:

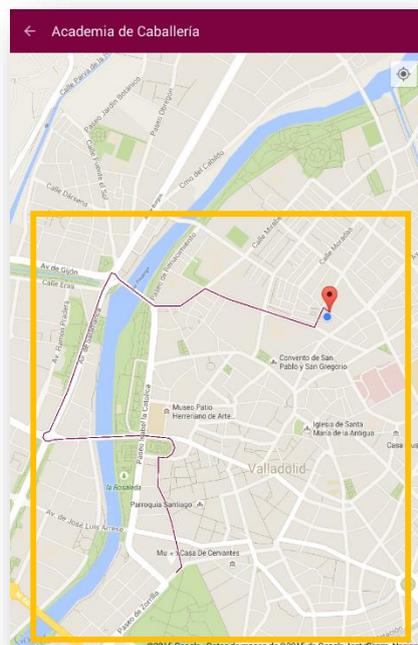


Ilustración 42 Mapas

Si desea realizar una **búsqueda**, acceda a la pantalla principal y deslice el menú lateral. Seleccione la opción de “Búsqueda”, como en la figura siguiente:

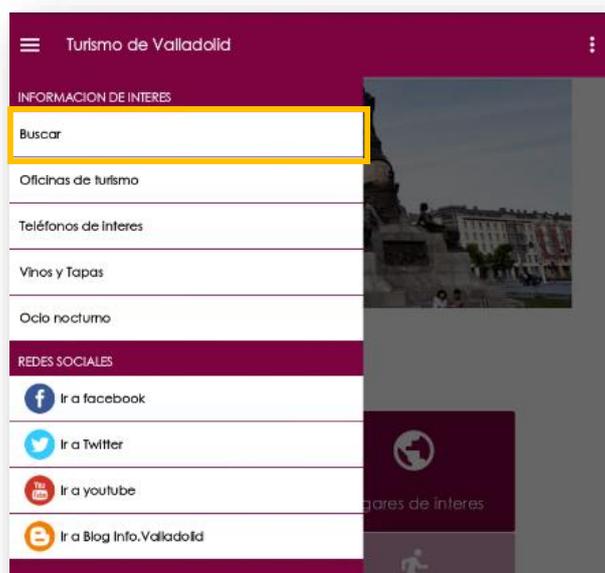


Ilustración 43 Menú de búsqueda

La aplicación le mostrará una pantalla donde poder realizar la búsqueda de lugares turísticos insertando algunos filtros.

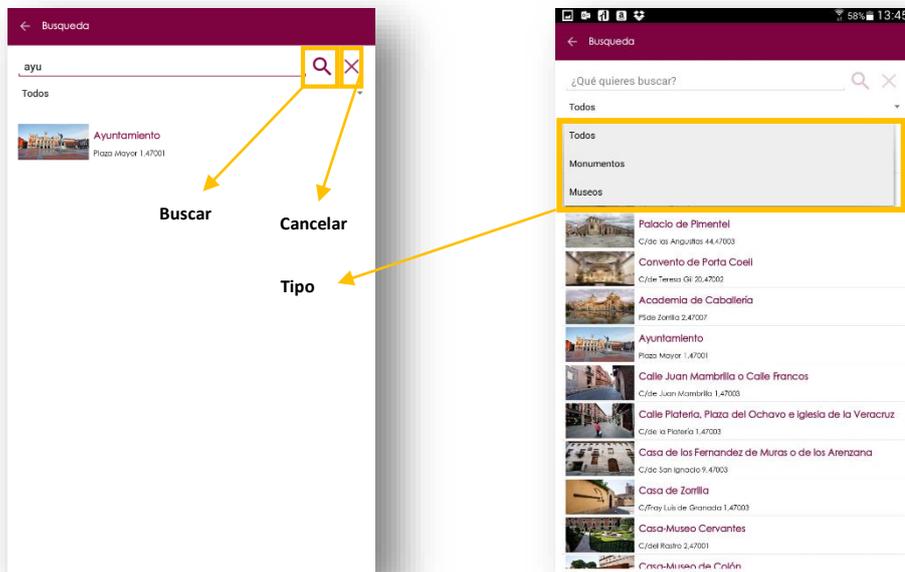


Ilustración 44 Búsqueda de lugares

Si realiza una búsqueda que no produce resultados, la aplicación le mostrará un mensaje de error como el siguiente:



Ilustración 45 Búsqueda sin resultados

A.2. Mapas

Si lo desea también puede visualizar todos los lugares señalados en el mapa de la ciudad. Para esto, diríjase al menú principal y seleccione la opción "Mapas". Podrá visualizar en un mapa todos los lugares de interés turístico de la ciudad, además de poder filtrarlos.

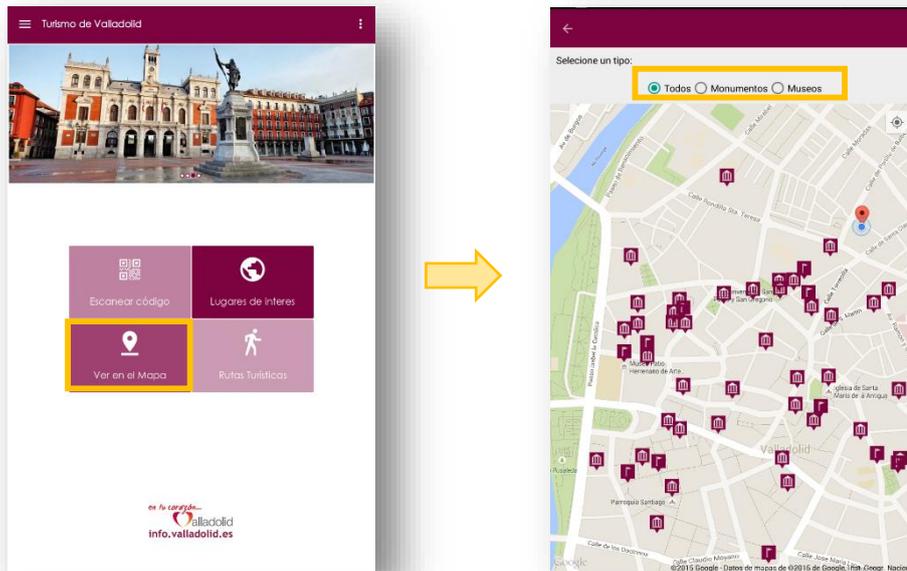


Ilustración 46 Vista del mapa

Además, si desea obtener más información de un lugar situado en el mapa puede pulsar sobre éste para obtener más datos.

A.3. Otra información de interés

También puede seleccionar otra información de interés

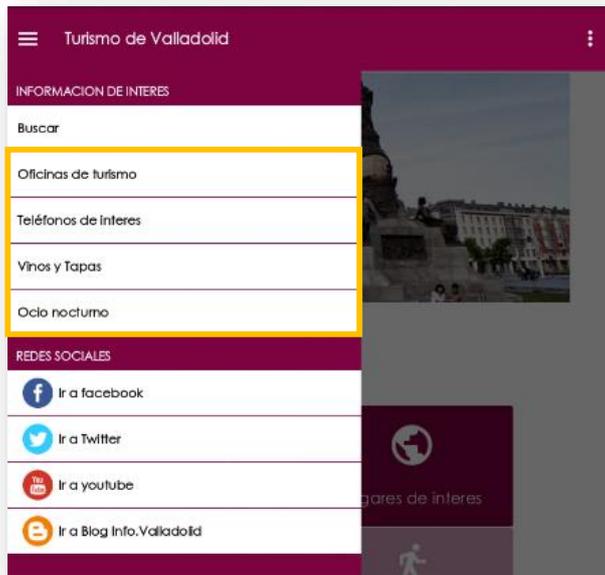


Ilustración 47 Otras opciones, menú lateral

A.4. Redes sociales

La aplicación también ofrece la posibilidad de visitar las redes sociales de Cultura y Turismo de Valladolid, para ello diríjase a la pantalla principal de la aplicación y seleccione un elemento del apartado “Redes Sociales” del menú desplegable derecho:

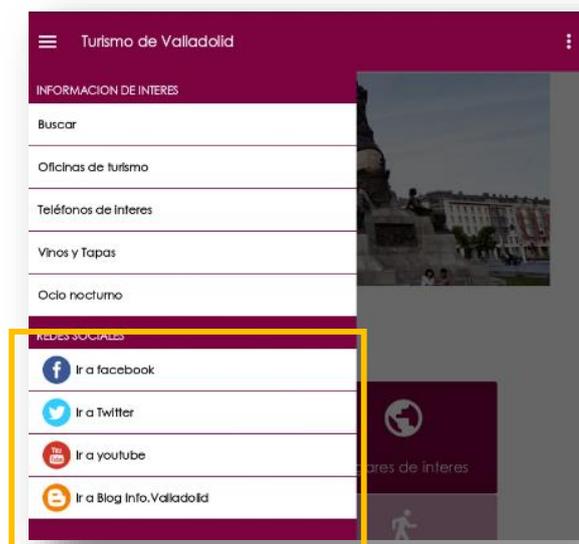


Ilustración 48 Redes sociales

Su dispositivo le avisará que debe abrir el enlace con otra aplicación y le dará la posibilidad de elegir una. Esto es debido a que son enlaces externos a páginas web o redes sociales con sus propias aplicaciones ya desarrollados.

A.5. Preferencias

Accediendo a las preferencias de la aplicación podrá modificar ciertos ajustes. Para acceder a este menú, diríjase al menú principal y pulse sobre el botón “Preferencias” situado en la parte superior derecha de la pantalla:

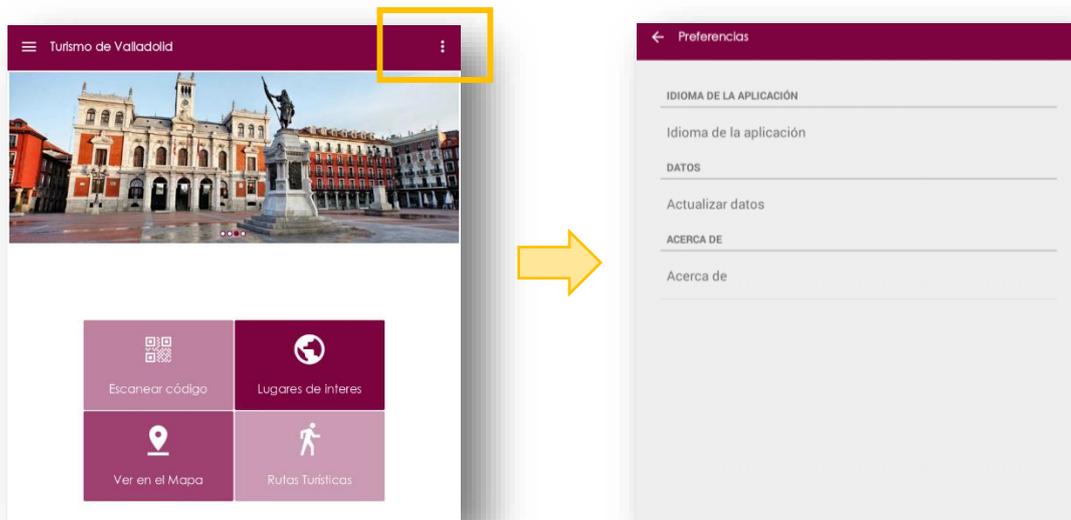


Ilustración 49 Menú de preferencias

Idioma de la aplicación

Puede configurar el idioma de la aplicación en el menú Preferencias > Cambiar Idioma. Se puede seleccionar entre dos idiomas posibles: español e inglés. Al cambiar el idioma toda la información turística y menús de la aplicación se mostrarán en el lenguaje seleccionado.

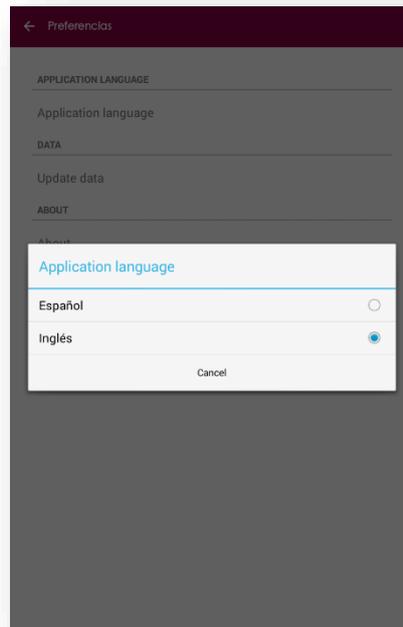


Ilustración 50 Cambio de idioma

Descarga de datos

Si desea volver a sincronizar los datos de la aplicación, también deberá hacerlo desde el menú preferencias de la aplicación. Pulsando en la opción Actualizar datos. La aplicación informará al usuario del proceso de descarga mediante una ventana de notificación como la siguiente:

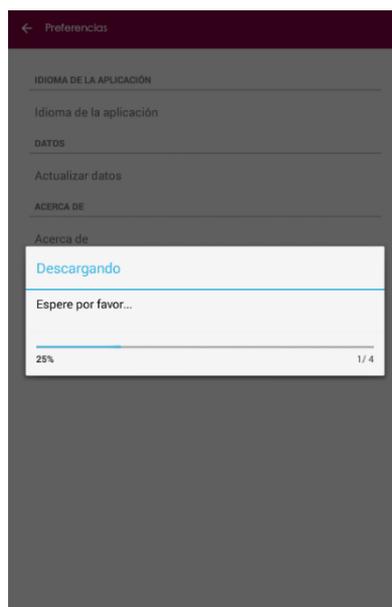


Ilustración 51 Sincronización de datos

Tras finalizar la descarga, la ventana desaparecerá y el usuario podrá hacer un uso normal de la aplicación.

B: Guía de Instalación

Para la instalación de la aplicación en un dispositivo Android, se precisa de algún medio para transmitirle datos desde el ordenador. Se aconseja el uso de un cable USB por su rapidez.

Igualmente será necesario disponer de un navegador de archivos, que viene por defecto instalado en la mayoría de los dispositivos Android del mercado. Mediante el navegador de ficheros, buscaremos nuestra aplicación APK, previamente almacenada en el dispositivo.

La aplicación tendrá el nombre: ValladolidTurismo.apk. Cuando el archivo se ha localizado mediante el navegador de archivos, pulsamos sobre él.

El sistema nos mostrará un diálogo con varias opciones, donde seleccionamos la opción de instalar. Una vez que se acepte, ya quedará instalada en el dispositivo.

Debe asegurarse primero de que en el dispositivo se permite la instalación de aplicaciones externas. Esto se comprueba en el menú de ajustes, dentro de aplicaciones, verificando si aparece marcada la opción “Orígenes desconocidos”.



Ilustración 52 Instalación de fuentes desconocidas



C: Glosario de Términos

En el siguiente glosario se recogen todos los términos y acrónimos utilizados en el presente documento.

API	<i>Application Programming Interface</i> . La Interfaz de Programación de Aplicaciones, es un grupo de rutinas que definen cómo invocar desde un programa, un servicio que éstos prestan. En otras palabras, una API representa una interfaz de comunicación entre componentes software.
APK	<i>Application Package File</i> . Se usa para distribuir e instalar componentes empaquetados para la plataforma Android para smartphones y tablets.
CP	Caso de Prueba
CU	Caso de Uso
DSS	Diagrama de secuencia del sistema.
FRAMEWORK	Estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.
GPS	<i>Sistema de posicionamiento global</i> . Permite determinar la posición de un objeto en el mundo con alta precisión.
HTTP	<i>Hypertext Transfer Protocol</i> , es un protocolo cliente-servidor utilizado en la World Wide Web para el intercambio de información.
IDE	<i>Integrated Development Environment</i> . Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.
Java	Lenguaje de programación orientado a objetos utilizado para el desarrollo del presente proyecto.
JDBC	<i>Java Database Connectivity</i> , es una API que permite realizar operaciones sobre una base de datos, utilizando el lenguaje de programación Java.
JSON	<i>JavaScript Object Notation</i> , formato de texto utilizado para el intercambio de información.
Layout	Esquema de distribución de los elementos en un diseño de una interfaz.
MVC	El <i>Modelo Vista Controlador</i> es un patrón de arquitectura del software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos.
SDK	<i>Software Development Kit</i> . Un kit de desarrollo de software es un conjunto de herramientas de desarrollo que permite a un programador crear aplicaciones para un sistema concreto.
Servicio Web	
URL	<i>UniformResourceLocator</i> . El localizador uniforme de recursos es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación.
UML	<i>UnifiedModelingLanguage</i> , es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo),



	incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.
URL	<i>UniformResourceLocator</i> . El localizador uniforme de recursos es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación.
URI	<i>Uniform Resource Identifier</i> , identificador de recursos en la red, formado por una cadena de caracteres que identificada de forma unívoca los recursos en una red.
XML	<i>eXtensibleMarkupLanguage</i> . El lenguaje de marcas extensible es un metalenguaje extensible de etiquetas, desarrollado por el <i>World Wide Web Consortium</i> y permite definir una gramática de lenguajes específicos. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de forma segura, fiable y fácil.
XPath	<i>XML Path Language</i> , lenguaje que permite recorrer un XML mediante la construcción de expresiones, para obtener la información necesario

D: Código fuente del proyecto

El código fuente del proyecto se encuentra ubicado en un sistema de almacenamiento en la nube en la siguiente dirección URL:

https://dl.dropboxusercontent.com/u/10938789/TFG_RaquelGonzalezSuarez.rar

El fichero se encuentra comprimido en formato “.rar” y será necesario descomprimirlo para acceder al código fuente del proyecto.



Bibliografía

- [1] LARMAN, Craig. *Applying UML and patterns*(ed. Prentice Hall). 2001. ISBN: 0-13-092569-1
- [2] GIRONÉS, Jesús Tomás. *El gran libro de Android, 3ª Edición*. (ed. Marcombo). 2013. ISBN: 978-84-267-1976-8
- [3] ARLOW, Jim. *UML 2*. (ed. Pearson Education). 2009. ISBN: 0-321-32127-8
- [4] SOMMERVILLE, Ian. *Ingeniería del Software*. (ed. Pearson Educación) 2005. ISBN: 84-7829-074-5

Referencias Web

- [1] Wikipedia; Lenguaje Java; 5/Febrero/2015;
[http://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci%C3%B3n\)](http://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)) ;
- [2] INUSUAL; REST vs SOAP al servicio de la Web; 4/Enero/2015;
<http://inusual.com/articulos/rest-vs-soap-al-servicio-de-la-web/>
- [3] Wikipedia; JAX-R; 4/Enero/2015; <http://es.wikipedia.org/wiki/JAX-RS>
- [4] Wikipedia; Servicio Web; 4/Enero/2015; http://es.wikipedia.org/wiki/Servicio_web
- [5] Business Apps; JSON vs SQL; 10/Enero/2015;
<https://appbus.wordpress.com/2013/12/08/json-vs-sql/>
- [6] Androideity; MVC en Android; 10/Enero/2015; <http://androideity.com/2012/05/10/la-importancia-del-mvc-en-android/>
- [7] Asociación Desarrolladores de España; Servicios Web RESTful con HTTP; 10/Enero/2015;
<http://www.adwe.es/general/colaboraciones/servicios-web-restful-con-http-parte-i-introduccion-y-bases-teoricas>
- [8] Androideity; Arquitectura Android; 12/Enero/2015;
<http://androideity.com/2011/07/04/arquitectura-de-android/>
- [9] Software de comunicaciones; Arquitectura Android; 12/Enero/2015;
<https://sites.google.com/site/swcuc3m/home/android/generalidades/2-2-arquitectura-de-android>
- [10] Ranking de Sistemas Operativos más usados; 12/Enero/2015;
<https://blog.uchceu.es/informatica/ranking-de-sistemas-operativos-mas-usados-para-2015/>
- [11] API de Desarrollo de Android; 16/Mayo/2015;
<http://developer.android.com/intl/es/reference/packages.html>
- [12] Google Developers; Google Maps API; 21/Mayo/2015;
<https://developers.google.com/maps/?hl=es>

