

UNIVERSIDAD DE OVIEDO



ESCUELA DE INGENIERÍA INFORMÁTICA - MASTER DE
INGENIERÍA WEB

TRABAJO DE FIN DE MÁSTER

“SISTEMA DE GESTIÓN DE VIAJES SOCIAL COMPUESTO POR
APLICACIÓN MÓVIL ANDROID Y APLICACIÓN WEB”

DIRECTORA: M^a del Puerto Paule Ruiz

CODIRECTOR: Juan Ramón Pérez Pérez

V^o B^o del Director del
Proyecto

AUTOR: Diego Ruiz Álvarez

Resumen

Se ha desarrollado un sistema de gestión de viajes social compuesto por una aplicación móvil Android y una aplicación web, todo este sistema ha sido llamado “Map it!”.

“Map it!” se ha desarrollado para facilitar la experiencia de un viaje a cualquier persona que disponga de un dispositivo móvil con Android. Con este sistema se pretende agilizar y mejorar la forma en la que las personas contamos las experiencias vividas en nuestros viajes.

Cuando varias personas viajan juntas, siempre hay alguien que se encarga de inmortalizar los mejores momentos del viaje, después es el encargado de compartir todas las fotos con los demás lo que puede ser una tarea larga y pesada. “Map it!” pretende solucionar esto.

El sistema completo está formado por una aplicación móvil desde la que se tomaran las fotos, videos y audios del viaje que automáticamente se posicionarán en el mapa en el lugar que hayan sido tomadas, de esta forma nadie tendrá ninguna duda de dónde se ha sacado la foto, el video o la toma de audio.

Una vez que el usuario ha finalizado el viaje podrá exportarlo para que todos sus compañeros y amigos puedan verlo. Al exportar un viaje se crea una página específicamente para éste en el cual se mostrara toda la información recogida por el usuario. Además esta página permite descargar todos los archivos (fotos, videos y audio) en un único archivo zip, lo que hace que no tengas que enviar las fotos al ordenador, comprimir las, y buscar una manera de enviárselas a tus amigos, únicamente compartiendo el enlace del viaje exportado será suficiente. El usuario además puede proteger su viaje mediante una contraseña para que solo pueda ver el contenido del mismo la gente de su elección.

Los mapas generados por “Map it!” no son exclusivamente para viajes, sino que se pueden utilizar para muchos otros fines. Por ejemplo, un crítico de cocina puede utilizar “Map it!” como un blog reflejando en un mapa los distintos restaurantes que ha visitado haciendo de cada foto una entrada en su blog. “Map it!” permite incrustar el mapa generado en cualquier otra web, haciendo que puedas tener tu mapa en donde quieras.

En definitiva “Map it!” es una nueva forma de compartir de una forma más rápida y atractiva las experiencias vividas en un viaje.

Palabras Clave

Mapas, Viajes Geolocalización, Fotos, Aplicaciones móviles Android, Servicios Web, J2EE.

Índice General

CAPÍTULO 1. MEMORIA DEL PROYECTO	15
1.1 RESUMEN DE LA MOTIVACIÓN, OBJETIVOS Y ALCANCE DEL PROYECTO	15
1.2 RESUMEN DE TODOS LOS ASPECTOS	16
CAPÍTULO 2. INTRODUCCIÓN	19
2.1 JUSTIFICACIÓN DEL PROYECTO	19
2.2 OBJETIVOS DEL PROYECTO	20
2.3 ESTUDIO DE LA SITUACIÓN ACTUAL	21
2.3.1 <i>Evaluación de Alternativas</i>	21
CAPÍTULO 3. ASPECTOS TEÓRICOS	26
3.1 ANDROID	26
3.1.1 <i>Arquitectura</i>	27
3.1.2 <i>Características</i>	<i>¡Error! Marcador no definido.</i>
3.2 BASE DE DATOS SQLITE.....	30
3.3 BASE DE DATOS MYSQL	30
3.3.1 <i>Aplicaciones</i>	31
3.3.2 <i>Características</i>	<i>¡Error! Marcador no definido.</i>
3.4 SERVICIOS WEB REST	32
3.5 JAVASCRIPT	33
3.5.1 <i>JavaScript en el lado del servidor</i>	34
3.5.2 <i>Desarrollos posteriores</i>	34
3.6 JQUERY.....	34
3.7 PATRÓN DE ARQUITECTURA MODELO VISTA CONTROLADOR (MVC).....	35
3.7.1 <i>Modelo</i>	35
3.7.2 <i>Vista</i>	35
3.7.3 <i>Controlador</i>	35
3.7.4 <i>Comunicación</i>	35
3.7.5 <i>Unión del modelo con la vista y el controlador</i>	36
3.8 STRUTS2	36
3.9 INYECCIÓN DE DEPENDENCIAS CON SPRING	36
3.9.1 <i>Modulos</i>	37
3.10 MÉTRICA V3	38
CAPÍTULO 4. PLANIFICACIÓN DEL PROYECTO Y RESUMEN DE PRESUPUESTOS	41
4.1 PLANIFICACIÓN.....	41
4.2 RESUMEN DEL PRESUPUESTO	43
CAPÍTULO 5. ANÁLISIS	45
5.1 DEFINICIÓN DEL SISTEMA	45
5.1.1 <i>Determinación del Alcance del Sistema</i>	45
5.2 REQUISITOS DEL SISTEMA	47
5.2.1 <i>Obtención de los Requisitos del Sistema</i>	47
5.2.2 <i>Identificación de Actores del Sistema</i>	49
5.2.3 <i>Especificación de Casos de Uso</i>	49

5.3	IDENTIFICACIÓN DE LOS SUBSISTEMAS EN LA FASE DE ANÁLISIS	56
5.3.1	<i>Descripción de los Subsistemas</i>	56
5.3.2	<i>Descripción de los Interfaces entre Subsistemas</i>	58
5.4	DIAGRAMA DE CLASES PRELIMINAR DEL ANÁLISIS	60
5.4.1	<i>Diagrama de Clases</i>	61
5.5	ANÁLISIS DE CASOS DE USO Y ESCENARIOS	61
5.5.1	<i>Registro de usuario</i>	62
5.5.2	<i>Inicio de sesión en aplicación web</i>	62
5.5.3	<i>Ver mapa exportado en la aplicación web</i>	63
5.5.4	<i>Listar mapas</i>	64
5.5.5	<i>Eliminar mapa</i>	64
5.5.6	<i>Inicio de sesión desde la aplicación móvil</i>	65
5.5.7	<i>Crear nuevo mapa</i>	65
5.5.8	<i>Eliminar mapa (aplicación móvil)</i>	66
5.5.9	<i>Ver mapa de fotos (aplicación móvil)</i>	66
5.5.10	<i>Ver listado de fotos (aplicación móvil)</i>	66
5.5.11	<i>Añadir foto desde la cámara (aplicación móvil)</i>	67
5.5.12	<i>Añadir foto desde la galería de imágenes (aplicación móvil)</i>	68
5.5.13	<i>Eliminar foto del mapa (aplicación móvil)</i>	68
5.5.14	<i>Ver detalle de imagen (aplicación móvil)</i>	69
5.5.15	<i>Ver indicaciones al lugar en el que se sacó una foto (aplicación móvil)</i>	69
5.5.16	<i>Compartir foto (aplicación móvil)</i>	70
5.5.17	<i>Exportar mapa (aplicación móvil)</i>	70
5.5.18	<i>Actualizar mapa (aplicación móvil)</i>	71
5.5.19	<i>Compartir mapa exportado en las redes sociales</i>	71
5.5.20	<i>Comentar mapa exportado</i>	72
5.5.21	<i>Generar código para insertar mapa en otra web</i>	72
5.5.22	<i>Descargar todas las fotos de un mapa</i>	73
5.6	ANÁLISIS DE INTERFACES DE USUARIO	74
5.6.1	<i>Descripción de la Interfaz</i>	74
5.6.2	<i>Descripción del Comportamiento de la Interfaz</i>	83
5.6.3	<i>Diagrama de Navegabilidad</i>	84
5.7	ESPECIFICACIÓN DEL PLAN DE PRUEBAS	86
5.7.1	<i>Pruebas Unitarias</i>	86
5.7.2	<i>Pruebas del sistema</i>	89
5.7.3	<i>Pruebas de usabilidad</i>	90
CAPÍTULO 6.	DISEÑO DEL SISTEMA	91
6.1	ARQUITECTURA DEL SISTEMA	91
6.1.1	<i>Diagramas de Paquetes</i>	91
6.1.2	<i>Diagramas de Despliegue</i>	94
6.2	DISEÑO DE CLASES	97
6.2.1	<i>Diagrama de Clases</i>	97
6.3	DIAGRAMAS DE INTERACCIÓN Y ESTADOS	101
6.3.1	<i>Registro de usuario</i>	101
6.3.2	<i>Inicio de sesión aplicación web</i>	102
6.3.3	<i>Ver mapa exportado en la aplicación web</i>	103
6.3.4	<i>Listar mapas</i>	104
6.3.5	<i>Eliminar Mapa</i>	105
6.3.6	<i>Inicio de sesión desde la aplicación móvil</i>	106

6.3.7	Exportar mapa	107
6.3.8	Actualizar mapa	108
6.4	108	
6.5	DISEÑO DE LA BASE DE DATOS.....	109
6.5.1	Descripción del SGBD Usado	109
6.5.2	Diagrama E-R.....	109
6.6	DISEÑO DE LA INTERFAZ	111
6.6.1	Aplicación móvil.....	111
6.6.2	Aplicación web.....	115
6.7	ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS.....	120
6.7.1	Pruebas Unitarias.....	120
6.7.2	Pruebas de Integración y del Sistema	125
6.7.3	Pruebas de Usabilidad y Accesibilidad	126
6.7.4	Pruebas de Rendimiento	129
CAPÍTULO 7.	IMPLEMENTACIÓN DEL SISTEMA	131
7.1	LENGUAJES DE PROGRAMACIÓN	131
7.1.1	Java	131
7.1.2	JavaScript.....	133
7.2	HERRAMIENTAS Y PROGRAMAS USADOS PARA EL DESARROLLO.....	135
7.2.1	Eclipse	135
7.2.2	Netbeans.....	135
7.2.3	Android Developemnt Tools	136
7.2.4	Photoshop.....	137
7.3	CREACIÓN DEL SISTEMA	139
7.3.1	Problemas Encontrados	139
7.3.2	Descripción Detallada de las Clases	140
CAPÍTULO 8.	DESARROLLO DE LAS PRUEBAS	141
8.1	PRUEBAS UNITARIAS	141
8.2	PRUEBAS DE INTEGRACIÓN Y DEL SISTEMA	146
8.3	PRUEBAS DE USABILIDAD Y ACCESIBILIDAD.....	147
8.3.1	Pruebas de Usabilidad	147
8.3.2	Pruebas de Accesibilidad.....	151
8.4	PRUEBAS DE RENDIMIENTO.....	157
CAPÍTULO 9.	MANUALES DEL SISTEMA	159
9.1	MANUAL DE INSTALACIÓN	159
9.1.1	Instalación de la aplicación móvil.....	159
9.1.2	Instalación de la aplicación web	159
9.2	MANUAL DE USUARIO	163
9.2.1	Aplicación móvil.....	163
9.3	MANUAL DEL PROGRAMADOR.....	172
9.3.1	Configuración de la aplicación web	172
9.3.2	Configuración aplicación móvil.....	173
CAPÍTULO 10.	CONCLUSIONES Y AMPLIACIONES	175
10.1	CONCLUSIONES	175
10.2	AMPLIACIONES.....	175
10.2.1	Introducción de videos.....	175
10.2.2	Viajes gestionados por varios usuario	176

10.2.3	<i>Comentarios en fotos</i>	176
10.2.4	<i>Sistema de valoraciones de mapas y top de mapas</i>	176
10.2.5	<i>Tags y búsqueda</i>	176
10.2.6	<i>Aplicación para IOs</i>	176
CAPÍTULO 11.	PRESUPUESTO	177
11.1	COSTES DE DESARROLLO DEL PROYECTO	177
11.2	COSTES PARA EL CLIENTE	179
11.2.1	<i>Presupuesto final</i>	179
CAPÍTULO 12.	REFERENCIAS BIBLIOGRÁFICAS	181
12.1	REFERENCIAS EN INTERNET	181
CAPÍTULO 13.	APÉNDICES	182
13.1	GLOSARIO Y DICCIONARIO DE DATOS	182
13.2	CONTENIDO ENTREGADO EN EL CD-ROM	183
13.2.1	<i>Contenidos y estructura</i>	183
13.2.2	<i>Código Ejecutable e Instalación</i>	184
13.2.3	<i>Ficheros de Configuración</i>	184
13.3	CÓDIGO FUENTE	185
13.3.1	<i>Aplicación móvil</i>	185
13.3.2	<i>Aplicación web</i>	211

Índice de Figuras

Figura 1.1. Ejemplo	¡Error! Marcador no definido.
Figura 4.1. Ejemplo de diagrama de Gantt	¡Error! Marcador no definido.
Figura 4.2. Ejemplo de cronograma	¡Error! Marcador no definido.
Figura 5.1. Ejemplo de caso de uso 1	¡Error! Marcador no definido.
Figura 5.2. Ejemplo de caso de uso 2	¡Error! Marcador no definido.
Figura 5.3. Diagrama de clases de ejemplo	¡Error! Marcador no definido.
Figura 5.4. Descripción de las actividades de un escenario con un diagrama de robustez (I)	¡Error! Marcador no definido.
Figura 5.5. Descripción de las actividades de un escenario con un diagrama de robustez (II)	¡Error! Marcador no definido.
Figura 5.6. Boceto de una interfaz	¡Error! Marcador no definido.
Figura 6.1. Ejemplo simple de arquitectura del sistema	¡Error! Marcador no definido.

Capítulo 1. Memoria del Proyecto

1.1 Resumen de la Motivación, Objetivos y Alcance del Proyecto

En la época en la que vivimos todo está informatizado, a la gente le gusta que los demás sepan lo que están haciendo, como se divierten, que lugares conocieron... Hay una gran cantidad de dispositivos que permiten capturar estos momentos que después podremos enseñar a los demás.

Esto no ha cambiado en muchos años ya que antes la gente sacaba fotos y las revelaba para luego guardarlas en álbumes archivando estas fotos según su origen, lugar, fecha... La única diferencia es que el proceso que había que llevar a cabo para poder compartir estas fotos era más tedioso. Al igual que el revelar las fotos, comprar los álbumes y archivar las fotos, hoy en día también es una tarea tediosa archivar las fotos (aunque menos que antes). Por esta razón se ha creado Map it!.

El objetivo de Map it es facilitar la compartición de fotos separadas por mapas (álbumes) lo que permite saber donde se ha sacado esta foto sin tener que averiguar el lugar. Map it no necesita cables para archivar las imágenes y además crea una página web para cada mapa desde el que se pueden descargar todas las imágenes, esto hace que la persona que haya sacado las fotos no tenga que descargar las imágenes de la cámara, comprimir todas estas imágenes, buscar la manera de pasársela a sus amigos para que ellos también las tengan...

Map it es un conjunto de servicios que para este proyecto permitirá crear los mapas de imágenes desde una aplicación Android, añadiendo las imágenes tanto de la cámara como de la galería y posicionándolas en el mapa detectando la localización del usuario automáticamente o manualmente en caso de que el usuario no disponga de señal GPS. Cuando el usuario decida que es el momento de compartir su mapa podrá exportar el mismo, esto generará una página web en la que se podrá ver toda la información del mapa y descargar todas las imágenes en un solo archivo zip.

Además el usuario podrá ir actualizando su mapa desde la aplicación pudiendo añadir más fotos o borrar fotos ya añadidas.

La página web generada posee un sistema para comentar con Facebook y un generador de código para incrustar el mapa en la web que se quiera, permitiendo compartir el mapa desde blogs, foros...

En definitiva Map it permite compartir mapas de imágenes y actualizarlos en tiempo real de un modo sencillo, rápido e intuitivo.

1.2 Resumen de Todos los Aspectos

Para la realización de este proyecto se han llevado a cabo distintas tareas por las que todo proyecto ha de pasar. En los siguientes apartados se explicaran las mismas de forma más técnica.

El proyecto comienza buscando información sobre las soluciones que hay en el mercado al problema planteado, compartir fotos geolocalizadas de forma sencilla y rápida. Se evaluaron cada una de estas alternativas viendo de las funcionalidades de las que carecían para añadirlas a este proyecto y observando el buen uso de otras funcionalidades que inspiraron también este proyecto. Una vez que se tuvo toda esta información se fija el alcance del proyecto de forma que se limite el desarrollo del mismo, marcando un punto hasta el que desarrollar el proyecto.

A continuación se realiza el análisis del sistema. Esto permite definir claramente lo que se va a hacer, es decir, lo que el programa ha de realizar. En esta fase se define el sistema, y los diferentes subsistemas, se definen las funcionalidades del proyecto y las pruebas que este ha de pasar una vez que este desarrollado y durante el proceso de desarrollo.

Cuando se ha finalizado el análisis empieza la fase de diseño donde se ha definido como se van a hacer las funcionalidades definidas en el análisis. Se busca la mejor alternativa para que todo el sistema funcione correctamente y de la mejor forma. En esta fase se diseña la base de datos, el modelo de datos y la interfaz del sistema. Lo que permitirá definir las pantallas clave que se han de crear. Además se especifican técnicamente las pruebas anteriormente citadas, de forma que en la fase de pruebas únicamente haya que comprobar que la aplicación funciona como se describe en esta especificación.

Una vez decidido como se iba a hacer el proyecto, que iba a necesitar y como se iban a comunicar los diferentes módulos del sistema se empieza con la implementación del mismo empezando por la aplicación de Android con la que se toman las fotos y se crea el mapa. En paralelo se van desarrollando los módulos necesarios del servidor para recibir la información de los mapas desde los dispositivos móviles. Una vez que se acaba de desarrollar todo el sistema se prueba cogiendo las pruebas diseñadas en el apartado anterior.

Para realizar las pruebas se cuenta con personas de diferentes perfiles ya que no solo se ha de probar que la aplicación funciona correctamente sino que es fácil de usar, e intuitivo ya que en la sociedad actual la mayoría de las personas tienen Smartphone y los utilizan a diario.

Cuando se ha comprobado que todo funciona correctamente se ha puesto visible para hacer una prueba de implantación en un servidor real de forma que pueda ser usada por muchas personas a la vez.

A lo largo de este proyecto también se ha contado con una planificación de forma que se cumplan los plazos de entrega del mismo y se ha desarrollado un presupuesto en función de

las tareas desarrolladas y del tiempo empleado en estas de forma que se puede ver el precio de un sistema de este tipo con aplicaciones clientes (Android) que se comunican con un servidor central.

Capítulo 2. Introducción

2.1 Justificación del Proyecto

A todo el mundo le gusta viajar, cuando se dispone de tiempo libre, de vacaciones, de personas con las que compartir experiencias la gente viaja y comparte lo que les gusta y lo que hacen distinto a su vida cotidiana. Actualmente con la tecnología que se ha desarrollado en los últimos años cualquier persona puede tener acceso a su localización actual, lo que permite añadir mucha funcionalidad a tareas que se llevan realizando desde hace mucho tiempo.

Nos gusta recordar lo que hicimos, con fotos, y donde lo hicimos. Además nos gusta compartirlo con distintas personas, ya sea porque hayan venido de viaje con nosotros, o porque tengan unos gustos parecidos a los nuestros... Este proyecto se ha creado para facilitar esta tarea, la tarea de compartir recuerdos de forma que se tenga la máxima información de estos, tanto visual como una referencia espacial, es decir, el lugar de donde viene este recuerdo.

Hay gran cantidad de alternativas a este proyecto, con diferentes puntos de vista, orientadas a guías de viaje, orientadas a la compartición social. Cada una de estas alternativas tiene sus puntos a favor y en contra pero el planteamiento de las mismas hace que sean totalmente diferentes a Map it. En los siguientes apartados se comentaran las diferencias de Map it con las alternativas que actualmente están en el mercado.

Map it tiene un fin, se ha desarrollado un producto que cubre este fin sin abarcar más de lo necesario, ya que no vamos a reinventar la rueda.

La aplicación se ha desarrollado para Android debido al alto porcentaje de usuarios que utilizan esta plataforma, a que es código libre y por tanto gratis. Además ya tengo experiencia con Android lo que hace el desarrollo más ágil.

El sistema en conjunto se ha desarrollado de una forma muy desacoplada, de forma que la ampliación de este proyecto se realizaría de forma fácil y transparente. Se cuenta con la posibilidad de realizar aplicaciones móviles para crear los mapas en los restantes sistemas operativos móviles que hay en el mercado como IOs o Windows Phone.

2.2 Objetivos del Proyecto

- Facilitar la compartición de fotos geolocalizadas con un único dispositivo. Además se permitirá, al compartir, descargar todos los archivos de un mapa en un único zip.
- Desarrollo de una aplicación Android con un diseño e interacción sencillo e intuitivo que permita a todo tipo de personas sacar fotos y crear un mapa con las mismas de forma automática de forma que la información básica, como la fecha y hora o la localización, se tomen de manera transparente al usuario.
- Crear una aplicación en el servidor que recoja y catalogue los distintos archivos del usuario.
- Crear una página web por cada mapa de forma que esta se pueda compartir. En esta web se podrán descargar todas las fotos del proyecto.
- Integración con las herramientas sociales actuales de forma que la compartición del mapa generado sea posible.
- Desarrollar un proyecto escalable que permita la posibilidad de añadir futuras ampliaciones de forma sencilla y de ampliar los dispositivos móviles compatibles con el sistema para crear los mapas.
- Creación de un widget que permita generar un código para incrustar el mapa creado en la página que se quiera, de forma que se añada otra forma de compartir el mapa.
- Utilización de diversas tecnologías para la práctica y aprendizaje de estas.

2.3 Estudio de la Situación Actual

En el mercado actual existen gran cantidad de aplicaciones móviles para sacar fotos, muchas otras para ver mapas, otras para ver puntos de interés o encontrar lo que queramos a nuestro alrededor. Cada una de ellas cubre una necesidad, pero ninguna de ellas cubre la necesidad de la que trata este proyecto.

Actualmente se supone siempre una conexión a internet, o quizás una conexión por GPS. Esto no es siempre posible ya porque no estemos en nuestro país y no dispongamos de conexión 3g, o porque estemos en una zona aislada donde tampoco haya disponible una conexión. Map it ofrece la posibilidad guardar los archivos de un viaje sin conexión a internet. En el momento que el usuario desee exportar las imágenes, porque tenga una mejor conexión a internet podrá exportarlo y así actualizar el mapa.

Map it no reduce la calidad de las fotos, ya que la finalidad es compartir estas imágenes y que se utilicen como se podrían haber utilizado si se hubieran sacado desde una cámara de fotos normal. Muchas de las aplicaciones actuales reducen la calidad de estas para así poder reducir el tamaño que estas ocupan y poder almacenar más.

Además la mayoría de aplicaciones que permiten compartir imágenes, e información textual no permiten la opción de descargar todos los archivos en un solo paso, Map it permite esta opción para hacer más sencillo el envío de estos archivos a todos sus destinatarios.

La gente además cada vez se preocupa más por la seguridad de sus datos y de sus archivos, Map it permite la creación de mapas privados, dejando así que el usuario decida quien ve y quien no ve el mapa compartido.

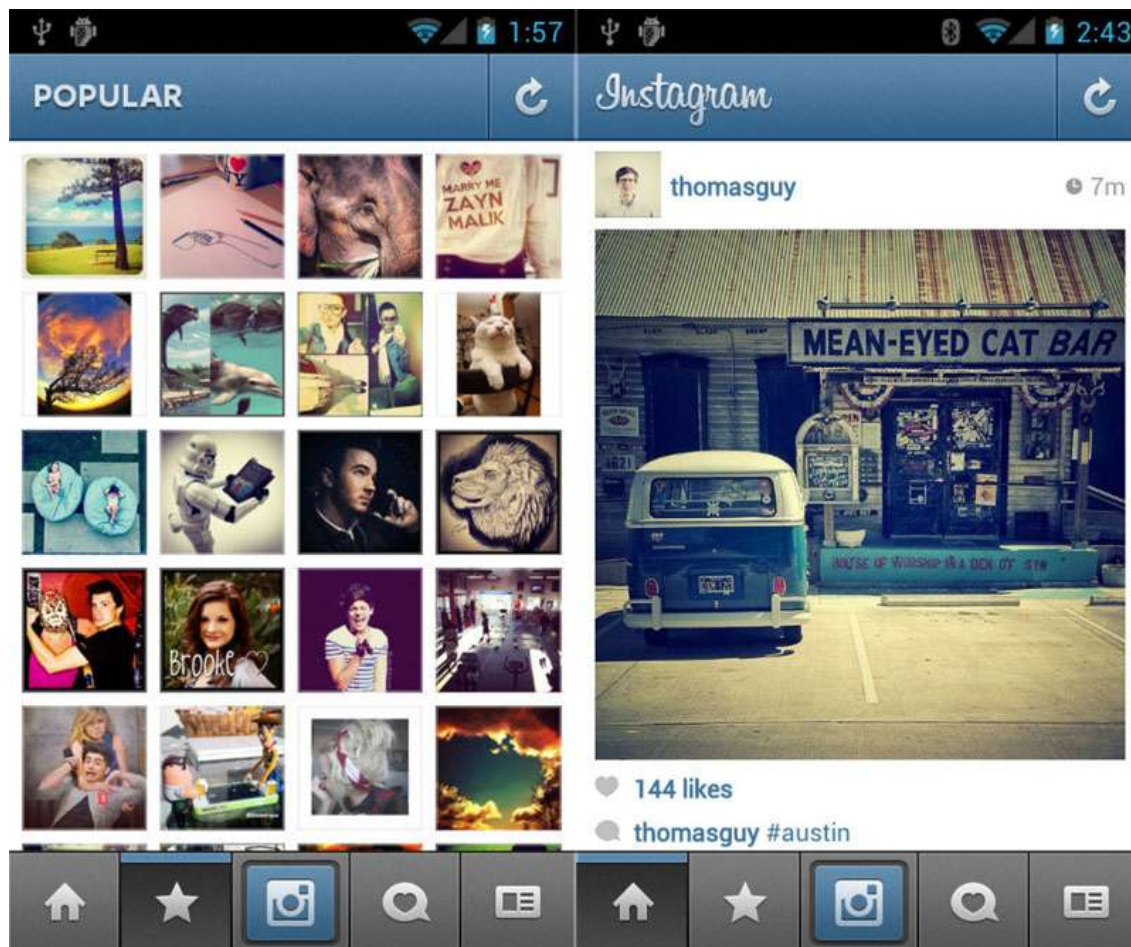
Al contrario que la mayoría de servicios actuales Map it no quiere ser una red social ni mucho menos, ya existen muchas redes sociales y Map it utiliza servicios de estas para la compartición de mapas pero no pretende sustituir ninguna de estas redes sino ser un complemento para ellas.

2.3.1 Evaluación de Alternativas

A continuación se estudian las diferentes alternativas que se encuentran actualmente funcionando en el mercado y que de alguna forma tienen algo en común con Map it.

2.3.1.1 Instagram

Instagram es una red social muy famosa de compartición de imágenes. Esta red social ha sido recientemente comprada por la empresa Facebook debido a su gran número de usuarios y a su repentino éxito.



Instagram en un principio únicamente permitía compartir imágenes, y aplicar filtros a estas imágenes (funcionalidad que hizo tan famosa a esta red social), actualmente también permite compartir videos cortos y localizar las imágenes en un mapa de imágenes.

Al igual que Map it podemos ver las fotos de los diferentes usuarios localizadas en un mapa. Sin embargo no cubre la principal necesidad de este proyecto que es la compartición de estas imágenes en bloque. Instagram genera este mapa pero no permite descargar estos archivos.

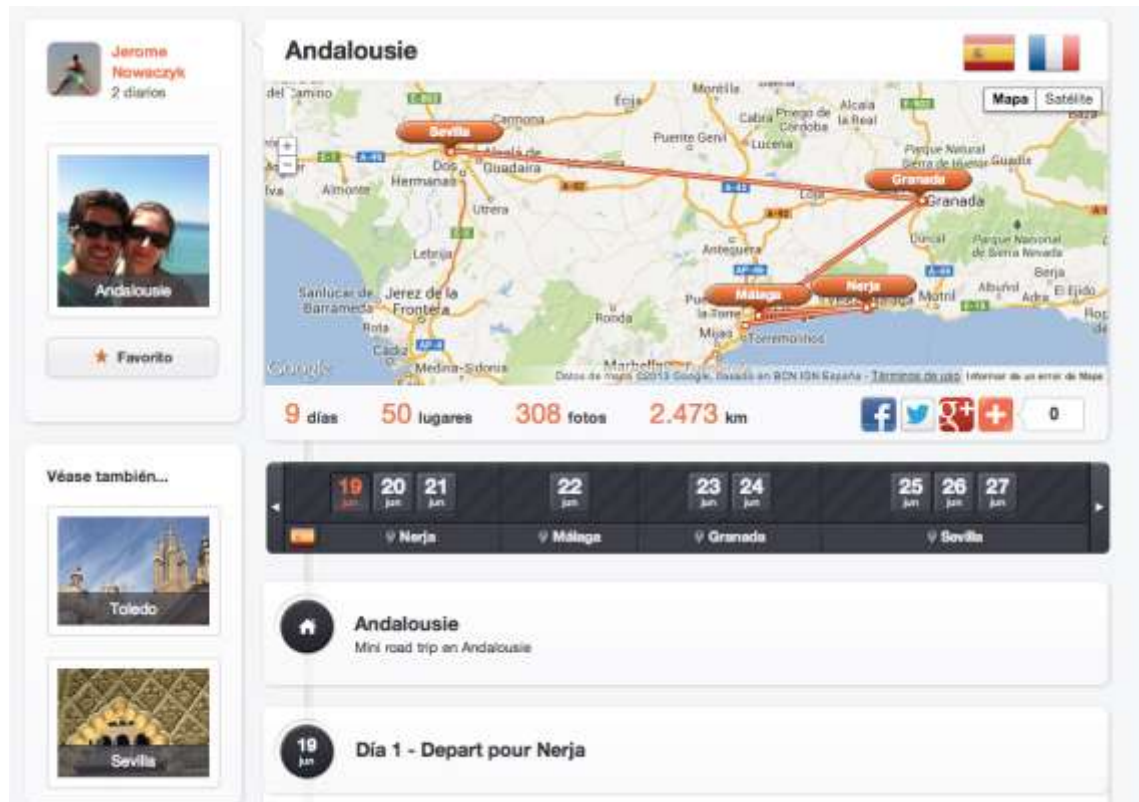
Instagram está muy orientada a ser usada desde teléfonos móviles aunque cuenta con interfaz web desde la que ver los diferentes perfiles. Sin embargo estos perfiles no contienen el mapa de imágenes que se puede ver desde la aplicación móvil, lo que hace que pierda puntos en la compartición de este mapa que es lo que se pretende en este proyecto.

Instagram únicamente permite imágenes cuadradas y de una determinada calidad ya que estas imágenes son tratadas desde la aplicación dándoles un tamaño y una calidad máxima. Esto no permite ser flexible en las imágenes que sacamos y lo que se pretende es que el usuario siga sacando las fotos de forma normal como si de una cámara se tratase sin perder campo de visión porque se tenga que ajustar a un determinado formato de imagen.

Quizás sea porque aun es una red social joven y está en constante evolución pero desde que se creó no para de actualizarse y añadir nuevas funcionalidades lo que me hace pensar que no tardaran en cubrir nuevas necesidades y así ampliar esta red social.

2.3.1.2 Mobylytrip

Este servicio está más orientado a otro fin de Map it, que es compartir experiencias geolocalizadas. Mobylytrip permite registrar todo el recorrido y lugares visitados en un viaje, creando también al igual que Map it una web donde se puede compartir toda esta información.



Como se puede ver en la imagen anterior la aplicación registra los días, lugares, fotos y kilómetros recorridos, además nos crea un blog en el que podemos redactar nuestras experiencias.

Esta aplicación es muy similar a map it pero sigue sin contar con la funcionalidad principal que permite descargar todas las fotos de una sola vez. De forma que la gente que haya viajado contigo pueda tener las mismas fotos que tú has sacado.

La página generada por Map it también tiene forma de blog, pero como se ha comentado antes no pretende ser una red social mientras que Mobylytrip tiene este tipo de formato, con seguidores, perfiles...

Mobylytrip está más orientado a crear guías de viajes que a compartir los archivos de los mismos, cuenta con una gran cantidad de guías de viajes que marcan puntos de interés en

cada ciudad, y estas guías son creadas por los propios usuarios, de forma que se puede elegir entre todas las guías la que más se adecue a nuestro viaje.

2.3.1.3 Tripshare

Tripshare al igual que el resto de alternativas tiene una gran cantidad de puntos en común con Map it. Esta aplicación está enfocada a los viajes también, pero con un fin distinto que el de compartir estos viajes con los demás, su fin es planear los viajes.

Con tripshare se descubren destino, vuelos, hoteles, restaurantes. Se crean itinerarios con cada uno de estos elementos y permite compartir esta información con las personas que van a viajar con nosotros. Lo que facilita la planificación del viaje entre varias personas que pueden buscar alternativas y valorar las distintas opciones en un mismo lugar. Después de planear tu viaje Tripshare te permite reservar lo que necesites desde el propio sistema (hoteles, vuelos...)

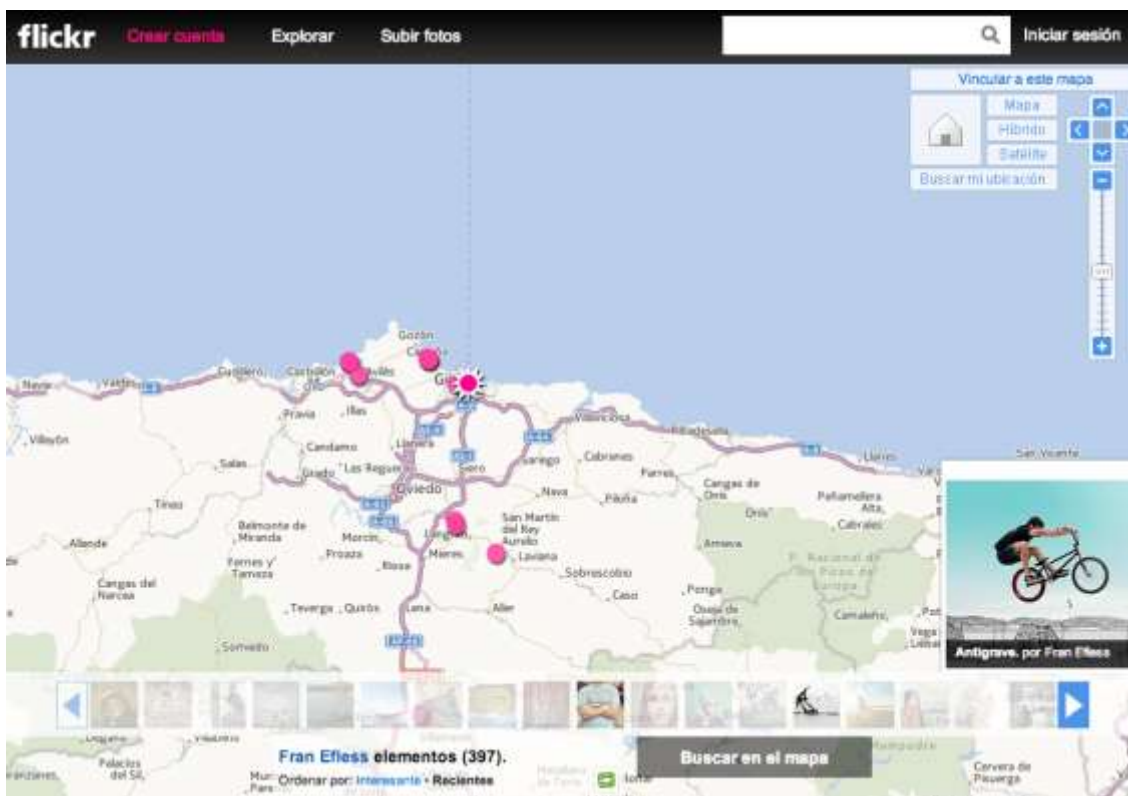


Este sistema aunque tiene muchos puntos en común con Map it no intenta cubrir las mismas necesidades, sino que pretende compartir pasos previos a un viaje al contrario que Map it que pretende compartir el resultado de un viaje.

2.3.1.4 Flickr

Flickr es un servicio muy conocido de compartición de fotos. En este sistema los usuarios crean su perfil y suben las fotos que sacan. Flickr está más orientado a fotógrafos y personas que se preocupan por la calidad de las imágenes, ya que permite subir imágenes de distintas resoluciones permitiendo imágenes en HD.

Recientemente Flickr ha añadido la opción de geolocalizar las imágenes y crear un mapa con las mismas.



Como se ha dicho anteriormente flickr no quiere cubrir las mismas necesidades, y no permite la descarga de todos los archivos de un mapa, que es lo que facilita la compartición del mismo. Flickr cuenta con una gran comunidad de usuarios y un gran diseño y optimización para ver con mucha calidad las imágenes de que cada usuario ofrece en sus perfiles pero no ofrece la posibilidad de contar una historia, de contar las experiencias vividas con un determinado hilo que la gente pueda seguir.

Capítulo 3. Aspectos Teóricos

3.1 Android

Android es un sistema operativo basado en Linux para dispositivos móviles, como teléfonos móviles o tablets. Fue desarrollado inicialmente por Android Inc, una firma comprada por Google en 2005. Es el principal producto de la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio.

Android es un sistema operativo desarrollado de forma abierta al contrario de otros sistemas operativos para dispositivos móviles por lo tanto es posible acceder al código fuente e incluso reportar soluciones a problemas aun no solucionados.

La tienda de aplicaciones Android conocida como "Google play" retribuye a los desarrolladores el 70% del precio de su aplicación. Asimismo el desarrollo de aplicaciones para Android no requiere aprender lenguajes complejos de programación. Todo lo necesario es un conocimiento aceptable de Java y estar en posesión del kit de desarrollo de software o SDK provisto por Google el cual se puede descargar gratuitamente.

Android representa en la actualidad uno de los principales sistemas operativos móviles del mercado creciendo cada vez más los usuarios que lo utilizan y las empresas que desarrollan dispositivos con este sistema operativo.

3.1.1 Arquitectura

Los principales componentes de Android son:

- **Aplicaciones:** las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.
- **Marco de trabajo de aplicaciones:** los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.
- **Bibliotecas:** Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.
- **Runtime de Android:** Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecuta archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato .dex por la herramienta incluida "dx".
- **Núcleo Linux:** Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software

3.1.2 Características

Diseño de dispositivo	La plataforma es adaptable a pantallas más grandes, VGA, biblioteca de gráficos 2D, biblioteca de gráficos 3D basada en las especificaciones de la OpenGL ES 2.0 y diseño de teléfonos tradicionales.
Almacenamiento	<u>SQLite</u> , una base de datos liviana, la cual es usada para propósitos de almacenamiento de datos.
Conectividad	Android soporta las siguientes tecnologías de conectividad: <u>GSM/EDGE</u> , <u>IDEN</u> , <u>CDMA</u> , <u>EV-DO</u> , <u>UMTS</u> , <u>Bluetooth</u> , <u>Wi-Fi</u> , <u>LTE</u> y <u>WiMAX</u> .
Mensajería	<u>SMS</u> y <u>MMS</u> son formas de mensajería, incluyendo mensajería de texto y ahora la Android Cloud to Device Messaging Framework (<u>C2DM</u>) es parte del servicio de Push Messaging de Android.
Navegador web	El navegador web incluido en Android está basado en el motor de renderizado de código abierto <u>WebKit</u> , emparejado con el motor JavaScript V8 de Google Chrome.
Soporte de Java	Aunque las aplicaciones son escritas en Java, no hay una Máquina Virtual de Java en la plataforma. El código Java no es ejecutado. El código Java se compila en el ejecutable Dalvik y corre en la Máquina Virtual Dalvik. Dalvik es una máquina virtual especializada diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados. El soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME MIDP Runner. ⁶⁴
Soporte multimedia	Android soporta los siguientes formatos multimedia: <u>WebM</u> , <u>H.263</u> , <u>H.264</u> (en <u>3GP</u> o <u>MP4</u>), <u>MPEG-4 SP</u> , <u>AMR</u> , <u>AMR-WB</u> (en un contenedor 3GP), <u>AAC</u> , <u>HE-AAC</u> (en contenedores MP4 o 3GP), <u>MP3</u> , <u>MIDI</u> , <u>Ogg Vorbis</u> , <u>WAV</u> , <u>JPEG</u> , <u>PNG</u> , <u>GIF</u> y <u>BMP</u> .
Soporte para streaming	Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video> tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player. Se planea el soporte de Microsoft Smooth Streaming con el port de Silverlight a Android. Adobe Flash HTTP Dynamic Streaming estará disponible mediante una actualización de Adobe Flash Player.
Soporte para hardware adicional	Android soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, termómetro, aceleración 2d y 3d.
Entorno de	Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del

desarrollo	rendimiento del software. El entorno de desarrollo integrado es Eclipse (actualmente 3.4 o 3.5) usando el plugin de Herramientas de Desarrollo de Android.
Market	El Google Play es un catálogo de aplicaciones que pueden ser descargadas e instaladas en dispositivos Android sin la necesidad de un PC.
Multi-táctil	Android tiene soporte nativo para pantallas multi-táctiles que inicialmente hicieron su aparición en dispositivos como el HTC Hero. La funcionalidad fue originalmente desactivada a nivel de kernel (posiblemente para evitar infringir una patente de Apple relacionada con tecnología de pantallas táctiles). Más tarde, Google publicó una actualización para el <u>Nexus One</u> y el <u>Motorola Droid</u> que activa el soporte para pantallas multi-táctiles de forma nativa.
Bluetooth	El soporte para A2DP y AVRCP fue agregado en la versión 1.5; el envío de archivos (OPP) y la exploración del directorio telefónico fueron agregados en la versión 2.0; y el marcado por voz junto con el envío de contactos entre teléfonos lo fueron en la versión 2.2.
Videollamada	La versión principal de Android no soporta videollamada, Sin embargo algunos dispositivos podrían tener una versión personalizada del sistema operativo que lo soporta, ya sea por la red del operador (como el Samsung i9000 Galaxy S) o sobre IP.
Multitarea	Multitarea real de aplicaciones está disponible
Características basadas en voz	La búsqueda en Google a través de voz está disponible como "Entrada de Búsqueda" desde la versión inicial del sistema.
Tethering	Android soporta tethering, el cual permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico (Todos los teléfonos con la versión 2.2., no oficial en teléfonos con versión 1.6 o superiores mediante aplicaciones disponibles en el Android Market, por ejemplo PdaNet). Para permitir a un computador portátil usar la conexión 3G del móvil Android se podría requerir la instalación de software adicional.

3.1.3

3.2 Base de datos SQLite

Las bases de datos SQLite han sido utilizadas en la aplicación Android para almacenar la información del viaje del usuario que luego va a exportar.

SQLite es un sistema de gestión de bases de datos relacional compatible con ACID (acrónimo de Atomicidad, Consistencia, Aislamiento y Durabilidad en español), contenida en una relativamente pequeña biblioteca en C. SQLite es un proyecto de dominio público creado por D. Richard Hipp.

Son ya muchos los productos que utilizan SQLite. El claro ejemplo de esto es que Android usa SQLite para almacenamiento de datos estructurados. Mozilla Firefox lo utiliza, por ejemplo, para almacenar las cookies, el historial y los favoritos. El nuevo sistema de almacenamiento implementado en el núcleo de Mozilla (MozStorage) utiliza SQLite. Mozilla Firefox 3 usa un sistema basado en SQLite para almacenar bookmarks y el histórico de navegación. Se está considerando también utilizar SQLite en Mozilla Calendar y Mozilla Thunderbird.

Una base de datos SQLite puede ser accedida desde gran cantidad de lenguajes de programación, como por ejemplo: Java, C/C++, REALbasic, Pythom, Visual Basic, Delphi, PHP, Pascal. También existe un programa independiente de nombre “sqlite” que puede ser utilizado para consultar y gestionar los ficheros de base de datos SQLite.

3.3 Base de datos MYSQL

Las bases de datos MYSQL han sido utilizadas en el servidor, para almacenar los datos exportados del usuario.

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.¹ MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones

contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson y Michael Widenius.

3.3.1 Aplicaciones

MySQL es muy utilizado en aplicaciones web, como Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación.

3.3.2 Características

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad.

Poco a poco los elementos de los que carecía MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferente velocidad de operación, soporte físico, capacidad, distribución geográfica, transacciones...
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

MySQL es un sistema de administración de bases de datos. Una base de datos es una colección estructurada de tablas que contienen datos. Esta puede ser desde una simple lista de compras a una galería de pinturas o el vasto volumen de información en una red corporativa. Para agregar, acceder a y procesar datos guardados en un computador, usted necesita un administrador como MySQL Server. Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir qué puede hacer y qué no puede hacer con el software en diferentes situaciones. Si usted no se ajusta al GPL o requiere introducir código MySQL en aplicaciones comerciales, usted puede comprar una versión comercial licenciada.

3.4 Servicios web REST

Los servicios web REST se han utilizado para la comunicación entre la aplicación móvil Android y el servidor.

La Transferencia de Estado Representacional (Representational State Transfer) o REST es una técnica de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web. El término se originó en el año 2000, en una tesis doctoral sobre la web escrita por Roy Fielding, uno de los principales autores de la especificación del protocolo HTTP y ha pasado a ser ampliamente utilizado por la comunidad de desarrollo.

Si bien el término REST se refería originalmente a un conjunto de principios de arquitectura en la actualidad se usa en el sentido más amplio para describir cualquier interfaz web simple que utiliza XML y HTTP, sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes como el protocolo de servicios web SOAP. Es posible diseñar sistemas de servicios web de acuerdo con el estilo arquitectural REST de Fielding y también es posible diseñar interfaces XMLHTTP de acuerdo con el estilo de llamada a procedimiento remoto pero sin usar SOAP. Estos dos usos diferentes del término REST causan cierta confusión en las discusiones técnicas, aunque RPC no es un ejemplo de REST.

Los sistemas que siguen los principios REST se llaman con frecuencia RESTful; los defensores más acérrimos de REST se llaman a sí mismos RESTafaris.

REST afirma que la web ha disfrutado de escalabilidad como resultado de una serie de diseños fundamentales clave:

- Un protocolo cliente/servidor sin estado: cada mensaje HTTP contiene toda la información necesaria para comprender la petición. Como resultado, ni el cliente ni el servidor necesitan recordar ningún estado de las comunicaciones entre mensajes. Sin embargo, en la práctica, muchas aplicaciones basadas en HTTP utilizan cookies y otros mecanismos para mantener el estado de la sesión (algunas de estas prácticas, como la reescritura de URLs, no son permitidas por REST)
- Un conjunto de operaciones bien definidas que se aplican a todos los recursos de información: HTTP en sí define un conjunto pequeño de operaciones, las más

importantes son POST, GET, PUT y DELETE. Con frecuencia estas operaciones se equiparan a las operaciones CRUD que se requieren para la persistencia de datos, aunque POST no encaja exactamente en este esquema.

- Una sintaxis universal para identificar los recursos. En un sistema REST, cada recurso es direccionable únicamente a través de su URI.
- El uso de hipermedios, tanto para la información de la aplicación como para las transiciones de estado de la aplicación: la representación de este estado en un sistema REST son típicamente HTML o XML. Como resultado de esto, es posible navegar de un recurso REST a muchos otros, simplemente siguiendo enlaces sin requerir el uso de registros u otra infraestructura adicional.

3.5 JavaScript

JavaScript ha sido utilizado en la aplicación web para la interacción del mapa y para los componentes dinámicos que existen en la página.

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,³ basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas⁴ aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Una cuarta edición está en desarrollo e incluirá nuevas características tales como paquetes, espacio de nombres y definición explícita de clases.

JavaScript fue desarrollado originalmente por Brendan Eich de Netscape con el nombre de Mocha, el cual fue renombrado posteriormente a LiveScript, para finalmente quedar como JavaScript. El cambio de nombre coincidió aproximadamente con el momento en que Netscape agregó soporte para la tecnología Java en su navegador web Netscape Navigator en la versión 2.002 en diciembre de 1995. La denominación produjo confusión, dando la

impresión de que el lenguaje es una prolongación de Java, y se ha caracterizado por muchos como una estrategia de mercadotecnia de Netscape para obtener prestigio e innovar en lo que eran los nuevos lenguajes de programación web.

3.5.1 JavaScript en el lado del servidor

Netscape introdujo una implementación de script del lado del servidor con Netscape Enterprise Server, lanzada en diciembre de 1994 (poco después del lanzamiento de JavaScript para navegadores web).^{9 10} A partir de mediados de la década de los 2000, ha habido una proliferación de implementaciones de JavaScript para el lado servidor. Node.js es uno de los notables ejemplos de JavaScript en el lado del servidor, siendo usado en proyectos importantes

3.5.2 Desarrollos posteriores

JavaScript se ha convertido en uno de los lenguajes de programación más populares en internet. Al principio, sin embargo, muchos desarrolladores renegaban el lenguaje porque el público al que va dirigido lo formaban publicadores de artículos y demás aficionados, entre otras razones.¹³ La llegada de Ajax devolvió JavaScript a la fama y atrajo la atención de muchos otros programadores. Como resultado de esto hubo una proliferación de un conjunto de frameworks y librerías de ámbito general, mejorando las prácticas de programación con JavaScript, y aumentado el uso de JavaScript fuera de los navegadores web, como se ha visto con la proliferación de entornos JavaScript del lado del servidor. En enero de 2009, el proyecto CommonJS fue inaugurado con el objetivo de especificar una librería para uso de tareas comunes principalmente para el desarrollo fuera del navegador web.

3.6 jQuery

Para facilitar el uso de Javascript se ha utilizado la librería jQuery.

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones (FLV) y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC. jQuery es la biblioteca de JavaScript más utilizado.¹

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos.² jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

Las empresas Microsoft y Nokia anunciaron que incluirán la biblioteca en sus plataformas.³ Microsoft la añadirá en su IDE Visual Studio⁴ y la usará junto con los frameworks ASP.NET AJAX y ASP.NET MVC, mientras que Nokia los integrará con su plataforma Web Run-Time.

3.7 Patrón de arquitectura Modelo Vista Controlador (MVC)

El patrón de arquitectura MVC (Modelo Vista Controlador) es un patrón que define la organización independiente del Modelo (Objetos de Negocio), la Vista (interfaz con el usuario u otro sistema) y el Controlador (controlador del workflow de la aplicación).

De esta forma, dividimos el sistema en tres capas donde, como explicaremos más adelante, tenemos la encapsulación de los datos, la interfaz o vista por otro y por último la lógica interna o controlador.

El patrón de arquitectura "modelo vista controlador", es una filosofía de diseño de aplicaciones, compuesta por:

3.7.1 Modelo

- Contiene el núcleo de la funcionalidad (dominio) de la aplicación.
- Encapsula el estado de la aplicación.
- No sabe nada / independiente del Controlador y la Vista.

3.7.2 Vista

- Es la presentación del Modelo.
- Puede acceder al Modelo pero nunca cambiar su estado.
- Puede ser notificada cuando hay un cambio de estado en el Modelo.

3.7.3 Controlador

- Reacciona a la petición del Cliente, ejecutando la acción adecuada y creando el modelo pertinente

Para entender cómo funciona nuestro patrón Modelo vista controlador, se debe entender la división a través del conjunto de estos tres elementos y como estos componentes se comunican unos con los otros y con otras vistas y controladores externos al modelo principal. Para ello, es importante saber que el controlador interpreta las entradas del usuario (tanto teclado como el ratón), enviado el mensaje de acción al modelo y a la vista para que se proceda con los cambios que se consideren adecuados

3.7.4 Comunicación

El modelo, la vista y el controlador deben comunicarse de una manera estable los unos con los otros, de manera que sea coherente con las iteraciones que el usuario realizara. Como es lógico la comunicación entre la vista y el controlador es bastante básica pues están diseñados

para operar juntos, pero los modelos se comunican de una manera diferente, un poco más sutil

3.7.5 Unión del modelo con la vista y el controlador

Como no todos los modelos pueden ser pasivos, necesitamos algo que comunique al controlador y a la vista, por lo que en este caso, sí que necesitamos el modelo, ya que solo este puede llevar a cabo los cambios necesarios al estado actual en el que estos se encuentran.

Al contrario que el modelo, que puede ser asociado a múltiples asociaciones con otras vistas y controladores, cada vista solo puede ser asociada a un único controlador, por lo que han de tener una variable de tipo controler que notificara a la vista cual es su controlador o modelo asignado. De igual manera, el controlador tiene una variable llamada View que apunta a la vista. De esta manera, pueden enviarse mensajes directos el uno al otro y al mismo tiempo, a su modelo.

Al final, la vista es quien lleva la responsabilidad de establecer la comunicación entre los elementos de nuestro patrón MVC. Cuando la vista recibe un mensaje que concierne al modelo o al controlador, lo deja registrado como el modelo con el cual se comunicara y apunta con la variable controller al controlador asignado, enviándole al mismo su identificación para que el controlador establezca en su variable view el identificador de la vista y así puedan operar conjuntamente. El responsable de deshacer estas conexiones, seguirá siendo la vista, quitándose a sí misma como dependiente del modelo y liberando al controlador.

3.8 Struts2

Struts2 se ha utilizado como framework de desarrollo para la aplicación web.

Struts es una herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma Java EE (Java Enterprise Edition). Struts se desarrollaba como parte del proyecto Jakarta de la Apache Software Foundation, pero actualmente es un proyecto independiente conocido como Apache Struts.

Struts permite reducir el tiempo de desarrollo. Su carácter de "software libre" y su compatibilidad con todas las plataformas en las que Java Enterprise esté disponible lo convierten en una herramienta altamente disponible.

Con la versión 2 del framework se introdujeron algunas mejoras sobre la primera versión, de cara a simplificar las tareas más comunes en el desarrollo de aplicaciones web, así como mejorar su integración con AJAX, etc.

3.9 Inyección de dependencias con Spring

En Informática, Inyección de Dependencias (en inglés Dependency Injection, DI) es un patrón de diseño orientado a objetos, en el que se suministran objetos a una clase en lugar de ser la propia clase quien cree el objeto. El término fue acuñado por primera vez por Martin Fowler.

Spring es un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java.2

La primera versión fue escrita por Rod Johnson, quien lo lanzó junto a la publicación de su libro *Expert One-on-One J2EE Design and Development* (Wrox Press, octubre 2002). El framework fue lanzado inicialmente bajo la licencia Apache 2.0 en junio de 2003. El primer gran lanzamiento fue la versión 1.0, que apareció en marzo de 2004 y fue seguida por otros hitos en septiembre de 2004 y marzo de 2005. La versión 1.2.6 de Spring Framework obtuvo reconocimientos Jolt Awards y Jax Innovation Awards en 2006.3 4 Spring Framework 2.0 fue lanzada en 2006, la versión 2.5 en noviembre de 2007, Spring 3.0 en diciembre de 2009, y Spring 3.1 dos años más tarde. La versión actual es 3.2.0.5 . El inicio del desarrollo de la versión 4.0 fue anunciado en enero de 20136

Si bien las características fundamentales de Spring Framework pueden ser usadas en cualquier aplicación desarrollada en Java, existen variadas extensiones para la construcción de aplicaciones web sobre la plataforma Java EE. A pesar que no impone ningún modelo de programación en particular, este framework se ha vuelto popular en la comunidad al ser considerado una alternativa, sustituto, e incluso un complemento al modelo EJB (Enterprise JavaBean).

3.9.1 Módulos

Spring Framework comprende diversos módulos que proveen un rango de servicios, aunque para este proyecto únicamente se ha utilizado el primero “Contenedor de inversión de control”:

- Contenedor de inversión de control: permite la configuración de los componentes de aplicación y la administración del ciclo de vida de los objetos Java, se lleva a cabo principalmente a través de la inyección de dependencias.
- Programación orientada a aspectos: habilita la implementación de rutinas transversales.
- Acceso a datos: se trabaja con RDBMS en la plataforma java, usando Java Database Connectivity y herramientas de Mapeo objeto relacional con bases de datos NoSQL.
- Gestión de transacciones: unifica distintas APIs de gestión y coordina las transacciones para los objetos Java.
- Modelo vista controlador: Un framework basado en HTTP y servlets, que provee herramientas para la extensión y personalización de aplicaciones web y servicios web REST.
- Framework de acceso remoto: Permite la importación y exportación estilo RPC, de objetos Java a través de redes que soporten RMI, CORBA y protocolos basados en HTTP incluyendo servicios web (SOAP).

- Convención sobre configuración: el módulo Spring Roo ofrece una solución rápida para el desarrollo de aplicaciones basadas en Spring Framework, privilegiando la simplicidad sin perder flexibilidad.
- Procesamiento por lotes: a framework for high-volume processing featuring reusable functions including registro/trazado, transaction management, job processing statistics, job restart, skip, and resource management.
- Autenticación and Autorización: procesos de seguridad configurables que soportan un rango de estándares, protocolos, herramientas y prácticas a través del subproyecto Spring Security (formalmente Acegi Security System for Spring).
- Administración Remota: Configuración de visibilidad y gestión de objetos Java para la configuración local o remota vía JMX.
- Mensajes: Registro configurable de objetos receptores de mensajes, para el consumo transparente desde la a través de JMS, una mejora del envío de mensajes sobre las API JMS estándar.
- Testing: Soporte de clases para desarrollo de unidades de prueba e integración.

3.10 Métrica V3

MÉTRICA es una metodología de planificación, desarrollo y mantenimiento de sistemas de información. Promovida por el Ministerio de Administraciones Públicas del Gobierno de España para la sistematización de actividades del ciclo de vida de los proyectos software en el ámbito de las administraciones públicas.

Para realizar esta documentación se ha seguido esta metodología.

3.11

Capítulo 4. Planificación del Proyecto y Resumen de Presupuestos

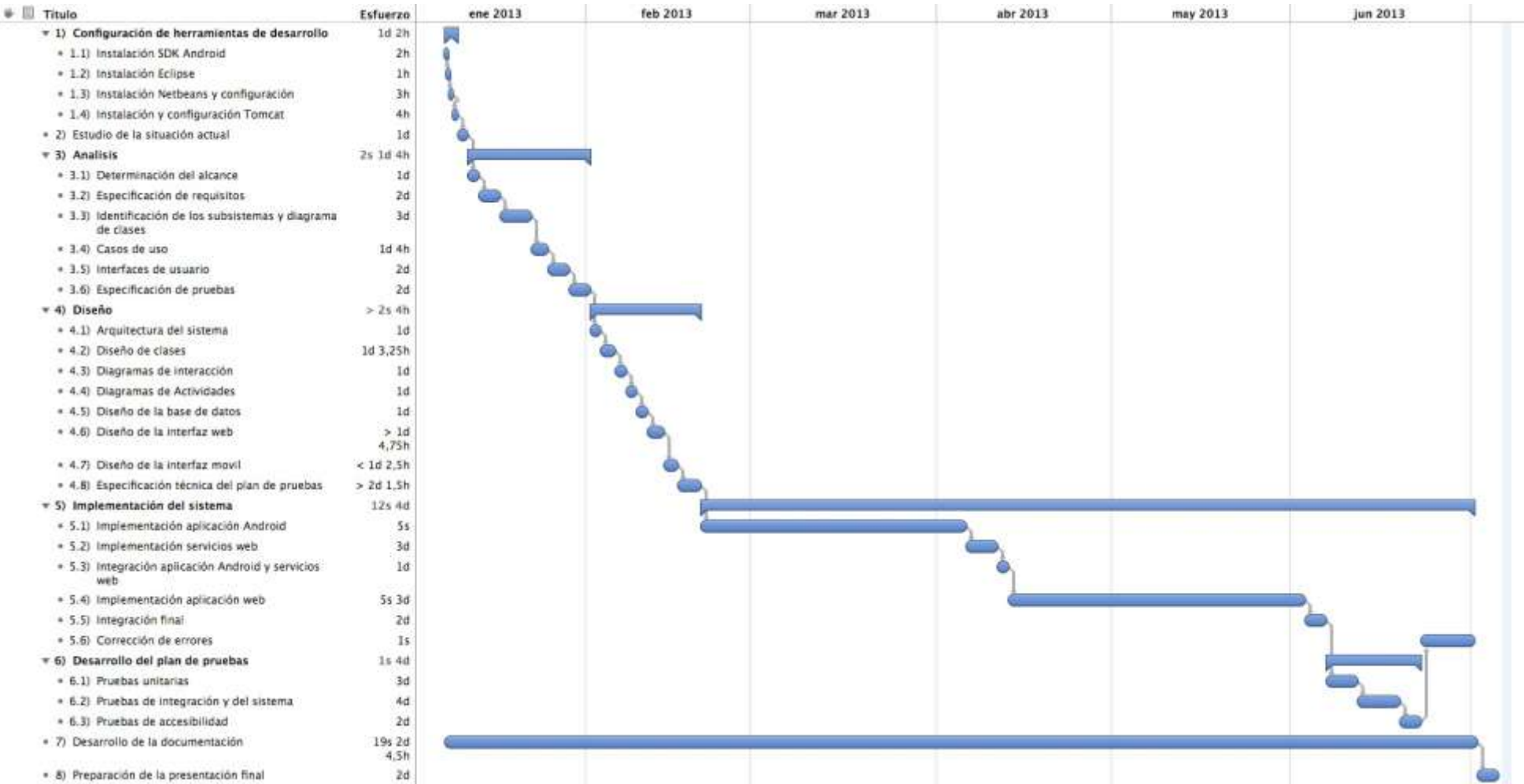
4.1 Planificación

En este apartado se describirán las distintas fases del proyecto teniendo en cuenta el tiempo gastado en cada una de las distintas fases del proyecto.

Para realizar la planificación se ha marcado el horario de tiempo libre que me permite desarrollar el proyecto y haciendo una estimación de cada una de las tareas según la experiencia de proyectos anteriores se ha generado una planificación que se ha intentado seguir para cumplir los plazos de cada una de las fases y poder llegar a la entrega final del proyecto.

Este proyecto al ser desarrollado por una única persona se ha extendido bastante en el tiempo, ya que no se han podido realizar tareas en paralelo. El desarrollo del mismo con varias personas.

Para planificar este proyecto se ha generado un diagrama de Gantt que muestra las diferentes tareas a realizar marcando el tiempo que deberían llevar y una fecha límite. En este diagrama también se puede ver el orden de ejecución de las tareas ya que para poder comenzar alguna de estas es necesario tener acabadas las anteriores, para así tener disponible información o recursos necesarios para la tarea a empezar.



4.2 Resumen del Presupuesto

A continuación se muestra un resumen del presupuesto explicado en el apartado 11. En este presupuesto se incluye el precio de cada una de las categorías presupuestadas.

Ítem	Concepto	Coste Ítem
01	Costes Software	179 €
02	Costes Hardware	376,84 €
03	Recursos Humanos	17.174,00 €
04	Otros	1.080,00 €
	Subtotal	18.809,84€
	IVA (21%)	3.950,06€
	TOTAL	22.759,9€

Capítulo 5. Análisis

5.1 Definición del Sistema

5.1.1 Determinación del Alcance del Sistema

La finalidad del sistema como se ha comentado en apartados anteriores es la de facilitar la compartición de archivos (y texto) que se encuentran geolocalizados, entre varias personas a través de un dispositivo móvil.

A continuación se explican las limitaciones del proyecto y por tanto hasta donde se va a desarrollar.

- La aplicación móvil de Android permitirá sacar fotos geolocalizadas, automática o manualmente, y asignarles un texto descriptivo, un nombre y un lugar en el que se han sacado. Las fotos se podrán sacar desde la cámara del dispositivo con android (tanto frontal como trasera) o desde la galería de imágenes del dispositivo.
- La aplicación móvil se desarrollara únicamente para el sistema operativo Android a partir de la versión 2.3, garantizando el correcto funcionamiento a partir de esta versión de Android.
- El usuario podrá iniciar sesión en la aplicación para exportar sus viajes a través de un email y una contraseña. Podrá cerrar sesión para cambiar el usuario con el que se quiere exportar un mapa.
- No se permitirá la subida de archivos de video debido al gran tamaño que estos pueden alcanzar, y a que se debería desarrollar una cámara de video a medida para este fin. Únicamente se podrán asignar imágenes al mapa.
- Al exportar un mapa se creara una página web para el mismo, la cual se podrá compartir ya que será un enlace publico que no necesitará de un inicio de sesión o registro de usuario para su visualización. En caso de que el mapa sea privado, por petición del usuario, para poder visualizarse se deberá introducir una contraseña previamente escogida por el usuario que ha exportado el mapa.
- El usuario podrá ver todos sus mapas en el panel de administración de su usuario. En este panel podrá eliminar los mapas que ya no quiera que se vean mas.
- Los mapas no se podrán modificar desde la interfaz web ya que esta únicamente se ha creado con el fin de compartir los mapas y sus imágenes no como aplicación de gestión o creación de los mismos.
- Cualquier usuario que tenga acceso a la vista de un mapa podrá comentar en el iniciando sesión con Facebook, yahoo, AOL o Hotmail, y podrá compartir el mismo. No se podrá comentar en un mapa sin tener una cuenta de alguno de estos servicios.
- El usuario únicamente podrá borrar los mapas desde la interfaz web.

- No se contempla la interacción entre usuarios, ya que este proyecto no se plantea como una red social, sino como un complemento de estas. El mapa se podrá compartir en todas las redes sociales.

5.2 Requisitos del Sistema

5.2.1 Obtención de los Requisitos del Sistema

Código	Nombre Requisito	Descripción del Requisito
R1.1	Registro de Usuario	Debe permitirse el registro de usuarios en el sistema para poder asignar los mapas a un determinado individuo.
R1.2	Inicio de sesión aplicación Web	Deben permitirse el inicio de sesión de un usuario en la aplicación web.
R1.3	Inicio de sesión en aplicación Android	Se permitirá el inicio de sesión de un usuario en la aplicación móvil para exportar un mapa.
R2.1	Añadir foto desde la cámara de la aplicación Android	Se permitirá sacar una foto con la cámara desde la aplicación Android para añadirla al mapa. Se podrá añadir la siguiente información a la foto manualmente: lugar, nombre, descripción. Se añadirá la siguiente información automáticamente: fecha, localización (se podrá elegir manualmente en el caso de que no se disponga de señal gps).
R2.2	Añadir foto desde la galería de imágenes de la aplicación Android	Se permitirá añadir una foto desde la galería de imágenes de la aplicación Android para añadirla al mapa. Se podrá añadir la siguiente información a la foto manualmente: lugar, nombre, descripción. Se añadirá la siguiente información automáticamente: fecha, localización (se podrá elegir manualmente en el caso de que no se disponga de señal gps).
R2.3	Eliminar foto	Se eliminará una foto del mapa. Al exportar el mapa se actualizará y se eliminara también de la vista web.
R2.4	Como llegar al lugar donde se sacó una foto	La aplicación móvil permitirá ver el camino de llegada más corto al lugar donde se ha sacado la misma.
R2.5	Compartir imagen	La aplicación móvil podrá compartir una imagen en las diferentes redes sociales, email, y aplicaciones que permitan la compartición de imágenes.
R3.1	Visualizar mapa de imágenes desde aplicación móvil	El usuario podrá ver un mapa con los lugares en los que se sacaron las fotos. Además el usuario tendrá una lista de sus diferentes mapas.
R3.2	Exportación de mapa para su compartición	Se podrá exportar un mapa (público o privado), lo cual creara una página web del mismo que será pública (en caso de no tener contraseña) de modo que se pueda compartir en las diferentes redes sociales.
R3.3	Actualización del mapa en	El usuario una vez haya exportado un mapa podrá

	tiempo real	actualizar el mismo, añadiendo o eliminando fotos, lo que hará que se actualice la página web generada al volver a exportar el mapa.
R4.1	Consulta y gestión de mapas en aplicación web	Se permitirá al usuario ver los mapas exportados en la aplicación web y eliminar los que se deseen. Para modificar el contenido de los mismos se realizara desde la aplicación móvil como se ha comentado anteriormente.
R4.2	Comentarios en mapa	Se podrá comentar en un mapa a través de una caja de comentarios de Facebook, que permite iniciar sesión con cuentas de Facebook, Aol, Yahoo y Hotmail.
R4.3	Generador de código para insertar mapa en otra web	Se permitirá generar un código html de un mapa que pegándolo en una página web permita ver el mapa desde la misma. Este widget permitirá modificar el tamaño del mapa para ajustarse al espacio que necesite el usuario en su web.
R4.4	Diseño del mapa responsivo	El mapa exportado deberá tener un diseño responsivo, es decir, que se adapte dependiendo del tamaño del dispositivo desde el que se acceda.
R5.1	Aplicación Android internacionalizada.	La aplicación móvil deberá estar internacionalizada permitiendo su uso en Español e Inglés.

Además de cumplir los requisitos descritos en la tabla anterior el sistema deberá cumplir los siguientes requisitos no funcionales:

- **Requisitos de Usuario:** La aplicación móvil y web ha de ser lo suficientemente sencilla e intuitiva para que cualquier persona, con unos conocimientos mínimos sobre el uso de un dispositivo con Android, pueda crear sus mapas y compartirlos.
- **Requisitos Tecnológicos:** Para poder instalar la aplicación móvil el dispositivo con Android deberá tener la versión de sistema operativo igual o mayor a la 2.3. Además deberá tener las siguientes características:
 - Tarjetas SD
 - Cámara fotográfica
 - Conexión a internet (para exportar los mapas e iniciar sesión)

Respecto a la aplicación web se podrá ver desde cualquier navegador actual que permita javascript.

- **Requisitos de Usabilidad:** Se deberá poder acceder a los mapas exportados desde cualquier dispositivo con navegador a internet, adaptando el diseño de este a las diferentes resoluciones.
- **Requisitos de Seguridad:** El contenido de los mapas podrá ser público o privado. En caso de que el mapa sea privado únicamente se podrá acceder al mismo introduciendo una contraseña. La lista de mapas de un usuario también ha de estar protegida por un inicio de sesión previo y únicamente podrá ser vista por el usuario que los ha exportado.

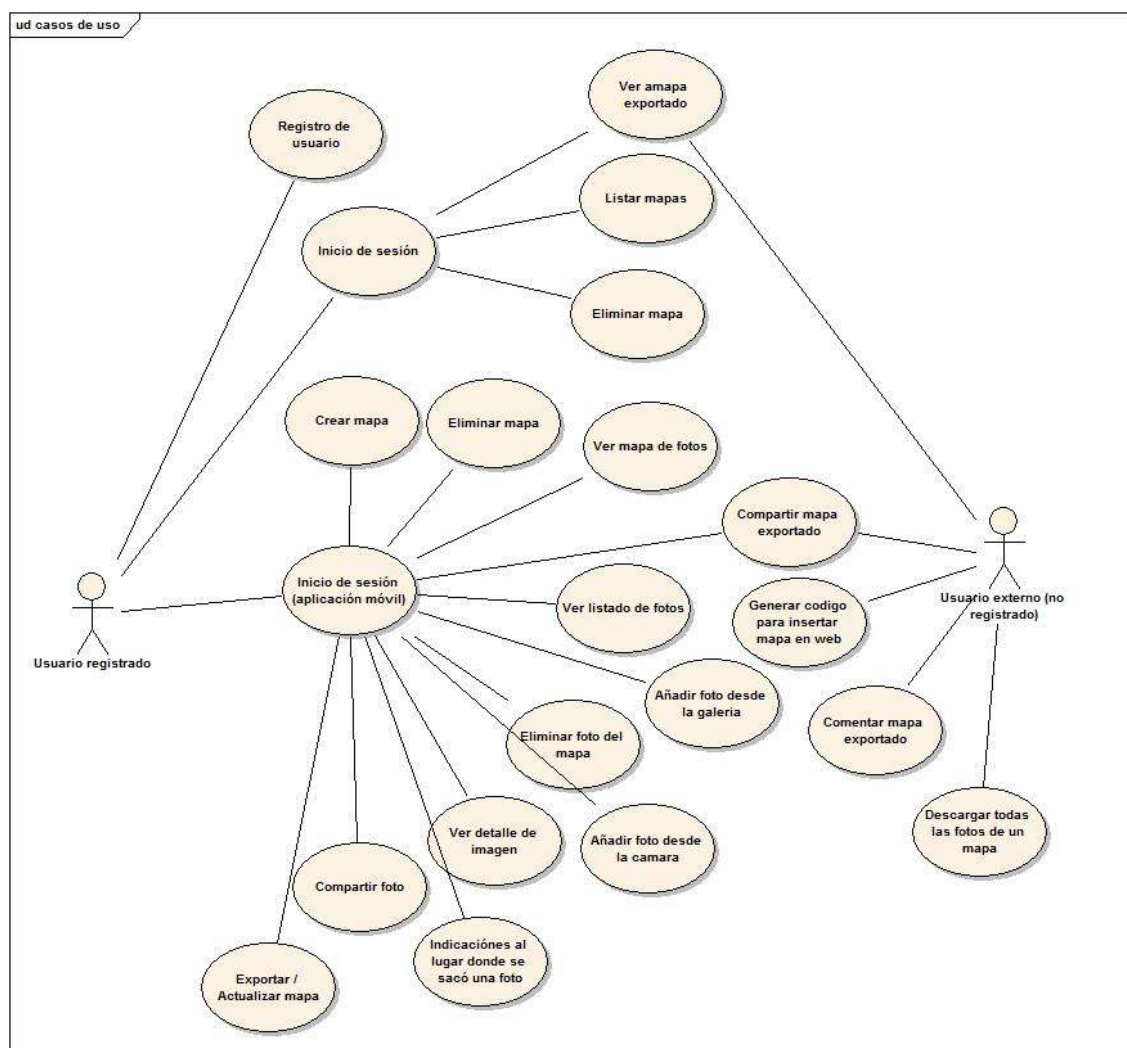
5.2.2 Identificación de Actores del Sistema

En este sistema se identifican los siguientes actores:

- **Usuario registrado:** este usuario podrá crear mapas a través de la aplicación móvil, exportarlos y gestionarlos teniendo total control sobre los mismos.
- **Usuario externo:** cualquier usuario que visualice un mapa generado por un “Usuario registrado”, comparta el mismo o comente a través del puglin de comentarios social. Este usuario no ha de estar registrado ya que el registro no es necesario para la visualización de mapas.

5.2.3 Especificación de Casos de Uso

A continuación se muestran los diferentes casos de uso del sistema, dependiendo del actor implicado en el mismo. Estos casos de uso representan las acciones que puede realizar cada actor en el sistema.



Nombre del Caso de Uso
Registro de usuario
Descripción
A través de un formulario web en el que se pida un email y una contraseña se registrara un nuevo usuario en el sistema. Con este usuario se podrá acceder a la aplicación móvil y al panel de control web.

Nombre del Caso de Uso
Inicio de sesión en aplicación web
Descripción
El usuario introducirá el email y la contraseña, con las que se ha registrado anteriormente, en un formulario web y accederá al panel de control de su usuario donde vera todos los mapas que ha exportado. En caso de que el usuario introduzca erróneamente el email o la contraseña se le volverán a solicitar.

Nombre del Caso de Uso
Ver mapa exportado en la aplicación web
Descripción
Tanto un usuario registrado como un usuario sin registrar podrán ver un mapa exportado. En este mapa se mostraran todas las imágenes exportadas con los siguientes datos: <ul style="list-style-type: none">• Titulo• Descripción• Fecha• Lugar• Imagen Además se mostrara el lugar exacto en el que se sacó cada imagen. Esta pantalla además contendrá las siguientes funcionalidades: <ul style="list-style-type: none">• Descargar zip con todas las imágenes• Generador de código para insertar el mapa en una web externa• Comentarios con Facebook, Aol, Yahoo y Hotmail. El usuario podrá acceder desde cualquier navegador a la página web creada para ver el contenido de un determinado mapa.

Esta página tendrá diseño responsivo por lo que el usuario podrá ver este mapa desde un navegador móvil o uno de escritorio.

Nombre del Caso de Uso

Eliminar mapa

Descripción

Una vez que el usuario haya iniciado sesión en el sistema y accedido a su panel de control podrá eliminar un mapa después de confirmar la acción. El mapa se eliminará del sistema y por tanto dejara de ser visible.

Nombre del Caso de Uso

Listar mapas

Descripción

Desde la aplicación móvil el usuario accederá a una pantalla donde se listarán todos sus mapas y desde donde podrá acceder a cada uno de ellos o eliminar un mapa determinado.

Nombre del Caso de Uso

Iniciar sesión desde la aplicación móvil
--

Descripción

El usuario cuando vaya a exportar el mapa se le mostrará un formulario desde el cual identificarse. Al igual que en el inicio de sesión en la aplicación web el usuario introducirá su nombre y contraseña y accederá al panel de exportación de un mapa.

Nombre del Caso de Uso
Crear nuevo mapa
Descripción
El introducirá un nombre para su nuevo mapa y se le añadirá a la lista de mapas de la aplicación móvil, desde la cual podrá acceder a cada uno de ellos.

Nombre del Caso de Uso
Eliminar mapa
Descripción
Se eliminará un mapa de la aplicación móvil y desaparecerá del listado de mapas, junto con sus imágenes. Este mapa no se borrará de la aplicación web sino que únicamente desaparecerá de la aplicación. Para borrar el mapa de la web se tendrá que acceder al panel de control del usuario en la web.

Nombre del Caso de Uso
Ver mapa de fotos
Descripción
Cuando el usuario seleccione uno de los mapas que tiene almacenados accederá a una primera vista en la que se mostrarán marcadores en el lugar en el que se sacaron las fotos puestas sobre un mapa. Pinchando en cada marcador podrá ver la foto que se ha sacado en ese punto. Si se pulsa sobre la foto se accederá a una vista de detalle de la misma.

Nombre del Caso de Uso
Ver listado de fotos
Descripción
El usuario podrá elegir ver las fotos que ha sacado en forma de listado. Estas se ordenarán según la fecha en la que hayan sido sacadas y pulsando en cada elemento de este listado se accederá a la vista de detalle de la foto. Desde esta vista se podrá eliminar una imagen del mapa.

Nombre del Caso de Uso
Añadir foto desde la cámara
Descripción
Una vez accedido a un mapa en la aplicación móvil se podrá añadir una nueva foto al mapa.

Se abrirá la cámara del dispositivo móvil y se guardará la foto sacada. Una vez guardada esta imagen se pedirá al usuario que introduzca los siguientes datos:

- Título
- Comentario
- Lugar

Una vez que el usuario pulse guardar, la imagen se añadirá al mapa de imágenes.

Nombre del Caso de Uso

Añadir foto desde la galería de imágenes
--

Descripción

Una vez accedido a un mapa en la aplicación móvil se podrá añadir una nueva foto al mapa. Se abrirá la galería de imágenes del dispositivo móvil y cuando el usuario seleccione una imagen esta se guardará. Una vez guardada esta imagen se pedirá al usuario que introduzca los siguientes datos:

- Título
- Comentario
- Lugar

Después de rellenar estos datos la imagen se añadirá al mapa de imágenes.

Nombre del Caso de Uso

Eliminar foto

Descripción

Desde el listado de imágenes se podrá eliminar una foto. Esta foto desaparecerá también del mapa exportado al volver a exportar el mapa, es decir, al actualizarlo.

Nombre del Caso de Uso

Ver detalle de imagen

Descripción

El usuario podrá acceder tanto desde un listado de imágenes, como desde el mapa a la vista de detalle de una imagen en la que se mostrará lo siguiente:

- Título
- Comentario

- Fecha
- Lugar
- Imagen

Desde esta pantalla se podrá:

- Compartir la imagen
- Ver indicaciones al lugar en el que se sacó la imagen

Nombre del Caso de Uso
Ver indicaciones al lugar en el que se sacó una foto
Descripción
Desde la vista de detalle de una foto se podrá acceder a las indicaciones para llegar al lugar en el que se ha sacado esa foto. Esto mostrará una ruta desde la situación actual del usuario hasta el punto exacto de la toma.

Nombre del Caso de Uso
Compartir foto
Descripción
Desde la vista de detalle de una foto el usuario podrá compartir dicha foto utilizando las aplicaciones de su dispositivo móvil que lo permitan, como por ejemplo Facebook, Gmail, Twitter, Whatsapp...

Nombre del Caso de Uso
Exportar mapa
Descripción
Si el usuario ha iniciado sesión podrá exportar el mapa. Esto significa que se enviarán las imágenes al servidor y se creará automáticamente una página web para compartir el mapa. Cuando el mapa haya sido exportado completamente se le facilitara la url del mapa al usuario y un botón de compartir con el cual podrá compartir este enlace en las distintas redes sociales y aplicaciones de su dispositivo que lo permitan. A la hora de exportar un mapa el usuario podrá decidir si este es privado o no, en el caso de que sea privado el usuario introducirá una contraseña para protegerlo.

Nombre del Caso de Uso
Actualizar mapa

Descripción
En el caso de que el usuario ya haya exportado un mapa y vuelva a pulsar el botón de exportar, en lugar de crear una nueva página para un nuevo mapa, se actualizará el contenido del mapa exportado anteriormente permitiendo así al usuario mantener actualizados en tiempo real sus mapas en el momento que este lo elija.

Nombre del Caso de Uso
Compartir mapa exportado en las redes sociales
Descripción
Tanto un usuario registrado como un usuario sin registrar al acceder a la vista de un mapa exportado podrán compartir el mismo en las distintas redes sociales a través de una herramienta que dejará seleccionar entre las redes sociales o servicios de compartición disponibles.

Nombre del Caso de Uso
Comentar mapa exportado
Descripción
En la vista de mapa exportado se dispondrá de una herramienta que permita a los visitantes del mismo dejar un comentario. Los comentarios serán guardados por cada mapa.

Nombre del Caso de Uso
Generar código para insertar mapa en otra web
Descripción
En la vista de mapa exportado se encontrara un widget que permitirá generar un código HTML que al pegarlo en cualquier otra web deje ver el mapa como si de un visor se tratase. El widget permitirá elegir el ancho y alto del mapa a insertar, permitiendo también que el mapa se ajuste a la estructura de la página automáticamente.

Nombre del Caso de Uso
Descargar todas las fotos de un mapa
Descripción
En la vista de mapa exportado se presentará un botón que permita descargar un único archivo zip con todas las imágenes del mapa.

5.3 Identificación de los Subsistemas en la Fase de Análisis

5.3.1 Descripción de los Subsistemas

En esta sección se analizarán los diferentes subsistemas en los que se ha descompuesto el sistema final.

5.3.1.1 Aplicación móvil cliente

La Aplicación móvil es el cliente de recolección de datos, es el encargado de recoger los datos del usuario y almacenarlos localmente hasta que este decida exportarlos. Este subsistema tiene las siguientes funciones:

- Creación y gestión de mapas
- Creación y gestión de fotos en cada mapa
- Recolección automática de datos relacionados con las imágenes, como fecha, localización... y de datos manuales asignados a una foto (titulo, comentario, lugar)
- Vista de mapa completa
- Vista de listado de fotos para un determinado mapa
- Indicaciones a el lugar de una foto
- Compartir una foto

Al exportar un mapa este subsistema envía a el subsistema de recepción y envió de datos desde el servidos tanto los datos del mapa creado, como las imágenes tomadas y sus datos asociados (titulo, comentario, lugar, fecha, localización...)

Además este subsistema será el encargado de sincronizar los datos locales con los datos en el servidor indicando que imágenes se han de borrar del mapa exportado y cuales se han de añadir en el caso de que se esté actualizando el mapa.

5.3.1.2 Subsistema de recepción y envió de datos desde el servidor

Este subsistema se encargará de recibir los datos enviados por el subsistema de exportación y tratarlos para así almacenarlos y poder generar la pagina web para el mapa exportado. También es el encargado de enviar información solicitada por las aplicaciones clientes sobre un usuario, un mapa o una foto.

Una vez recibidos los datos del cliente procesará los mismos y enviará esta información al subsistema de gestión de base de datos para que estos sean almacenados correctamente.

5.3.1.3 Subsistema de gestión de base de datos

El subsistema de gestión de base de datos recibe los datos del subsistema de recepción de datos en el servidor y se encarga de crear los directorios necesarios para almacenar las fotos de los mapas y de almacenar toda la información de los mapas y fotos en el servidor. Como es lógico también provee estos datos en el momento en el que se quiere acceder a un mapa exportado de un usuario.

Este subsistema también controla los datos almacenados sobre los usuarios.

El subsistema de gestión de base de datos se relaciona con el subsistema de recepción de datos en el servidor, con el subsistema de seguridad y con el subsistema de visualización de mapas.

5.3.1.4 Subsistema de seguridad

El subsistema de seguridad es el encargado de controlar el acceso a los diferentes contenidos de las aplicaciones. Este subsistema comprende desde el registro de usuario en la aplicación web hasta el inicio de sección en la aplicación móvil para poder realizar una exportación y el inicio de sesión en la aplicación web para acceder al panel de control de un usuario.

Además este subsistema es el encargado de proteger los mapas privados comprobando que se introduce la contraseña correcta para su visualización.

Este subsistema se comunica con el subsistema de gestión de base de datos para recibir los datos de los usuarios y enviárselos para su registro y con el subsistema de gestión y visualización de mapas. También se comunica con el subsistema de recepción de datos en el servidor para asegurarse de que los datos enviados se envían al usuario correcto y que este está correctamente identificado.

5.3.1.5 Subsistema de gestión y visualización de mapas

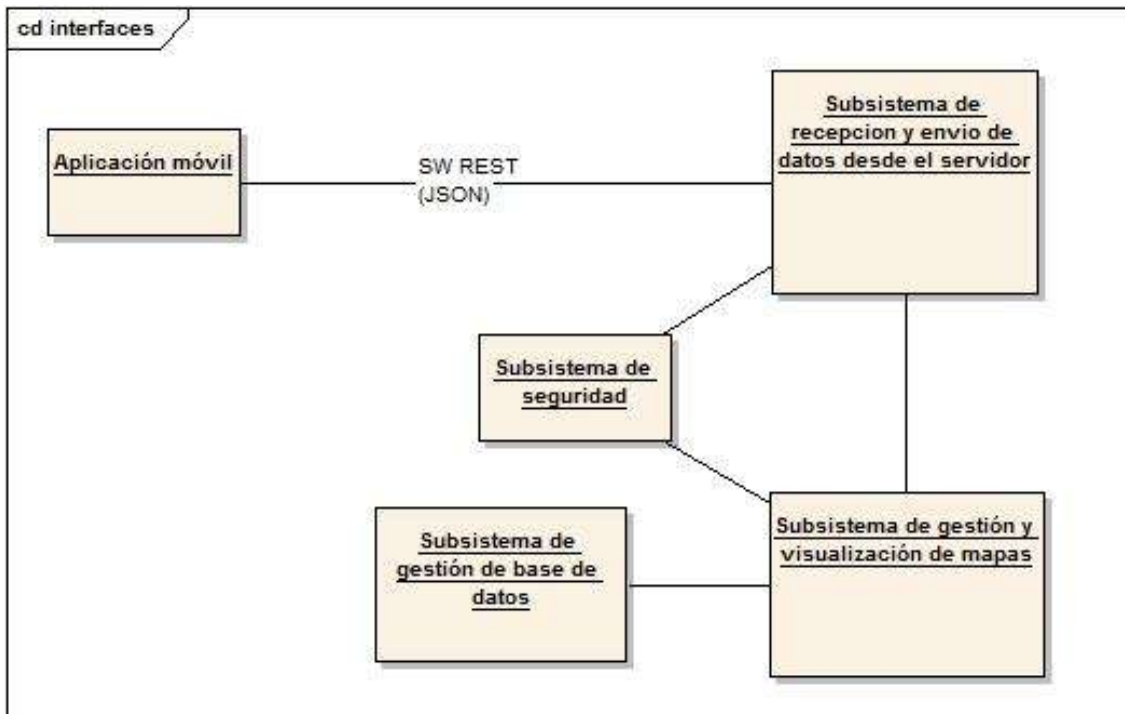
La función principal de este subsistema es la gestión de los mapas de un usuario controlando también el inicio de sesión del mismo y el registro.

Este subsistema se encargará de generar la vista de un mapa exportado de forma que pueda ser visible desde cualquier navegador. Además en este subsistema se gestionará la generación de código para insertar un mapa en otra web y los comentarios en un mapa.

Este subsistema recibe los datos del subsistema de gestión de base de datos y los procesa para generar un mapa en el que se muestre toda la información recogida por la aplicación móvil cliente y anteriormente exportada por el subsistema de exportación y recibida por el subsistema de recepción y envió de datos desde el servidor.

5.3.2 Descripción de los Interfaces entre Subsistemas

En la siguiente imagen se puede ver de forma grafica como se comunicarán los diferentes subsistemas entre ellos.



La Aplicación móvil guarda los datos recolectados en una base de datos en el dispositivo móvil, cuando esta tenga los datos del mapa preparados para su exportación se enviarán al servidor a través de Servicios Web REST con formato JSON. De esta forma es posible la comunicación entre el servidor y la aplicación cliente.

Una vez recibidos los datos del cliente el Subsistema de recepción y envío de datos desde el servidor envía estos datos al Subsistema de Gestión de Base de Datos que se encuentra en el mismo servidor para que estos se almacenen. La comunicación entre estos dos subsistemas es trivial ya que comparten el mismo código.

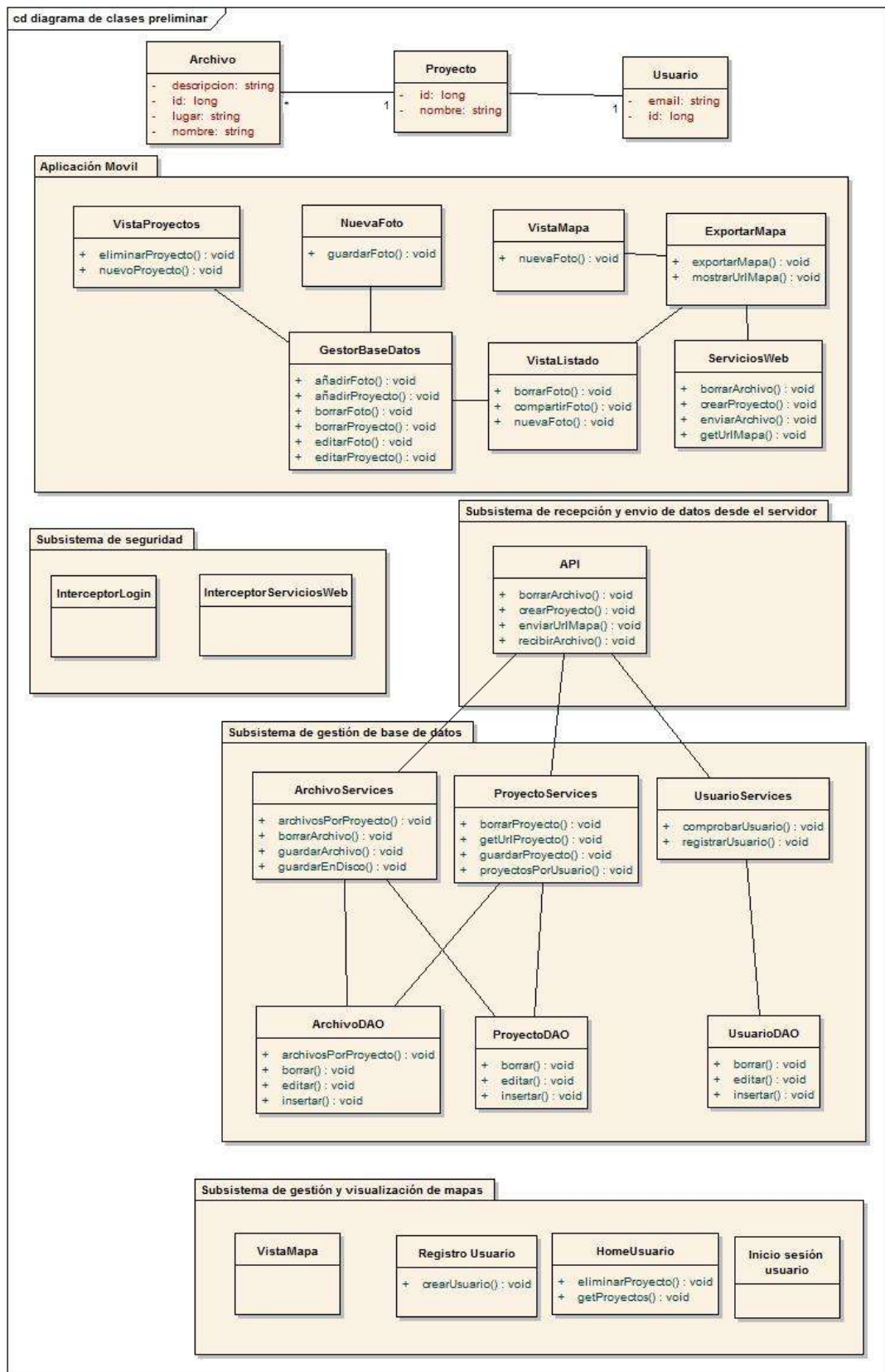
El Subsistema de gestión y visualización de mapas hace peticiones de los elementos que necesita al Subsistema de gestión de base de datos de forma que tenga todos los datos necesarios para generar el mapa y gestionar todo lo relacionado con este.

El Subsistema de Seguridad actúa como capa intermedia entre las comunicaciones de la aplicación móvil y el subsistema de recepción y envío de datos desde el servidor,

comprobando las credenciales enviadas en las peticiones de manera segura. Las peticiones desde el cliente tendrán un token que será validado en el servidor para comprobar que se tiene permiso para realizar una acción. De la misma forma el Subsistema de Seguridad actuara de capa intermedia entre el usuario y el Subsistema de visualización de mapas permitiendo únicamente ver los mapas públicos o pidiendo contraseña para los mapas privados, comprobando que esta contraseña es correcta mediante peticiones al Subsistema de Gestión de Base de Datos que tiene almacenados los datos del usuario.

5.4 Diagrama de Clases Preliminar del Análisis

5.4.1 Diagrama de Clases



5.5 Análisis de Casos de Uso y Escenarios

5.5.1 Registro de usuario

Registro de usuario	
Precondiciones	No
Poscondiciones	El usuario se registrará en el sistema e iniciará sesión automáticamente.
Actores	Iniciado por un usuario externo al sistema, es decir, sin un usuario sin registrar.
Descripción	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla de registro. 2. El usuario introduce su nombre , su clave y repite su clave 3. El sistema valida la información introducida comprobando que el usuario introducido no se encuentra ya registrado en el sistema 4. El usuario entra correctamente en el sistema e inicia sesión automáticamente.
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> • Usuario existente: Alta errónea porque el usuario ya está registrado <ul style="list-style-type: none"> ○ Notificar el hecho al usuario ○ Volver a la página de registro. • Email no valido: El email que se ha introducido no es valido <ul style="list-style-type: none"> ○ Notificar el hecho al usuario sin borrar el formulario • Las contraseñas no coinciden: Las contraseñas introducidas no son iguales. <ul style="list-style-type: none"> ○ Notificar el hecho al usuario sin borrar el formulario • Campo vacío: Algún campo del formulario se encuentra vacío <ul style="list-style-type: none"> ○ Notificar al usuario sin borrar el formulario

5.5.2 Inicio de sesión en aplicación web

Inicio de sesión en aplicación web	
Precondiciones	No
Poscondiciones	El usuario se registrará en el sistema e iniciará sesión automáticamente.
Actores	Iniciado por un usuario registrado del sistema.

Descripción	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla de inicio de sesión. 2. El usuario introduce su nombre y su contraseña 3. El sistema valida la información introducida comprobando que el usuario introducido esta registrado 4. El usuario entra correctamente en el sistema y se le redirige al panel de control donde puede ver sus mapas exportados
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> • Usuario no existente: El usuario no está registrado en el sistema <ul style="list-style-type: none"> ○ Notificar el hecho al usuario ○ Volver a la página de inicio de sesión.

5.5.3 Ver mapa exportado en la aplicación web

Ver mapa exportado en la aplicación web	
Precondiciones	No
Poscondiciones	El usuario accederá al mapa donde podrá ver las fotos exportadas, los comentarios y acceder a la descarga completa del Zip de fotos.
Actores	Tanto en usuario registrado como un usuario externo al sistema pueden acceder a un mapa exportado.
Descripción	<p>Al introducir la url de un mapa este se mostrará con distinto layout dependiendo del tamaño del dispositivo móvil.</p> <p>En este mapa se mostraran todas las imágenes exportadas con los siguientes datos:</p> <ul style="list-style-type: none"> • Titulo • Descripción • Fecha • Lugar • Imagen <p>Además se mostrara el lugar exacto en el que se sacó cada imagen.</p> <p>Esta pantalla además contendrá las siguientes funcionalidades:</p> <ul style="list-style-type: none"> • Descargar zip con todas las imágenes

	<ul style="list-style-type: none"> • Generador de código para insertar el mapa en una web externa • Comentarios con Facebook, Aol, Yahoo y Hotmail.
Variaciones (escenarios secundarios)	En caso de que el mapa sea privado se mostrara un formulario para introducir la clave de este. Si el usuario introduce la clave correctamente se mostrará el mapa, en caso contrario se volverá a pedir esta contraseña.

5.5.4 Listar mapas

Listar mapas	
Precondiciones	El usuario ha de haber iniciado sesión en la aplicación web
Poscondiciones	El usuario accederá al panel de control en el cual se listarán todos los mapas que este haya exportado.
Actores	Usuario registrado
Descripción	<p>Después de iniciar sesión si esta tarea se ha realizado correctamente el usuario accederá a un panel en el que se listarán todos los mapas exportados mostrando de ellos lo siguiente:</p> <ul style="list-style-type: none"> • Nombre • Botón para ver mapa • Botón para eliminar mapa

5.5.5 Eliminar mapa

Eliminar mapa	
Precondiciones	El usuario ha de haber iniciado sesión en la aplicación web.
Poscondiciones	El usuario eliminará un mapa del sistema, dejando este mapa de ser accesible.
Actores	Usuario registrado
Descripción	<p>Una vez que el usuario haya accedido a su panel de control el usuario deberá realizar las siguientes tareas para eliminar un mapa:</p> <ol style="list-style-type: none"> 1. Pulsar en el botón de eliminar un mapa 2. Se mostrará un dialogo de confirmación 3. Pulsar en el botón de confirmar

	4. El mapa se eliminará y se volverán a listar todos los mapas del usuario.
--	---

5.5.6 Inicio de sesión desde la aplicación móvil

Inicio de sesión desde la aplicación móvil	
Precondiciones	No
Poscondiciones	El usuario accederá al panel de exportación de mapas.
Actores	Usuario registrado
Descripción	<p>Cuando el usuario haya creado un mapa, y añadido fotos en el podrá pulsar sobre el botón de exportar mapa, si el usuario no ha iniciado sesión antes se le mostrara una pantalla de login, el usuario introducirá su email y contraseña y en el caso de introducirlos correctamente accederá al panel de exportación de un mapa.</p>
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> • Usuario o contraseña erróneos: Si el usuario o la contraseña son erróneos: <ul style="list-style-type: none"> ○ Se notificará al usuario ○ Se volverá a mostrar el login

5.5.7 Crear nuevo mapa

Crear nuevo mapa	
Precondiciones	No
Poscondiciones	El usuario añadirá un nuevo mapa a la aplicación móvil.
Actores	Usuario registrado y usuario externo (no registrado)
Descripción	<p>Al abrir la aplicación móvil se mostrara el listado de mapas, en esta pantalla el usuario tendrá que realizar lo siguiente para crear un mapa:</p> <ol style="list-style-type: none"> 1. Pulsar sobre el botón crear mapa 2. Introducir nombre del mapa 3. Se creará el mapa y se añadirá a la lista

5.5.8 Eliminar mapa (aplicación móvil)

Eliminar mapa (aplicación móvil)	
Precondiciones	No
Poscondiciones	Se eliminara el mapa de la aplicación móvil.
Actores	Usuario registrado y usuario externo (no registrado)
Descripción	<p>El usuario pulsará sobre el botón de eliminar mapa y este se eliminará junto con todas las fotos y datos asignados a él.</p> <p>Este mapa no se eliminará de la aplicación web sino que solo se eliminara del dispositivo móvil para liberar memoria.</p>

5.5.9 Ver mapa de fotos (aplicación móvil)

Ver mapa de fotos (aplicación móvil)	
Precondiciones	No
Poscondiciones	No
Actores	Usuario registrado y usuario externo (no registrado)
Descripción	<p>El usuario accederá a un mapa en el que se encontraran posicionados marcadores en los lugares en los que se ha sacado una foto.</p> <p>Pulsando sobre estos marcadores aparecerá una previsualización de la imagen que se ha tomado en ese lugar.</p> <p>Pulsando sobre la imagen se accederá a la vista de detalles de la misma.</p> <p>Estos marcadores se encontrarán unidos marcando un camino en el orden en el que las fotos se fueron añadiendo al mapa.</p>

5.5.10 Ver listado de fotos (aplicación móvil)

Ver listado de fotos (aplicación móvil)	
Precondiciones	No
Poscondiciones	No
Actores	Usuario registrado y usuario externo (no registrado)
Descripción	<p>Se listarán las fotos añadidas al mapa en orden en el que fueron tomadas, mostrando de cada una la siguiente información en el listado:</p> <ul style="list-style-type: none"> • Día y mes en el que fueron sacadas • Nombre de la foto

	<ul style="list-style-type: none"> • Lugar de la foto • <p>Desde esta pantalla también se podrá acceder a la vista de detalle pulsando sobre un elemento de la lista y se podrá añadir una nueva foto pulsando sobre el botón de añadir foto.</p>
--	---

5.5.11 Añadir foto desde la cámara (aplicación móvil)

Añadir foto desde la cámara (aplicación móvil)	
Precondiciones	No
Poscondiciones	Se añadirá una nueva foto al mapa
Actores	Usuario registrado y usuario externo (no registrado)
Descripción	<ol style="list-style-type: none"> 1. El usuario accederá al formulario de añadir nueva foto 2. Seleccionara sacar foto con la cámara 3. Sacara la foto 4. Rellenara los campos <ol style="list-style-type: none"> a. Nombre b. Comentario c. Lugar 5. Guardará y se añadirá la foto al mapa
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> • Nombre o lugar vacío: El campo nombre o lugar están vacíos: <ul style="list-style-type: none"> ○ Se mostrará un error al usuario indicando que esos campos son obligatorios. • No se encuentra la localización: No se encuentra la localización actual del usuario. <ul style="list-style-type: none"> ○ Se deberá introducir manualmente desde la pantalla de añadir localización

5.5.12 Añadir foto desde la galería de imágenes (aplicación móvil)

Añadir foto desde la galería de imágenes (aplicación móvil)	
Precondiciones	No
Poscondiciones	Se añadirá una nueva foto al mapa
Actores	Usuario registrado y usuario externo (no registrado)
Descripción	<ol style="list-style-type: none"> 1. El usuario accederá al formulario de añadir nueva foto 2. Seleccionará sacar foto desde la galería de imágenes 3. Seleccionará una foto 4. Rellenara los campos <ol style="list-style-type: none"> a. Nombre b. Comentario c. Lugar 5. Guardará y se añadirá la foto al mapa
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> • Nombre o lugar vacío: El campo nombre o lugar están vacíos: <ul style="list-style-type: none"> ○ Se mostrará un error al usuario indicando que esos campos son obligatorios. • No se encuentra la localización: No se encuentra la localización actual del usuario. <ul style="list-style-type: none"> ○ Se deberá introducir manualmente desde la pantalla de añadir localización

5.5.13 Eliminar foto del mapa (aplicación móvil)

Eliminar foto del mapa (aplicación móvil)	
Precondiciones	No
Poscondiciones	Se eliminará una foto del mapa
Actores	Usuario registrado y usuario externo (no registrado)
Descripción	<ol style="list-style-type: none"> 1. El usuario accederá al listado de fotos 2. Pulsará sobre el botón eliminar

	<ol style="list-style-type: none"> 3. La foto se eliminará y se recargara la lista de fotos 4. Cuando se vuelva a exportar (actualizar) el mapa se eliminará esta foto también del mapa exportado anteriormente
--	---

5.5.14 Ver detalle de imagen (aplicación móvil)

Ver detalle de imagen (aplicación móvil)	
Precondiciones	No
Poscondiciones	No
Actores	Usuario registrado y usuario externo (no registrado)
Descripción	<ol style="list-style-type: none"> 1. El usuario accederá al mapa o al listado de imágenes 2. Seleccionará una imagen para ver su detalle 3. Se mostrará: <ol style="list-style-type: none"> a. Imagen b. Lugar de la imagen c. Nombre d. Comentario e. Fecha y hora <p>Desde esta pantalla se podrá:</p> <ul style="list-style-type: none"> • Compartir la imagen • Ver indicaciones al lugar en el que se sacó la imagen

5.5.15 Ver indicaciones al lugar en el que se sacó una foto (aplicación móvil)

Ver detalle de imagen (aplicación móvil)	
Precondiciones	No
Poscondiciones	Se muestra un mapa con las indicaciones al lugar de la foto
Actores	Usuario registrado y usuario externo (no registrado)
Descripción	<ol style="list-style-type: none"> 1. Se abrirá una lista de mapa de google maps 2. Se indicará el modo de indicaciones

	<ol style="list-style-type: none"> a. Turismo b. Transporte urbano c. A pie <p>3. Se mostrara la ruta de la situación actual a la situación donde fue sacada la imagen</p>
Variaciones (escenarios secundarios)	En caso de que no se encuentre la posición actual se podrá seleccionar una posición de la que partir.

5.5.16 Compartir foto (aplicación móvil)

Compartir foto (aplicación móvil)	
Precondiciones	El usuario ha de tener alguna aplicación con la que compartir la foto instalada en el dispositivo
Poscondiciones	La foto se compartirá con la aplicación seleccionada
Actores	Usuario registrado y usuario externo (no registrado)
Descripción	<p>En la vista de detalle de una foto</p> <ol style="list-style-type: none"> 1. Pulsar compartir 2. Seleccionar aplicación con la que compartir 3. Compartir foto

5.5.17 Exportar mapa (aplicación móvil)

Exportar mapa (aplicación móvil)	
Precondiciones	No
Poscondiciones	Se creará un mapa en la aplicación web que será visible a través de una url que se le proporciona al usuario que exporta el mapa
Actores	Usuario registrado
Descripción	<p>En la vista de mapa o listado de fotos:</p> <ol style="list-style-type: none"> 1. Pulsar exportar mapa 2. Seleccionar si el mapa es privado o publico <ol style="list-style-type: none"> a. En caso de que sea privado introducir contraseña 3. Pulsar exportar 4. Se enviarán las fotos al servidor mientras se muestra un

	<p>mensaje de cargando</p> <p>5. Se mostrará al usuario la url del mapa exportado y un botón para compartir este</p>
Variaciones (escenarios secundarios)	<p>Si el mapa ya ha sido exportado se le indicara al usuario de que esta vez en lugar de crearse un nuevo mapa se actualizará el mapa anterior añadiendo o eliminando las imágenes que se hayan cambiado en la aplicación móvil.</p> <p>En caso de que haya ocurrido algún error durante la exportación se le comunicará al usuario con un mensaje en esta misma pantalla.</p>

5.5.18 Actualizar mapa (aplicación móvil)

Actualizar mapa (aplicación móvil)	
Precondiciones	El mapa debe haber sido exportado anteriormente y por tanto estar creado en la aplicación web
Poscondiciones	Se actualizará el mapa en la aplicación web
Actores	Usuario registrado
Descripción	<p>En la vista de mapa o listado de fotos:</p> <ol style="list-style-type: none"> 1. Pulsar exportar mapa 2. Pulsar actualizar 3. Se enviarán las fotos al servidor mientras se muestra un mensaje de cargando 4. Se mostrará al usuario la url del mapa exportado y un botón para compartir este
Variaciones (escenarios secundarios)	En caso de que haya ocurrido algún error durante la exportación se le comunicará al usuario con un mensaje en esta misma pantalla.

5.5.19 Compartir mapa exportado en las redes sociales

Compartir mapa exportado en las redes sociales	
Precondiciones	No
Poscondiciones	No

Actores	Usuario registrado, usuario no registrado
Descripción	<p>En la vista de mapa web:</p> <ol style="list-style-type: none"> 1. Pulsar en cualquier opción del plugin social para compartir en el servicio seleccionado 2. En caso de no estar el servicio seleccionado en los botones que se muestren se podrá elegir otro servicio a través de un popup

5.5.20 Comentar mapa exportado

Comentar mapa exportado	
Precondiciones	No
Poscondiciones	Se añadirá el comentario del usuario a la lista de comentarios del mapa
Actores	Usuario registrado, Usuario no registrado
Descripción	<p>En la vista de mapa web:</p> <ol style="list-style-type: none"> 1. Pulsar en comentarios 2. Iniciar sesión en el plugin de comentarios 3. Escribir comentario 4. Pulsar en comentar

5.5.21 Generar código para insertar mapa en otra web

Generar código para insertar mapa en otra web	
Precondiciones	No
Poscondiciones	Se generará un código html que se podrá insertar en una web externa a map it
Actores	Usuario registrado, Usuario no registrado
Descripción	<p>En la vista de mapa web:</p> <ol style="list-style-type: none"> 1. Pulsar en instar mapa 2. Seleccionar las medidas del mapa a generar <ol style="list-style-type: none"> a. El código del mapa cambiará automáticamente cuando se cambien las medidas de este

	3. Copiar código generado del mapa
Variaciones	Se podrá seleccionar que el mapa a insertar se ajuste al ancho de la pantalla.

5.5.22 Descargar todas las fotos de un mapa

Descargar todas las fotos de un mapa	
Precondiciones	No
Poscondiciones	Se descargará un único archivo zip con todas las fotos que contiene el mapa
Actores	Usuario registrado, Usuario no registrado
Descripción	<p>En la vista de mapa web:</p> <ol style="list-style-type: none"> 1. Pulsar en descargar archivos 2. Seleccionar ruta de descarga

5.6 Análisis de Interfaces de Usuario

5.6.1 Descripción de la Interfaz

5.6.1.1 Aplicación móvil

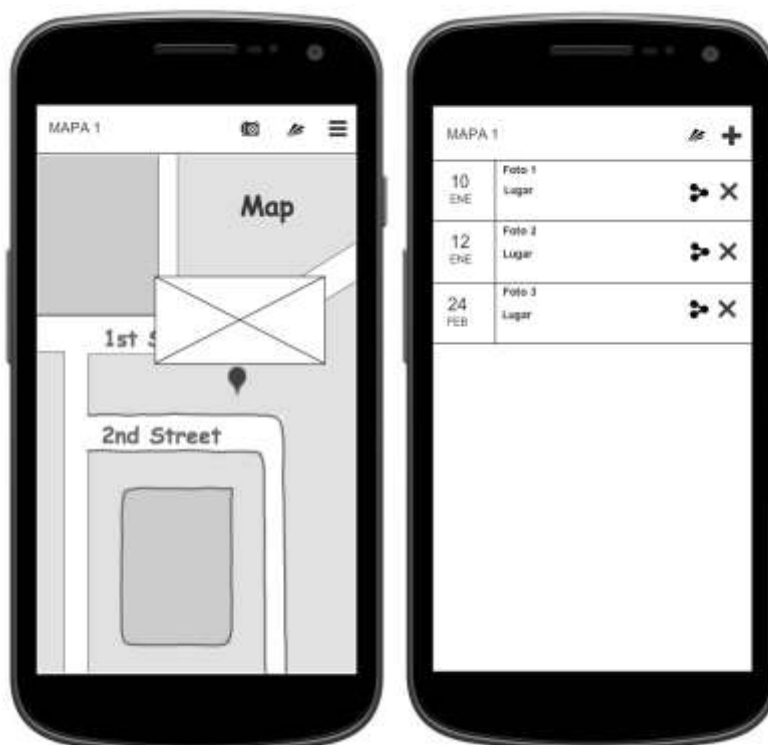
A continuación se explican las pantallas a desarrollar en la aplicación móvil cliente del sistema.



La pantalla principal de la aplicación mostrará los mapas creados por el usuario teniendo cada uno de ellos un botón para eliminar este mapa. En la esquina superior derecha se encuentra el botón de añadir mapa.



Si pulsamos sobre el botón “+” de añadir mapa se mostrara un popup que nos pedirá el nombre del mapa. Una vez se introduzca este nombre se añadirá un nuevo mapa al listado.



Las pantallas anteriores muestran las imágenes tomadas en el mapa de las dos formas posibles en la aplicación.

En la pantalla de la izquierda vemos la pantalla principal del mapa donde se pueden ver marcadores en el lugar en el que se sacaron las imágenes. Pulsando sobre uno de estos

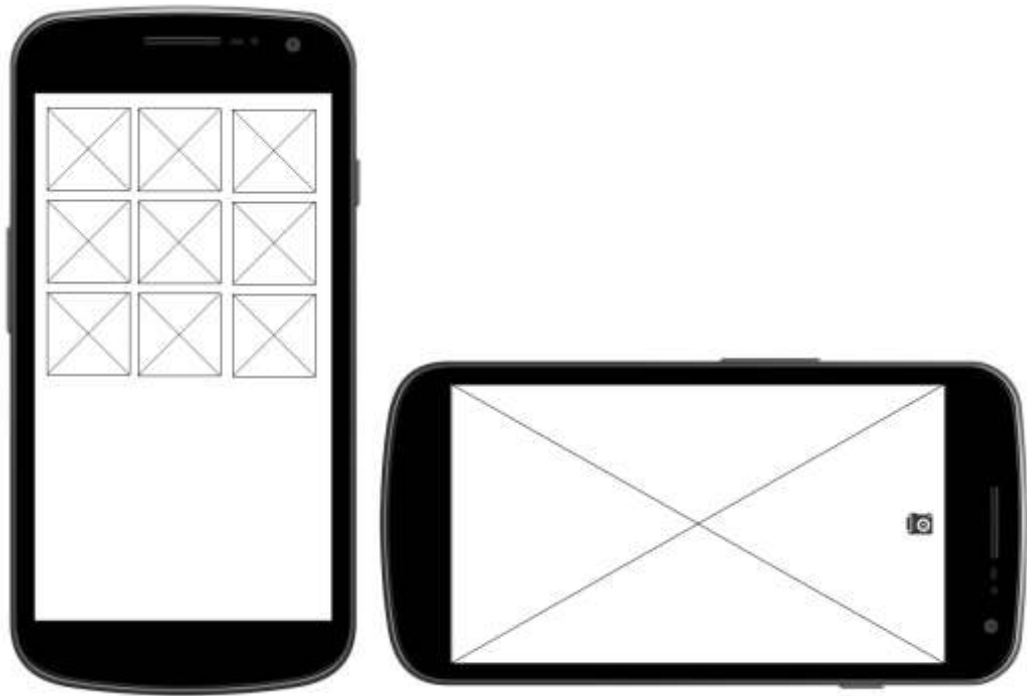
marcadores aparecerá la imagen en pequeño encima del mismo. Si pulsamos en esta imagen iremos a ver la vista de detalle de la imagen donde veremos toda su información.

Si pulsamos en el botón de arriba a la derecha iremos a la vista del listado (pantalla de la derecha) en el que se mostrarán las imágenes en forma de listado con las opciones de compartir y eliminar imagen.

Para añadir una imagen desde el mapa se pulsara el icono de la cámara de fotos mientras que para añadirla en la pantalla de listado se pulsara el “+”. Al pulsar este botón se mostrará la siguiente pantalla:



En esta pantalla seleccionaremos la forma de añadir la imagen, desde la cámara o desde la galería de imágenes. Las siguientes pantallas muestran un ejemplo de cómo se verán estas vistas (cámara y galería)



Al elegir la imagen a añadir se mostrara un formulario en el que se deberá introducir el nombre, comentario y lugar.



Una vez introducida esta información se verá la vista de detalle de la imagen tomada:



En esta pantalla podemos ver toda la información de una imagen y además las opciones de indicaciones y de compartir imagen que se apoyaran de aplicaciones instaladas en el teléfono para realizar estas tareas.



En la pantalla anterior podemos ver el inicio de sesión en el que se introducirá el email y la contraseña del usuario para poder exportar un mapa.



Una vez introducidos los datos del inicio de sesión correctamente se accederá al panel de exportación en el que se indicara si el mapa es privado, en caso afirmativo se introducirá una contraseña y se exportara el mapa. En la pantalla de la derecha podemos ver el proceso de exportación finalizado y como resultado la url del mapa generado.

En esta ultima pantalla podemos compartir la url del mapa utilizando las aplicaciones instaladas en el dispositivo móvil.

5.6.1.2 Aplicación web

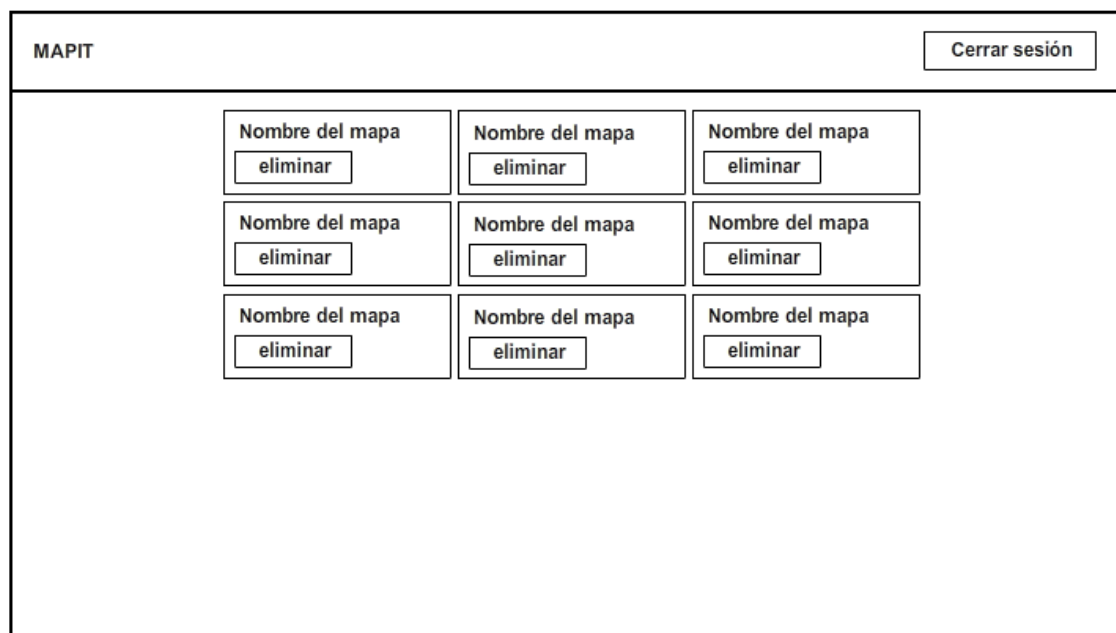
A continuación se describen las pantallas a desarrollar en la aplicación web del sistema.

The screenshot shows a web page titled "MAPIT" in the top left corner. In the top right corner, there is a button labeled "Iniciar sesión". The main content area is centered and contains a registration form titled "Registro de usuario". The form includes three input fields: "email", "contraseña", and "repite la contraseña". Below these fields is a button labeled "Registrar".

En esta primera pantalla se muestra el registro de usuario en el que se introducirán el email y la contraseña (repetiendo la contraseña por seguridad). Desde esta pantalla podemos acceder al inicio de sesión para entrar en el panel de un usuario.

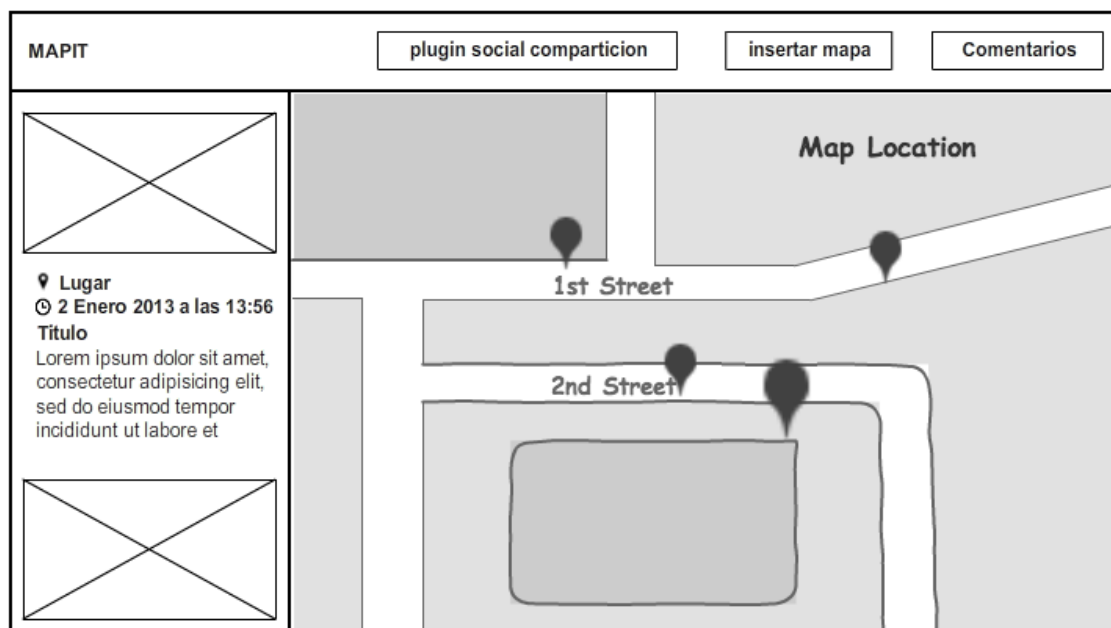
The screenshot shows a web page titled "MAPIT" in the top left corner. The main content area is centered and contains a login form titled "Inicio de sesión". The form includes two input fields: "email" and "contraseña". Below these fields is a button labeled "Acceder".

Esta pantalla muestra el inicio de sesión en la aplicación web, al introducir el usuario y la contraseña se accede al panel de usuario privado del usuario.



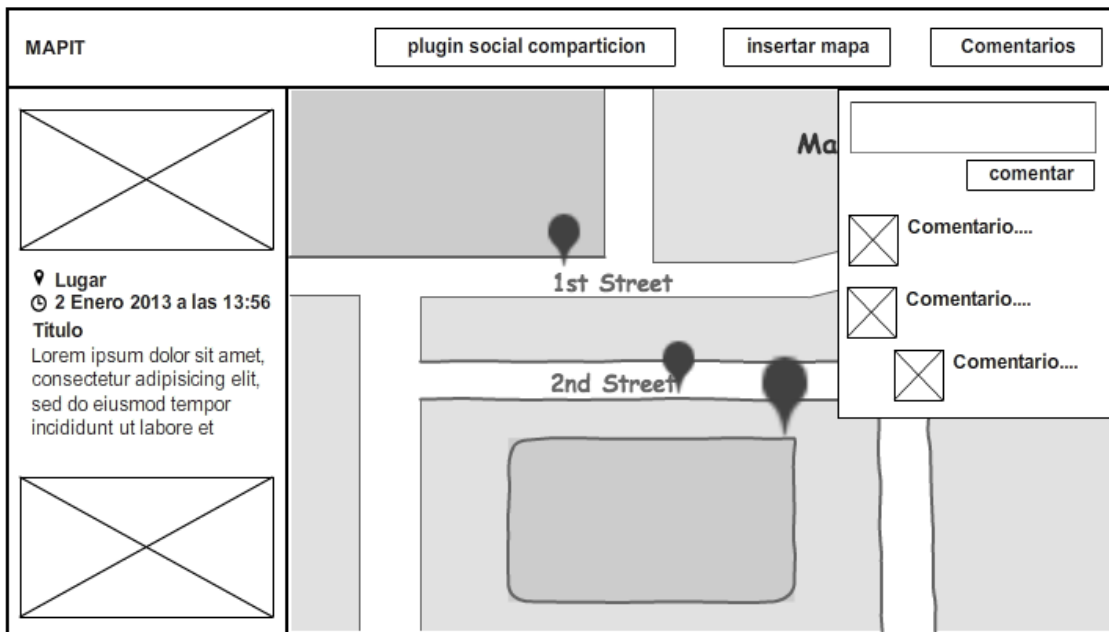
La pantalla anterior muestra el panel privado de un usuario donde se listan todos sus mapas y tiene la opción de verlos o eliminarlos. Además se puede cerrar sesión pulsando el botón situado en la esquina superior derecha.

A continuación se puede ver la pantalla del mapa generado por la aplicación.



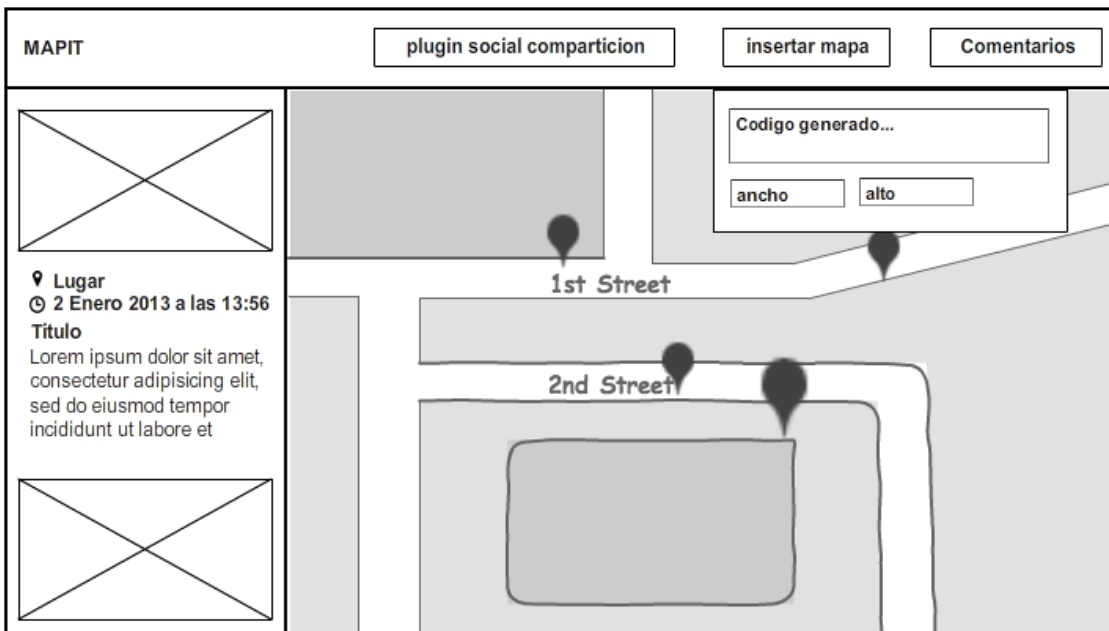
Esta pantalla muestra en una barra lateral izquierda las imágenes y toda su información relacionada. Haciendo scroll en esta pantalla irán apareciendo el resto de imágenes y el mapa se centrará automáticamente en el marcador que señale la localización de la imagen de la izquierda que se esté mostrando en pantalla en ese momento.

En esta pantalla podemos ver el plugin social para compartir el mapa, el generador de código para insertar el mapa en otra web y los comentarios del mapa.

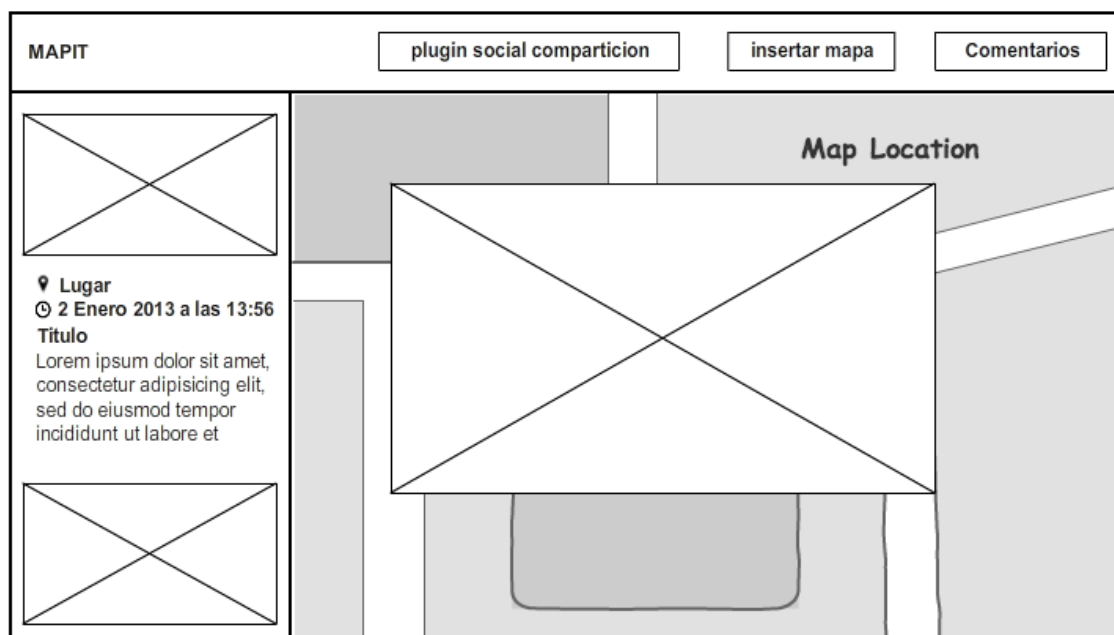


En la pantalla anterior se muestra la caja de comentarios desplegada, esta caja se desplegará al pulsar sobre el botón comentarios. En la parte superior se podrá añadir un comentario mientras que en la parte inferior se irán añadiendo los comentarios de los usuarios.

A continuación podemos ver el generador de código desplegado que muestra el código generado en el campo superior y dos campos para introducir el ancho y alto que se necesite para el mapa.



Además si pulsamos sobre una de las imágenes de la izquierda esta se abrirá en un popup en el centro de la web como se ve en la siguiente imagen.



5.6.2 Descripción del Comportamiento de la Interfaz

Durante la interacción del usuario con la aplicación podrán surgir errores, en caso de que estos errores sucedan en la aplicación móvil de Android se mostrara un popup indicando el error ocurrido. Mientras que si estos ocurren en la aplicación web se añadirá un campo al lado del lugar en el que se ha producido el error.

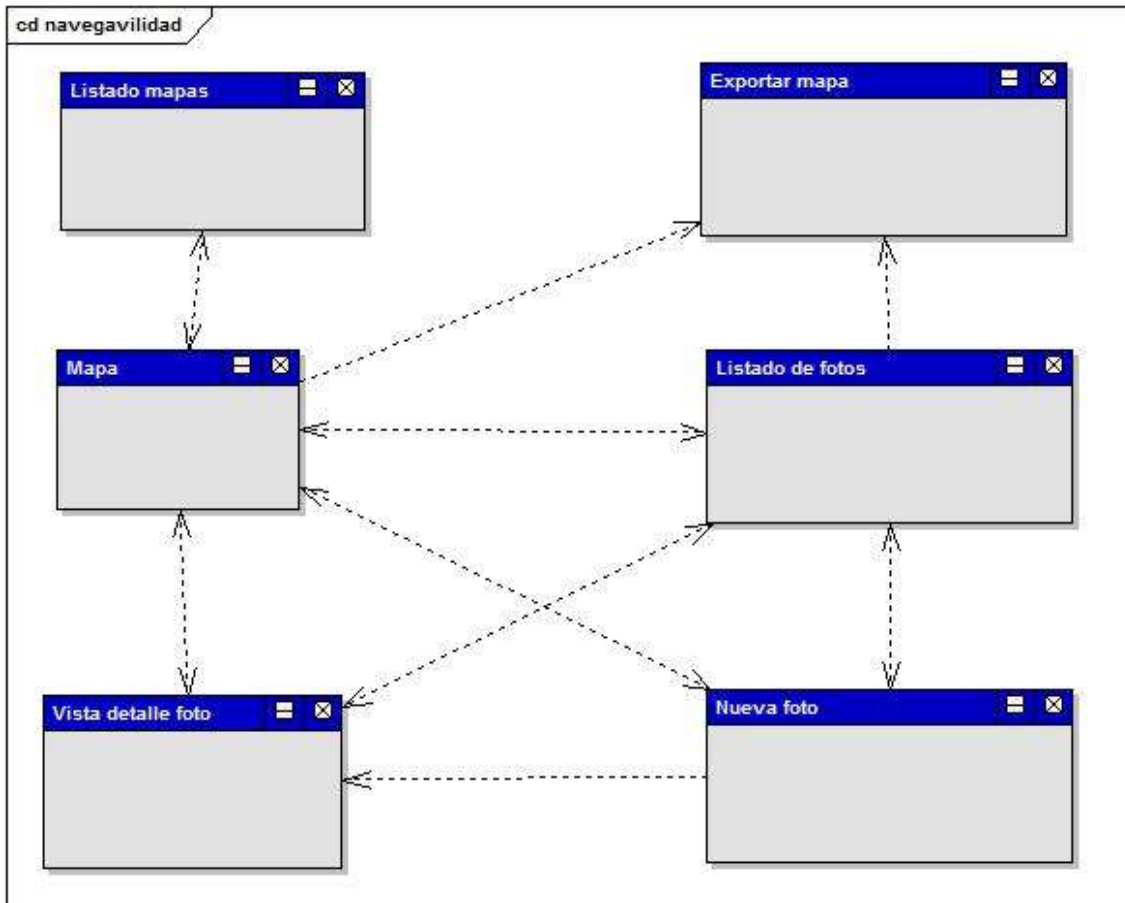
La validación de formularios incluirá mensajes de error por cada campo mostrando un error descriptivo de porque ha fallado esta validación y no borrando el resto de datos que se han introducido correctamente, haciendo que el usuario no tenga que meter estos datos otra vez.

En todo momento se informará al usuario de en que pantalla se encuentra en la parte superior de la aplicación móvil haciendo que así este nunca se encuentre perdido o no sepa que acción está realizando.

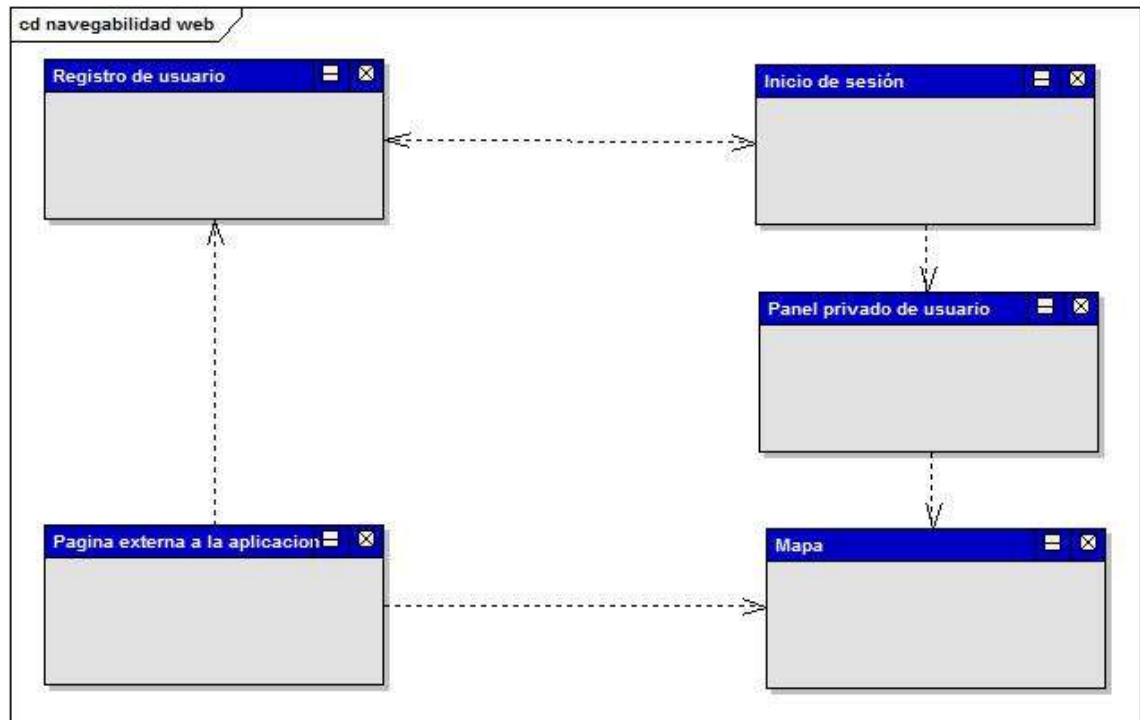
La aplicación móvil de Android utilizará los botones físicos de android (en algunos dispositivos) para la navegación entre pantallas (volver atrás)

5.6.3 Diagrama de Navegabilidad

5.6.3.1 Diagrama de navegabilidad de aplicación móvil



5.6.3.2 Diagrama de navegabilidad de aplicación web



5.6.4

5.7 Especificación del Plan de Pruebas

5.7.1 Pruebas Unitarias

<u>Registro de usuario</u>	
Prueba	Resultado Esperado
Añadir un usuario no existente	El usuario se añade correctamente
Prueba	Resultado Esperado
Añadir un usuario que ya existe	El usuario no se registra y se muestra un error.
Prueba	Resultado Esperado
El email no es valido	El usuario no se registra y se muestra un error sin borrar los campos introducidos por el usuario.
Prueba	Resultado Esperado
Las contraseñas no coinciden	El usuario no se registra y se muestra un error sin borrar los campos introducidos por el usuario.

<u>Inicio de sesión en aplicación web</u>	
Prueba	Resultado Esperado
Entrar con usuario existente	El usuario accede al panel de exportación de mapas
Prueba	Resultado Esperado
Entrar con usuario no existente	El usuario no accede al panel de exportación y se muestra un error.

<u>Ver mapa exportado en la aplicación web</u>	
Prueba	Resultado Esperado
Hacer scroll	Las imágenes irán apareciendo desde abajo y se seleccionará automáticamente la imagen superior en la pantalla. El mapa se moverá a la localización de esta imagen.
Prueba	Resultado Esperado
Pulsar en imagen	La imagen en la que se ha pulsado se abre en un popup en el centro de la pantalla.

<u>Eliminar mapa</u>	
Prueba	Resultado Esperado
Eliminar mapa y confirmar petición de eliminar	Se mostrará un mensaje de confirmación para eliminar, al confirmarlo el mapa se eliminará del sistema.
Prueba	Resultado Esperado
Eliminar mapa y cancelar	Se mostrará un mensaje de confirmación para eliminar, al cancelarlo el mapa no se eliminará.

<i>Inicio de sesión en aplicación móvil</i>	
Prueba	Resultado Esperado
Entrar con usuario existente	El usuario accede al panel de exportación de mapas
Prueba	Resultado Esperado
Entrar con usuario no existente	El usuario no accede al panel de exportación y se muestra un error.
<i>Crear nuevo mapa (aplicación móvil)</i>	
Prueba	Resultado Esperado
Se introduce el nombre del mapa y aceptar	Se crea el nuevo mapa con el nombre introducido
Prueba	Resultado Esperado
Se introduce el nombre del mapa y se pulsa cancelar	No se crea ningún mapa nuevo

<i>Eliminar mapa (aplicación móvil)</i>	
Prueba	Resultado Esperado
Eliminar mapa	Se eliminará el mapa de la aplicación móvil pero no de la aplicación web. Además se eliminarán las imágenes asignadas a este mapa.

<i>Ver mapa de fotos (aplicación móvil)</i>	
Prueba	Resultado Esperado
Ver mapa	Se indicarán todas las imágenes tomadas en este mapa en el lugar exacto.
Pulsar en marcado	Se verá la miniatura de la imagen tomada en el lugar del marcador indicado.
Pulsar en imagen del marcador	Se abrirá la vista de detalle de la imagen pulsada.

<i>Ver listado de fotos (aplicación móvil)</i>	
Prueba	Resultado Esperado
Ver listado	Se mostrarán todas las fotos añadidas al mapa y no se mostrarán las fotos que se hayan eliminado del mismo.

<i>Añadir foto desde la cámara (aplicación móvil)</i>	
Prueba	Resultado Esperado
Sacar foto y elegir	Se mostrará la cámara se sacara la foto y al aprobar la foto esta se añadirá al detalle de la vista de foto.
Sacar foto y descartar	Se mostrará la cámara se sacara la foto y al descartar la foto esta no se añadirá a la vista de detalle de foto permitiéndose volver a abrir la cámara para añadir la foto.
No se introduce nombre de foto	Se mostrará un mensaje indicando que este campo es obligatorio y no se almacenará la foto.
No se introduce lugar de foto	Se mostrará un mensaje indicando que este campo es obligatorio y no se almacenará la foto.
No se encuentra la localización el usuario	Se mostrará un mensaje indicando que el error y no se almacenará la foto.

<u>Añadir foto desde la galería de imágenes (aplicación móvil)</u>	
Prueba	Resultado Esperado
Elegir foto	Se mostrará la galería de imágenes se erigirá una foto y esta se añadirá al detalle de la vista de foto.
No se introduce nombre de foto	Se mostrará un mensaje indicando que este campo es obligatorio y no se almacenará la foto.
No se introduce lugar de foto	Se mostrará un mensaje indicando que este campo es obligatorio y no se almacenará la foto.
No se encuentra la localización el usuario	Se mostrará un mensaje indicando que el error y no se almacenará la foto.

<u>Eliminar foto (aplicación móvil)</u>	
Prueba	Resultado Esperado
Eliminar foto	Se eliminará la foto de la aplicación y por tanto no aparecerá en el mapa ni en el listado de fotos.

<u>Ver indicaciones al lugar en el que se sacó una foto (aplicación móvil)</u>	
Prueba	Resultado Esperado
No se encuentra ubicación	Se podrá elegir una ubicación desde la que empezar las ubicaciones
Ubicación encontrada	Se mostrarán las indicaciones desde la ubicación actual del usuario.

<u>Compartir foto (aplicación móvil)</u>	
Prueba	Resultado Esperado
Compartir foto	Se abrirá un selector de aplicaciones con las que compartir la foto y esta aparecerá en el servicio de compartición seleccionado.

<u>Exportar mapa (aplicación móvil)</u>	
Prueba	Resultado Esperado
Exportar por primera vez	Se enviarán todas las imágenes mostrando una barra de progreso y cuando el proceso acabe se mostrará la url del mapa al usuario
Exportar mapa privado	La url de retorno del mapa accederá a un mapa privado en el que habrá que introducir una contraseña
Exportar mapa publico	La url de retorno del mapa accederá a un mapa publico que podrá ser visto sin necesidad de introducir ninguna contraseña y sin estar registrado en el sistema

<u>Actualizar mapa (aplicación móvil)</u>	
Prueba	Resultado Esperado
Exportar por segunda vez el mismo mapa (actualizar mapa)	En lugar de crearse un nuevo mapa en el servidor se actualizará el mismo, eliminando las fotos que se hayan eliminado y añadiendo las nuevas en el orden correcto (según

	la fecha)
--	-----------

Compartir mapa exportado en las redes sociales (aplicación móvil)

Prueba	Resultado Esperado
Compartir	Se abrirá un selector de aplicaciones con las que compartir la url del mapa y esta aparecerá en el servicio de compartición seleccionado.

Comentar mapa exportado

Prueba	Resultado Esperado
Comentar	El comentario se añadirá al mapa en el que se ha comentado y únicamente a este mapa.
Eliminar comentario	El comentario se eliminará del mapa.

Generar código para insertar mapa en otra web

Prueba	Resultado Esperado
Cambiar ancho	En el momento en el que se cambia el ancho del mapa se cambia el código generado
Cambiar alto	En el momento en el que se cambia el ancho del mapa se cambia el código generado
Ajustar al ancho	En el momento en el que se cambia el ancho del mapa se cambia el código generado

Descargar todas las fotos de un mapa

Prueba	Resultado Esperado
Descargar zip	Se descargará un único zip con todas y cada una de las fotos del mapa

5.7.2 Pruebas del sistema

Para realizar las pruebas del sistema se realizarán los siguientes pasos:

- Integración de todos los subsistemas
- Paso de pruebas unitarias de cada uno de los subsistemas
- Se realizarán pruebas de integración comprobando que los diferentes subsistemas se comunican correctamente entre ellos y que no ha dejado de funcionar alguna funcionalidad en especial.
- Pruebas de despliegue, se probará a desplegar el sistema en un entorno real para comprobar que se podría poner en producción en un servidor preparado para este fin.

5.7.3 Pruebas de usabilidad

Las pruebas de usabilidad comprobarán que la aplicación es intuitiva, fácil de manejar y por tanto usable para un usuario común.

Para realizar estas pruebas se realizara un documento de pruebas con acciones a realizar en el sistema y se escogerá un grupo de usuarios de distintos perfiles (edad, trabajo, sexo..) que deberán realizar estas acciones. Se observara la reacción de los usuarios a las situaciones que han de resolver teniendo en cuenta el tiempo en realizar la acción, las dudas que estos tengan, las preguntas que hagan...

Una vez recolectados estos datos se hará un baremo que estimara si la aplicación es usable o no.

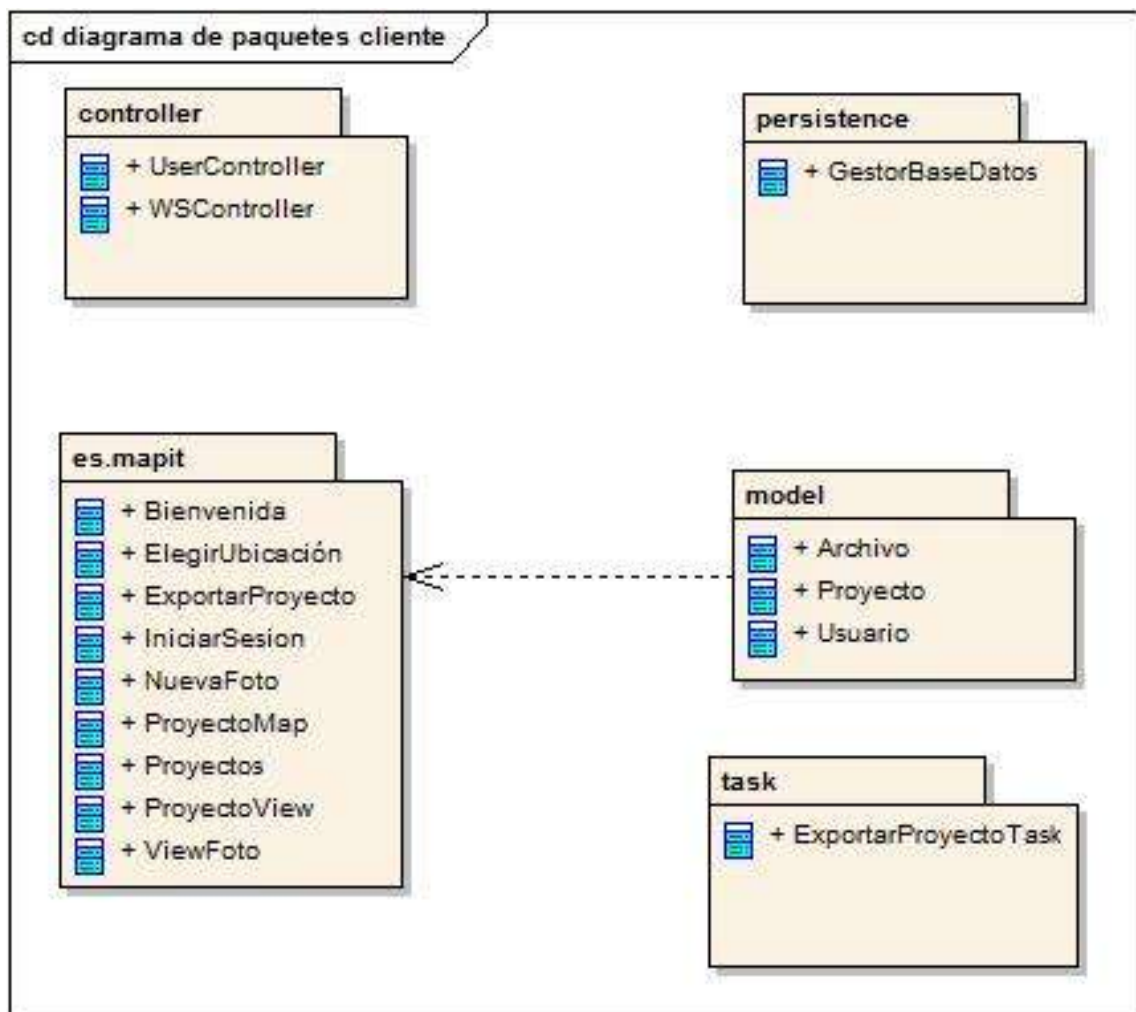
Además estas pruebas se realizarán en distintos dispositivos móviles, navegadores y sistemas operativos.

Capítulo 6. Diseño del Sistema

6.1 Arquitectura del Sistema

6.1.1 Diagramas de Paquetes

6.1.1.1 Diagrama de paquetes de aplicación móvil



6.1.1.1.1 Controller

Contiene las clases que funcionan como servicios para toda la aplicación. Estas clases son singletons, lo que permite tener una única instancia de ellas y poder ser llamadas desde cualquier otra clase almacenando los datos de forma estática.

6.1.1.1.2 Persistence

Este paquete se encarga de la gestión de la base de datos, cualquier salida o entrada de datos de la misma será a través de este paquete.

6.1.1.1.3 Model

Contiene las clases que representan el modelo de datos de la aplicación y que se utilizarán para encapsular los datos de la aplicación en objetos.

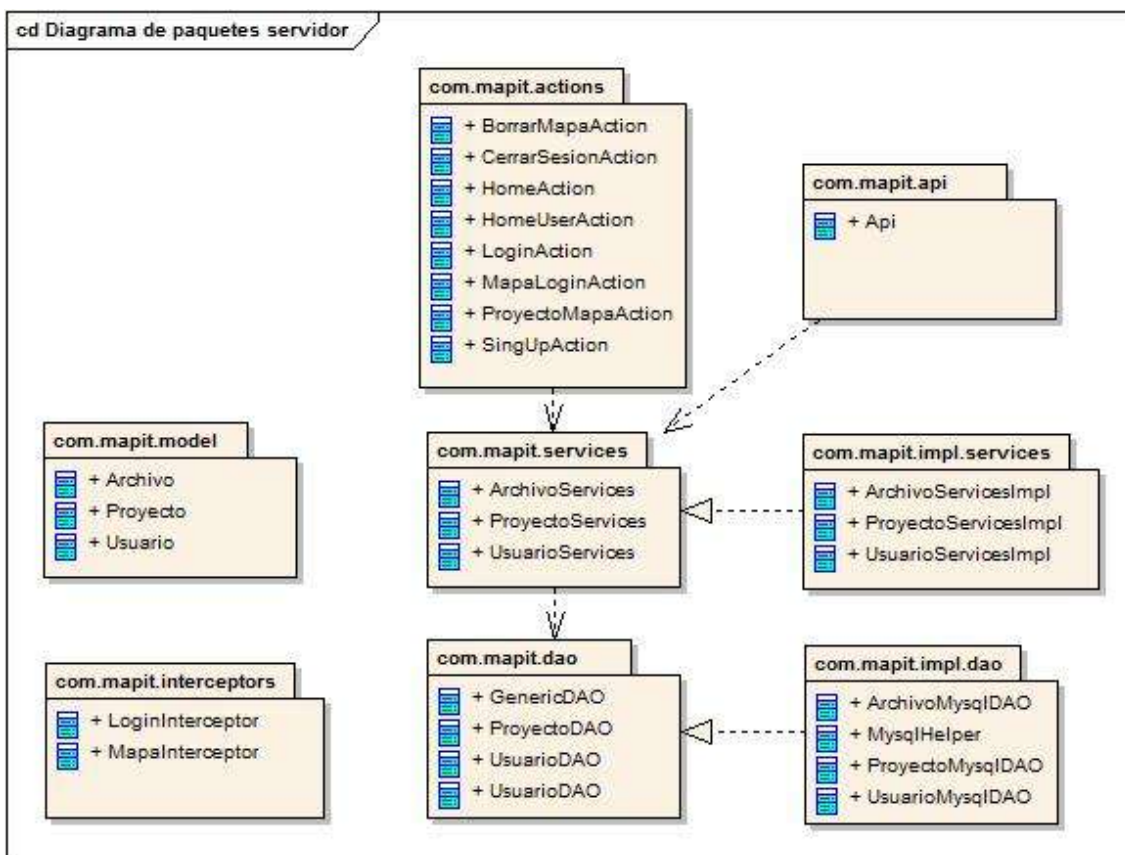
6.1.1.1.4 Task

Contiene todas las task utilizadas por la aplicación. Las task en Android representan hilos, es decir, tareas que se ejecutan en segundo plano dejando a la aplicación ejecutar otras tareas mientras se espera la respuesta del task ejecutado.

6.1.1.1.5 Es.mapit

Este paquete contiene los Actions de la aplicación y las clases que tratan la interfaz de usuario. Los actions son las clases que gestionan las vistas de la aplicación, y que hacen de controlador de estas.

6.1.1.2 Diagrama de paquetes de aplicación web



6.1.1.3 Com.mapit.actions

Este paquete contiene los actions de la aplicación, es decir, las clases que se ejecutan antes de mostrar la vista html de una página. En definitiva son los controladores de la aplicación web.

6.1.1.4 Com.mapit.model

El paquete model al igual que en la aplicación móvil contiene el modelo de datos del sistema.

6.1.1.5 Com.mapit.interceptors

Contiene los interceptores utilizados en la aplicación web. Los interceptores comprueban que se pueda acceder a una determinada acción que está protegida como por ejemplo los mapas privados, o el panel de control de un usuario, de esta forma los interceptores son los encargados de la seguridad del sistema.

6.1.1.6 Com.mapit.services

Este paquete contiene las interfaces de los servicios que deberán ser implementados por otras clases. Estos servicios son las funciones que se pueden realizar en la aplicación por tanto aquí se definirá la lógica de la aplicación que será implementada en el paquete com.mapit.impl.services.

6.1.1.7 Com.mapit.dao

El paquete dao contiene la descripción de las acciones que se pueden realizar en la base de datos. Contiene los Interfaces DAO que deberán ser implementados en el paquete com.mapit.impl.dao.

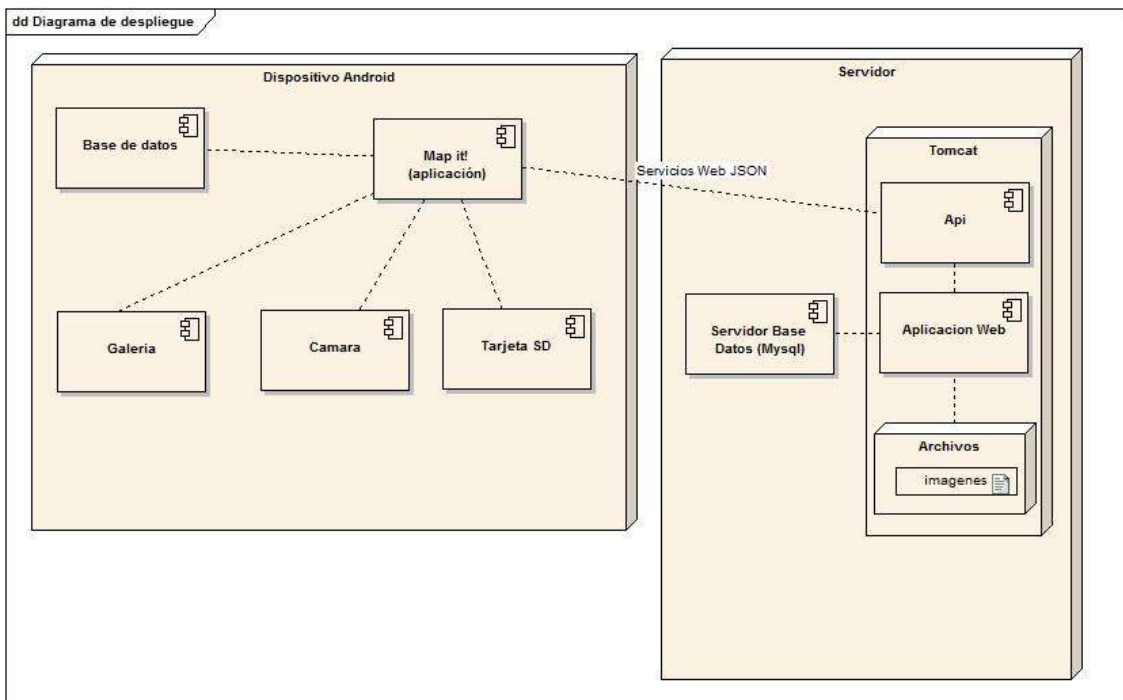
6.1.1.8 Com.mapit.impl.services

Este paquete contiene la implementación de los servicios de la aplicación, es decir, la lógica de la aplicación. Existirá una clase por cada entidad que pueda tener servicios (Archivo, Usuario, Proyecto)

6.1.1.9 Com.mapit.impl.dao

Contiene la implementación de los interfaces definidos en el paquete com.mapit.dao. Estas clases harán las consultas a la base de datos. En este caso los DAOs implementaran consultas a una base de datos mysql.

6.1.2 Diagramas de Despliegue



En el diagrama anterior se puede ver la distribución de los componentes en los distintos elementos de hardware necesarios para el despliegue. A continuación se describirán cada uno de estos componentes.

6.1.2.1 Map it! (aplicación)

La aplicación Android se instalará en el dispositivo móvil. Este componente representa a toda la aplicación en conjunto, que realizará las funcionalidades descritas en el análisis.

La conexión entre la aplicación y el servidor se realizará mediante Servicios Web REST en formato JSON. De esta forma la aplicación podrá enviar y recibir datos del servidor.

La aplicación móvil al instalarse crea una base de datos SQLite en el dispositivo que únicamente será accesible por esta aplicación. Además se comunicará con los componentes de galería de imágenes y cámara de fotos para obtener las fotos que el usuario desee agregar al mapa y con la tarjeta SD para almacenar las fotos antes de enviarlas al servidor al exportar un mapa.

6.1.2.2 Base de datos

Se creará una base de datos en el dispositivo móvil SQLite. En esta base de datos se almacenará toda la información de los mapas y archivos que el usuario agregue. Únicamente podrá ser accedida por la aplicación Map it! de forma que los datos se encontrarán seguros.

6.1.2.3 Galería

Este componente representa la galería de imágenes interna de Android. La galería proveerá a la aplicación móvil de las imágenes que el usuario seleccione de forma que pueda agregar estas fotos a sus mapas y exportarlas más tarde para crear un mapa.

6.1.2.4 Cámara

La cámara se comunicará con la aplicación para enviar la foto que el usuario haya sacado. Se abrirá como servicio asociado a la aplicación lo que hará las fotos creadas por la aplicación puedan ser gestionadas por esta y no por el sistema operativo.

6.1.2.5 Tarjeta SD

Este componente representa el almacenamiento de los archivos de la aplicación móvil. Al sacar una foto esta se almacenara en la tarjeta SD para que luego pueda ser accedida por la aplicación para ser exportada al servidor.

6.1.2.6 API

La api será el componente que se comunique con la aplicación móvil. Este componente representa los servicios web REST que reciben y envían información a la aplicación. Este componente será el único que podrá comunicarse con la aplicación de forma que el sistema sea más escalable y se puedan modificar tanto el cliente como el servidor sin necesidad de cambiar la comunicación entre ellos.

La API se comunicará con la aplicación web para solicitar y guardar los datos que recibe o envía.

La api será instalada en un servidor web Tomcat.

6.1.2.7 Aplicación Web

La aplicación web será se encarga de todas las funcionalidades que se pueden realizar desde el navegador. Ver el mapa, iniciar sesión, registrarse...

Este componente se encontrará instalado en un servidor web Tomcat al igual que la API. La aplicación web se desarrollara con Struts2 (Java, jsp).

6.1.2.8 Servidor de Base de datos (Mysql)

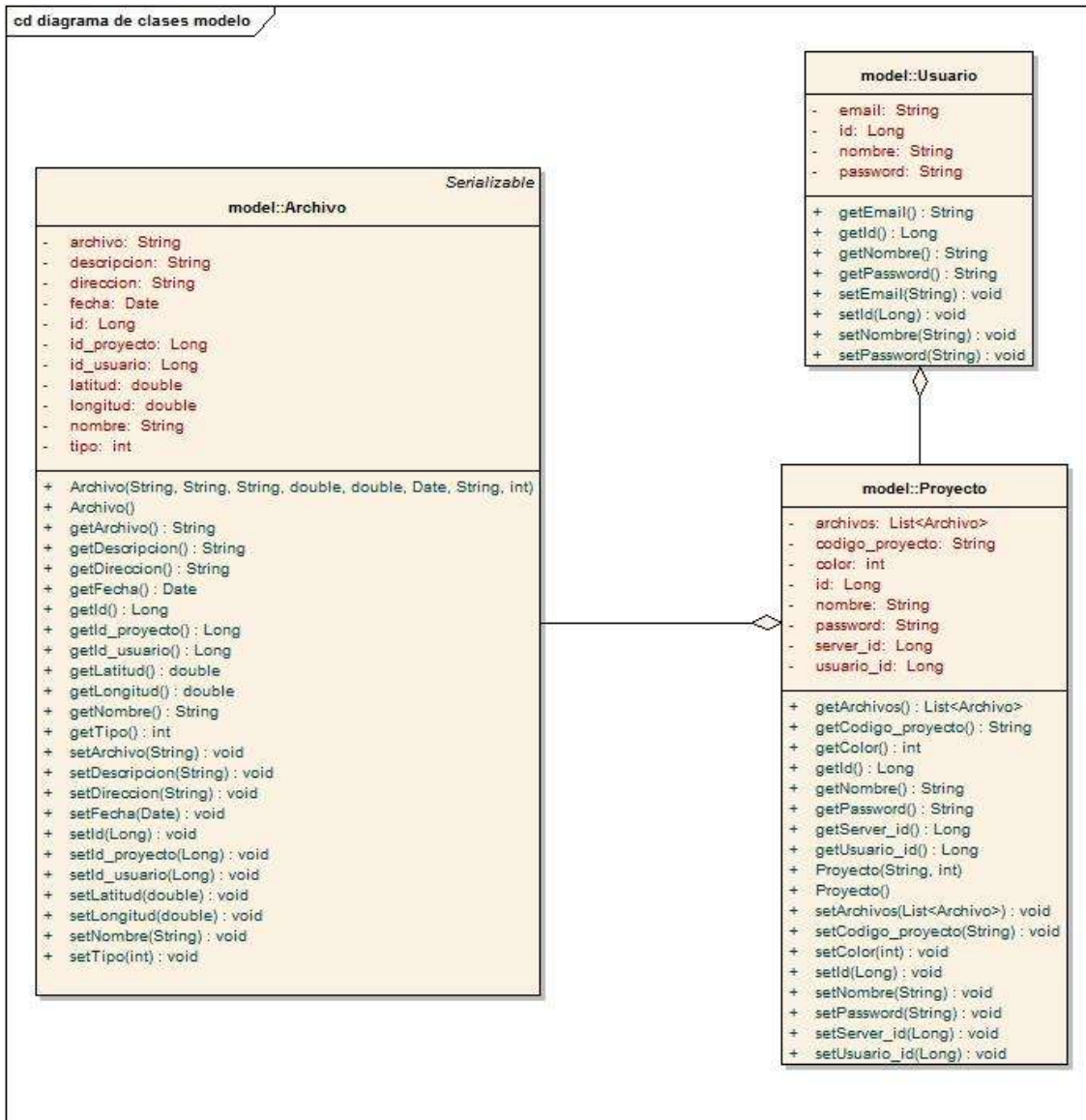
Este componente representa el servidor de base de datos. Se utilizará un servidor de base de datos Mysql instalado en el mismo servidor que el Tomcat que servirá la aplicación web y la API.

Será necesario crear una nueva base de datos en este servidor que pueda ser accedida por la aplicación web para almacenar los datos de la aplicación.

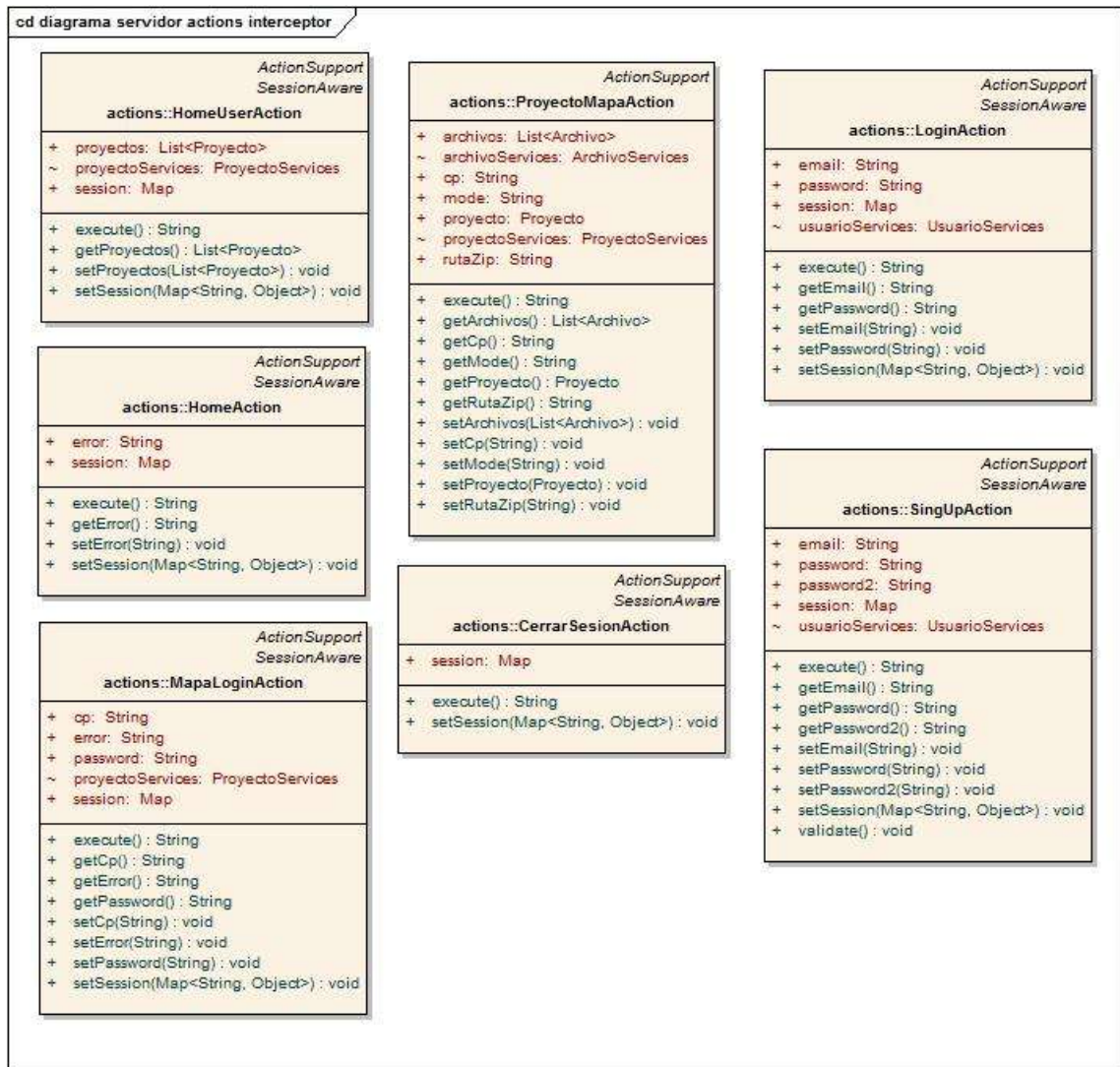
6.1.2.9 Archivos

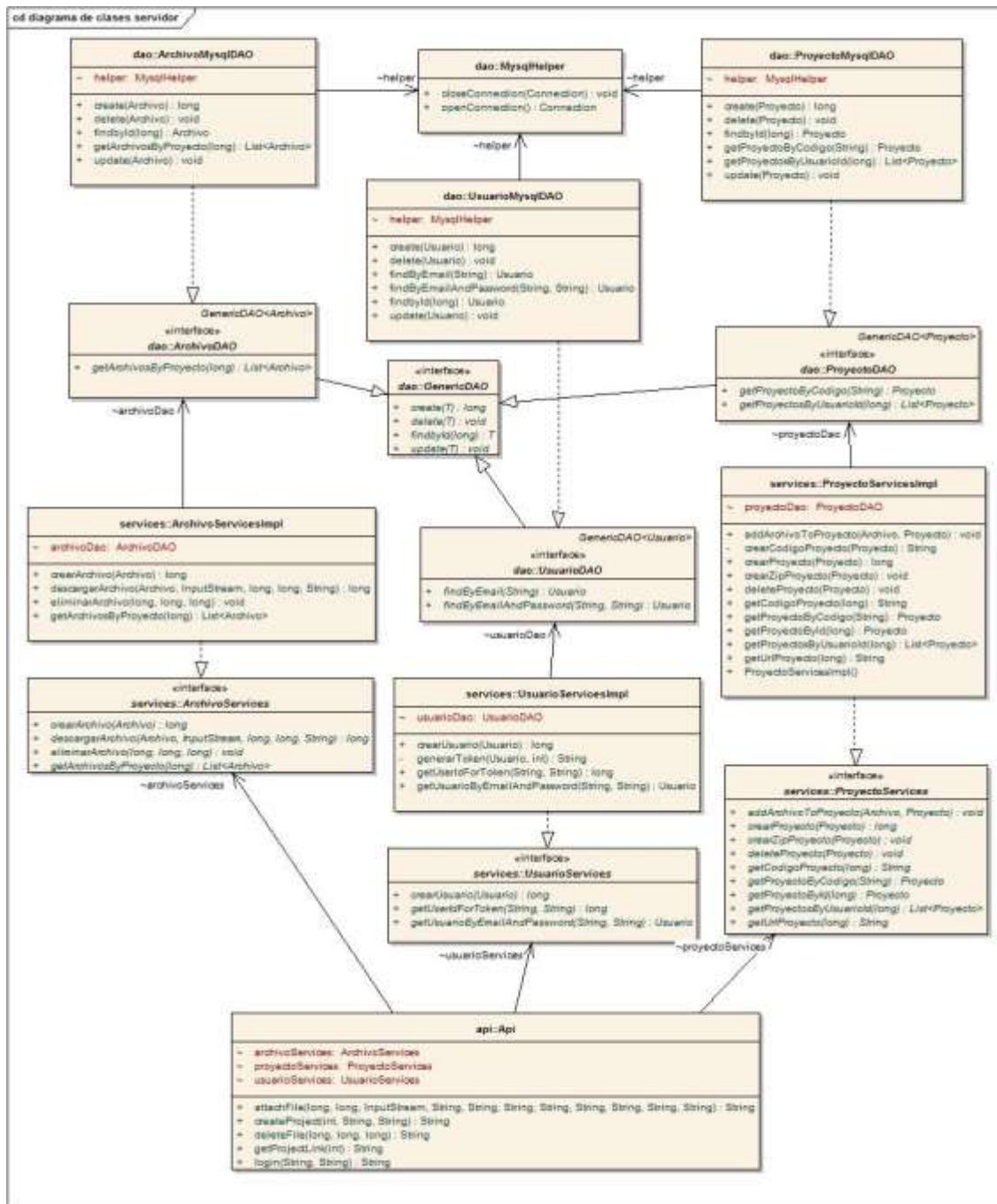
En el mismo servidor que el Tomcat y el servidor de bases de datos se encontrara la carpeta de archivos que almacenaran las fotos exportadas por los usuarios. Esta carpeta ha de ser accesible por el servidor web ya que se han de poder mostrar estas imágenes en el explorador de internet.

La carpeta se encontrará dentro del Tomcat lo que permitirá que estas imágenes sean visibles.



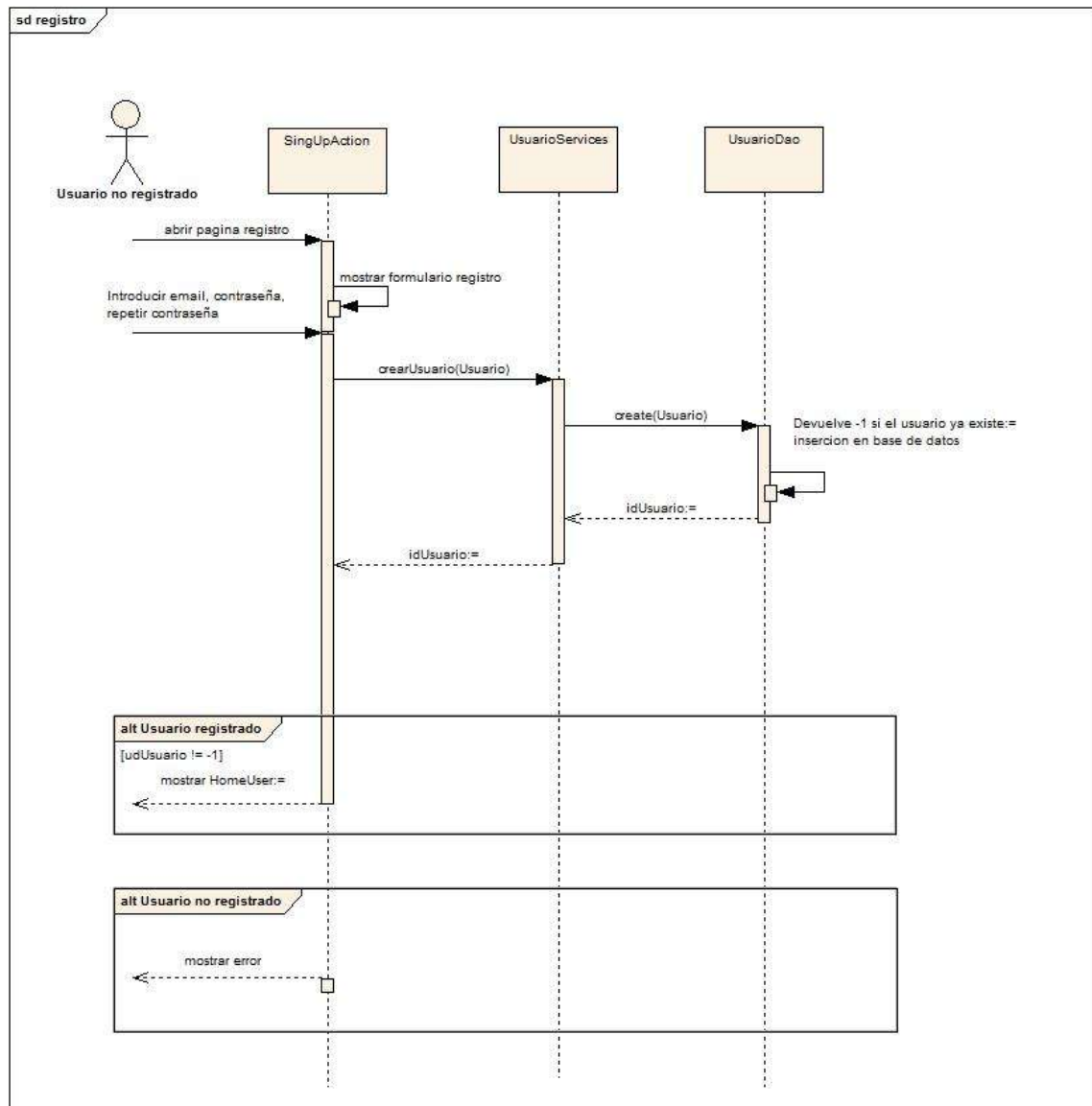
Los diagramas correspondientes al servidor se han separado en dos, uno muestra los actions de la aplicación web mientras que otro muestra la estructura de servicios y daos.



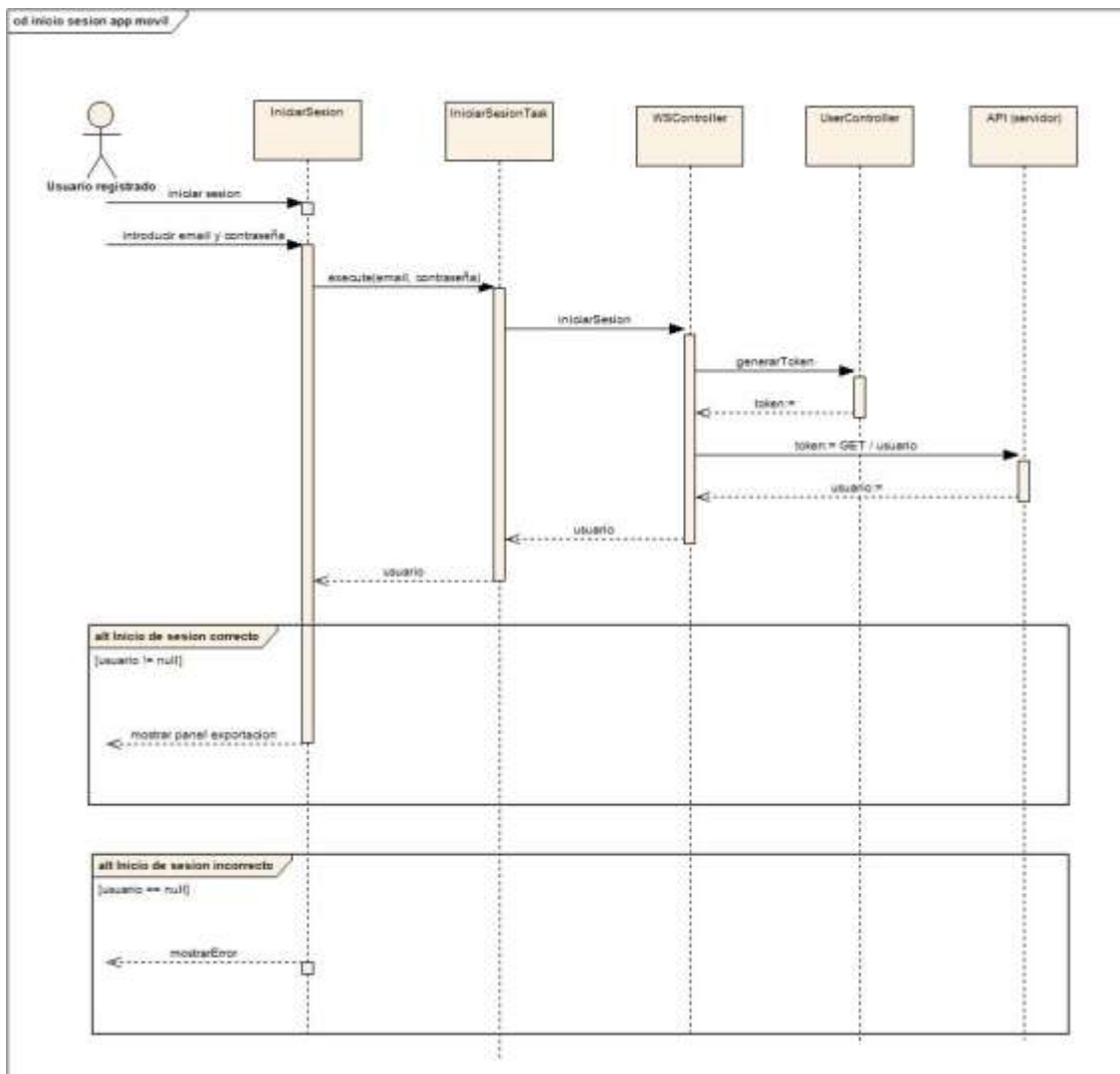


6.3 Diagramas de Interacción y Estados

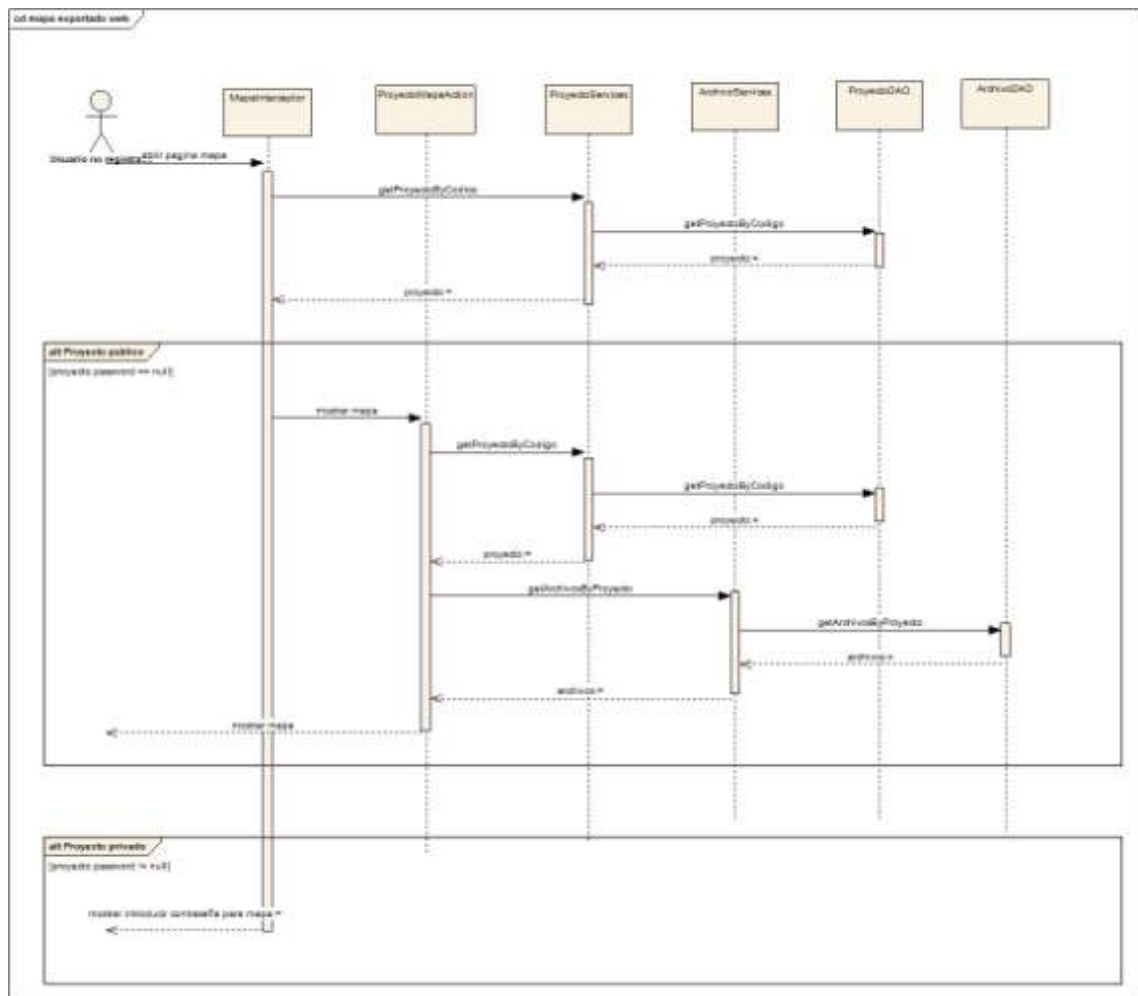
6.3.1 Registro de usuario



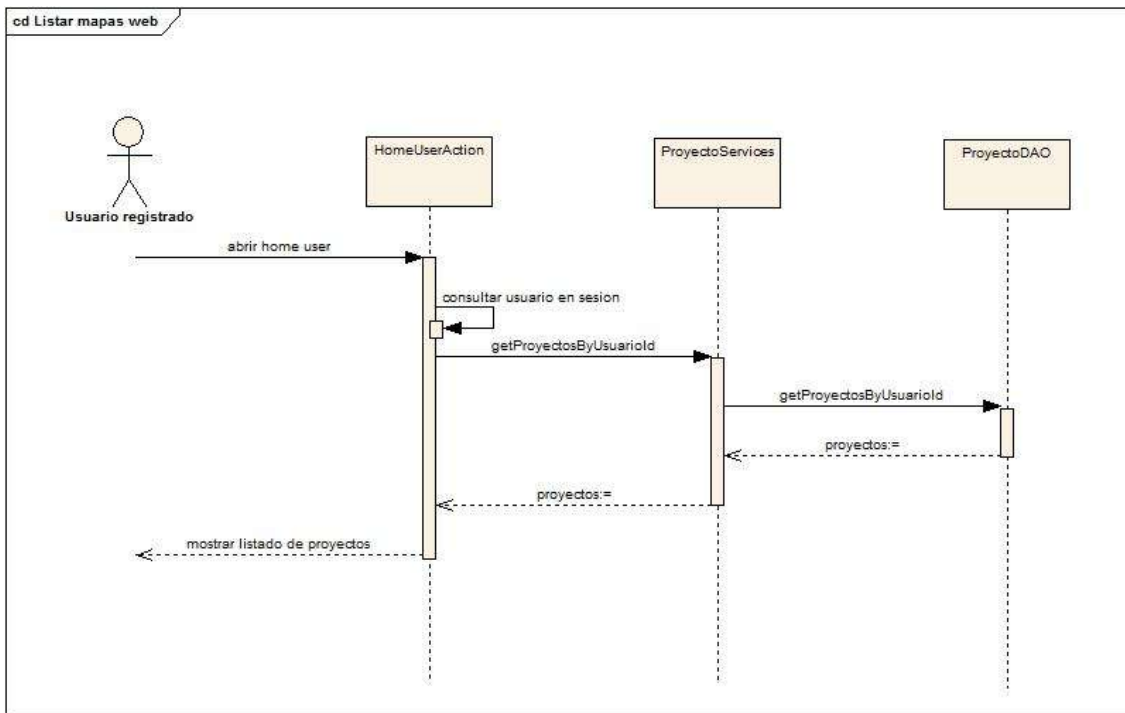
6.3.2 Inicio de sesión aplicación web



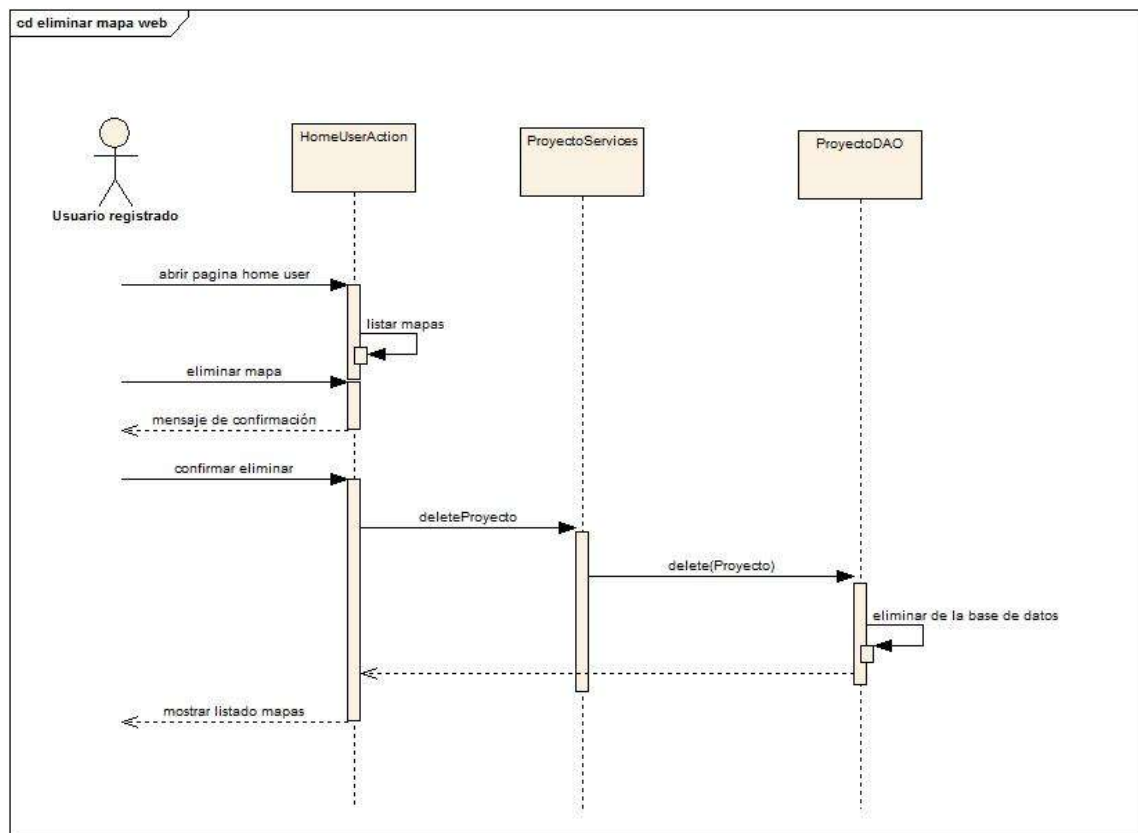
6.3.3 Ver mapa exportado en la aplicación web



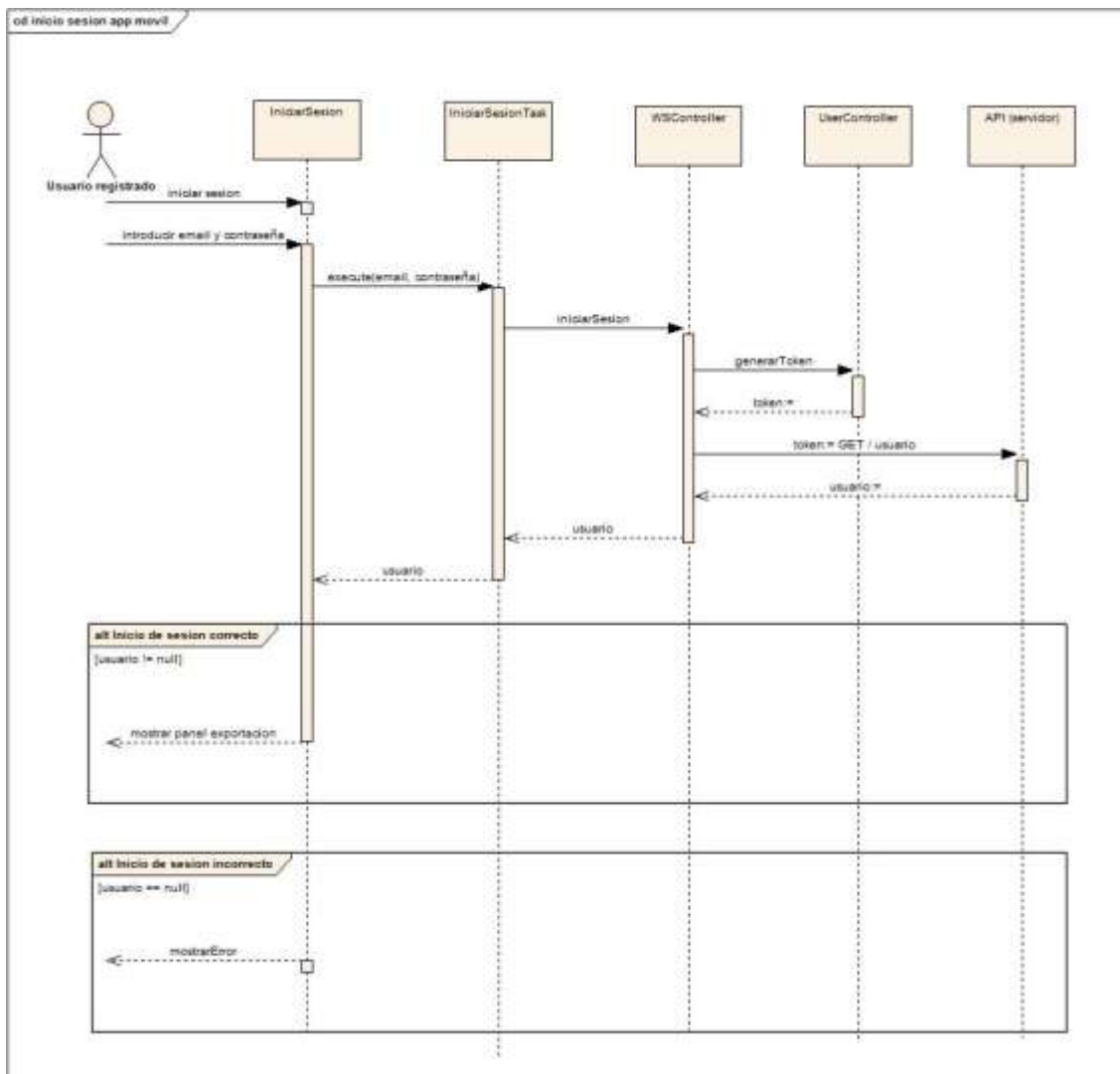
6.3.4 Listar mapas



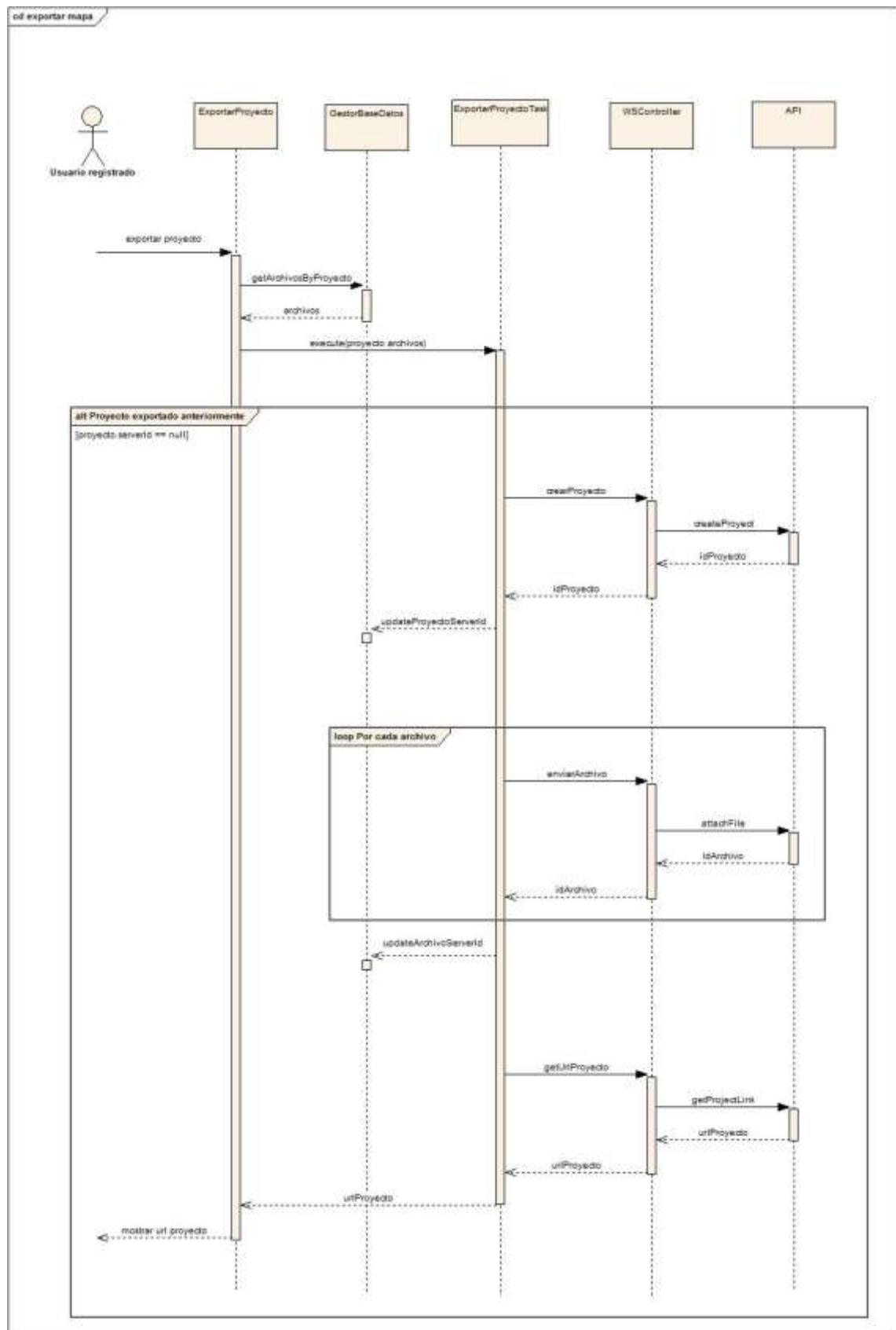
6.3.5 Eliminar Mapa



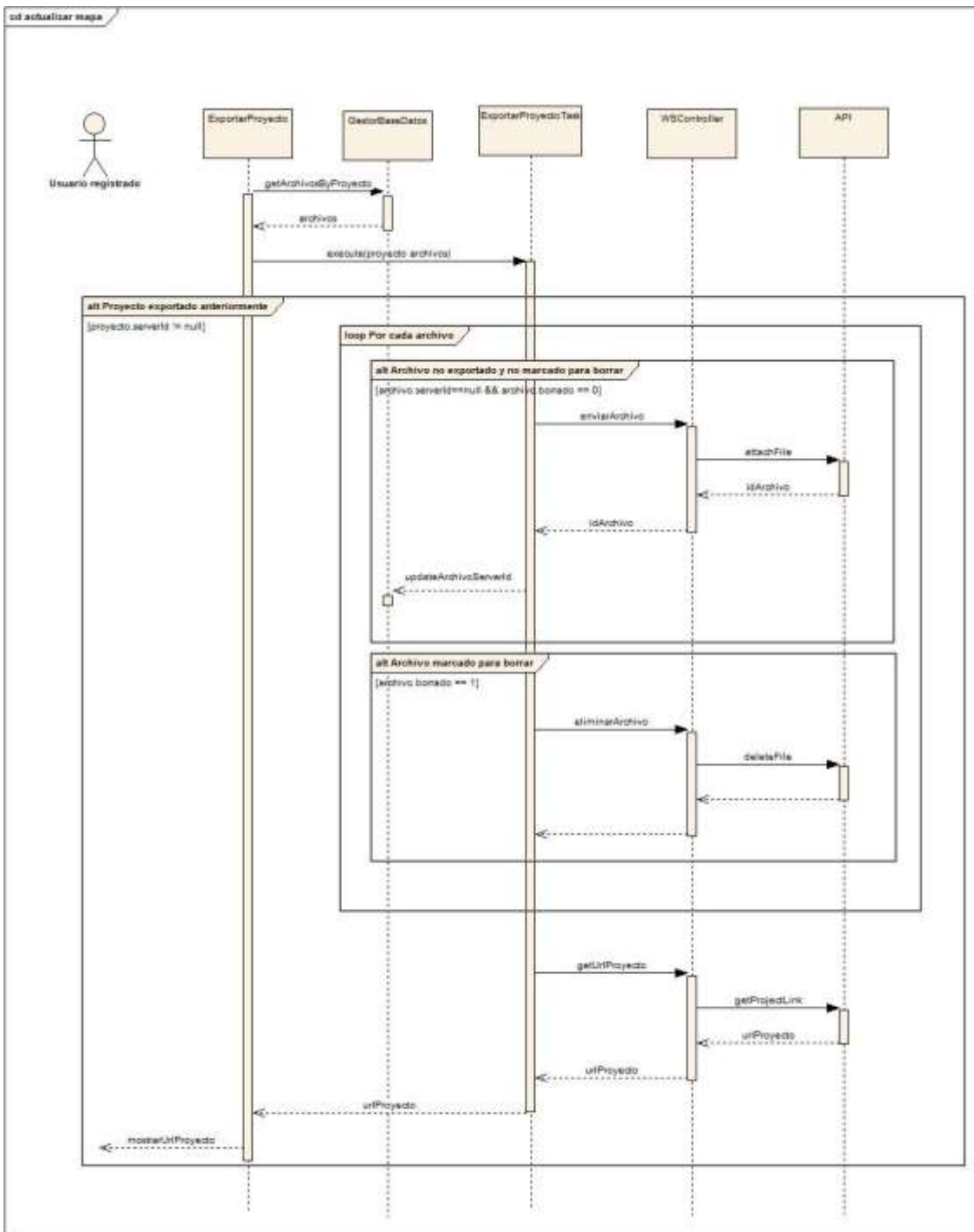
6.3.6 Inicio de sesión desde la aplicación móvil



6.3.7 Exportar mapa



6.3.8 Actualizar mapa



6.4

6.5 Diseño de la Base de Datos

6.5.1 Descripción del SGBD Usado

6.5.1.1 Aplicación móvil

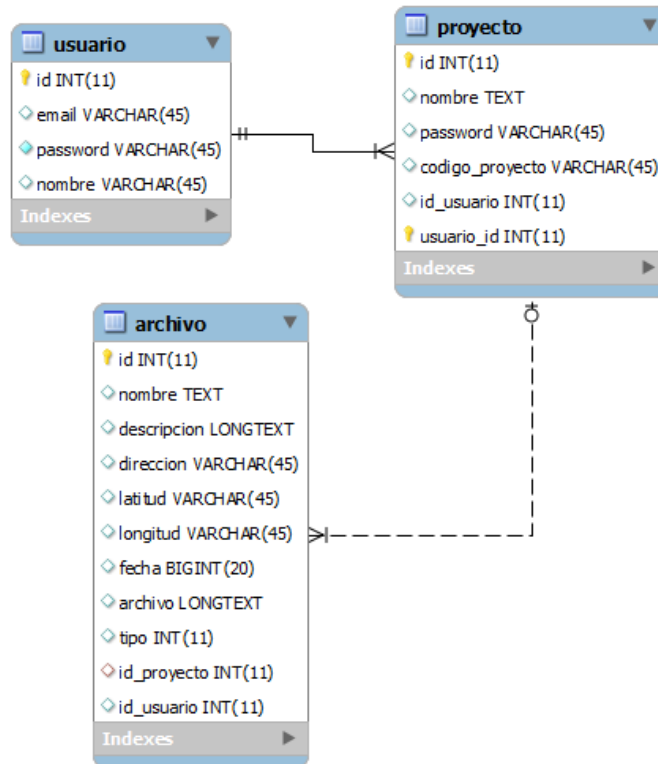
Android permite crear bases de datos SQLite desde una aplicación móvil en el propio dispositivo. Para la aplicación móvil se ha creado una base de datos en la propia memoria del dispositivo para almacenar localmente los datos de las fotos y mapas de los usuarios antes de ser exportados.

6.5.1.2 Servidor

En el servidor se ha instalado un servidor de base de datos Mysql y se ha creado una base de datos en este servidor que almacene la información de los mapas y fotos exportados por el usuario.

6.5.2 Diagrama E-R

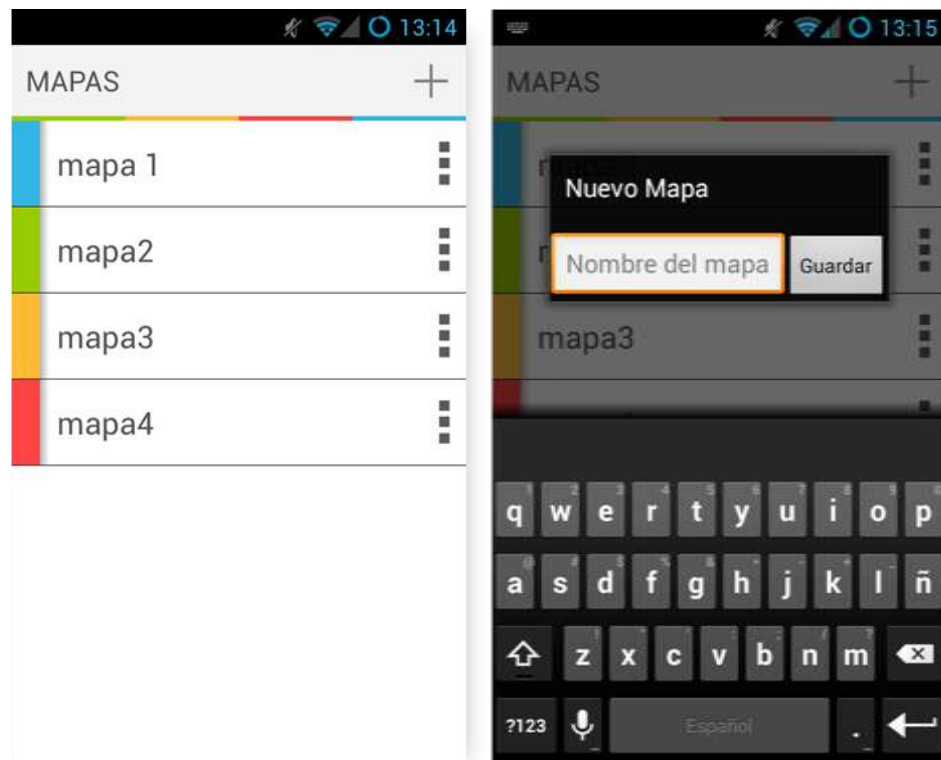
La base de datos del servidor y la base de datos de la aplicación comparten la misma estructura:



6.6 Diseño de la Interfaz

A continuación se muestra el diseño completo de la interfaz construido a partir de los diseños mostrados en el análisis.

6.6.1 Aplicación móvil

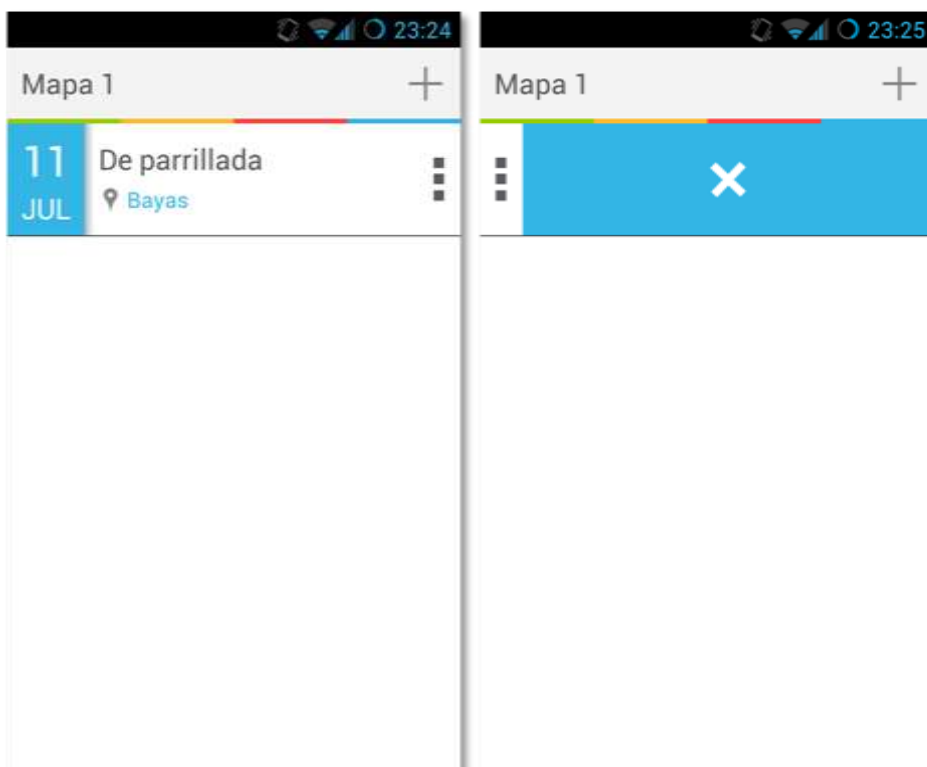


En las imágenes anteriores podemos ver la pantalla principal del listado de mapas y el popup para crear un mapa que aparece al pulsar la cruz de la esquina superior derecha. Además podemos ver que en listado se muestra un botón a la derecha de cada ítem que mostrará las opciones que se puedan realizar con este.



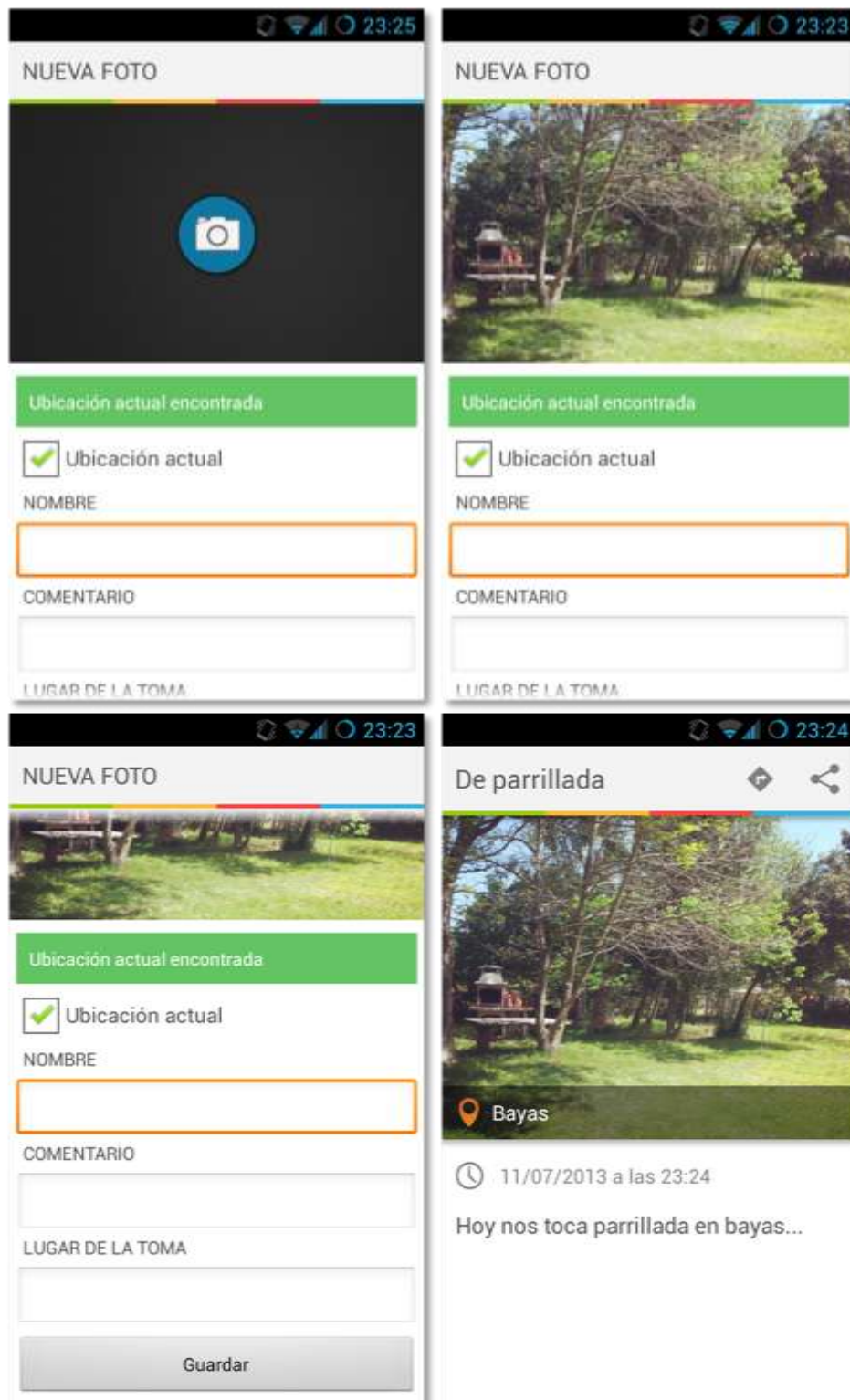
En la imagen anterior podemos ver el mapa de fotos del usuario. Al pulsar en un mapa esta es la vista que se mostrará. Las opciones de exportar y listar fotos del mapa se encuentran en la barra de acciones superior, junto con el nombre del mapa.

En el mapa podremos ver los marcadores de las imágenes que sacamos y un botón en la parte inferior de la pantalla que servirá para añadir una foto al mapa.



Las pantallas anteriores muestran el listado de imágenes de un mapa. Se puede ver de cada foto la fecha en la que fue tomada, el nombre de la foto y el lugar en el que fue sacada.

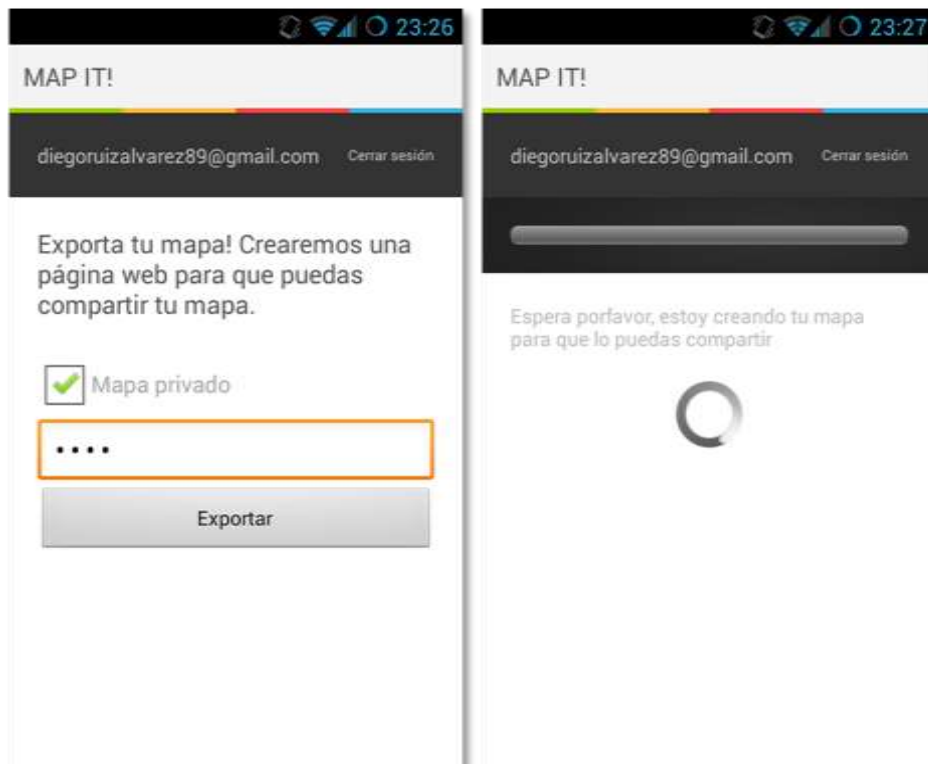
Además al igual que en la vista de mapas se tendrán opciones por cada foto, al pulsar este botón se mostrarán las opciones como se puede ver en la imagen de la derecha.



Las pantallas anteriores muestran el formulario de creación de una foto. Además de lo explicado en el análisis se ha incluido información sobre si se ha encontrado la localización del usuario.

En la última pantalla muestra el detalle de la imagen de un mapa, se puede ver toda la información de la misma y al pulsar sobre la imagen esta se abrirá en grande para poder verla con más detalle. Las acciones de compartir y mostrar indicaciones a la localización de la imagen se encuentran en la barra de acciones superior al igual que en toda la aplicación.

Por último se muestran las pantallas de exportación:



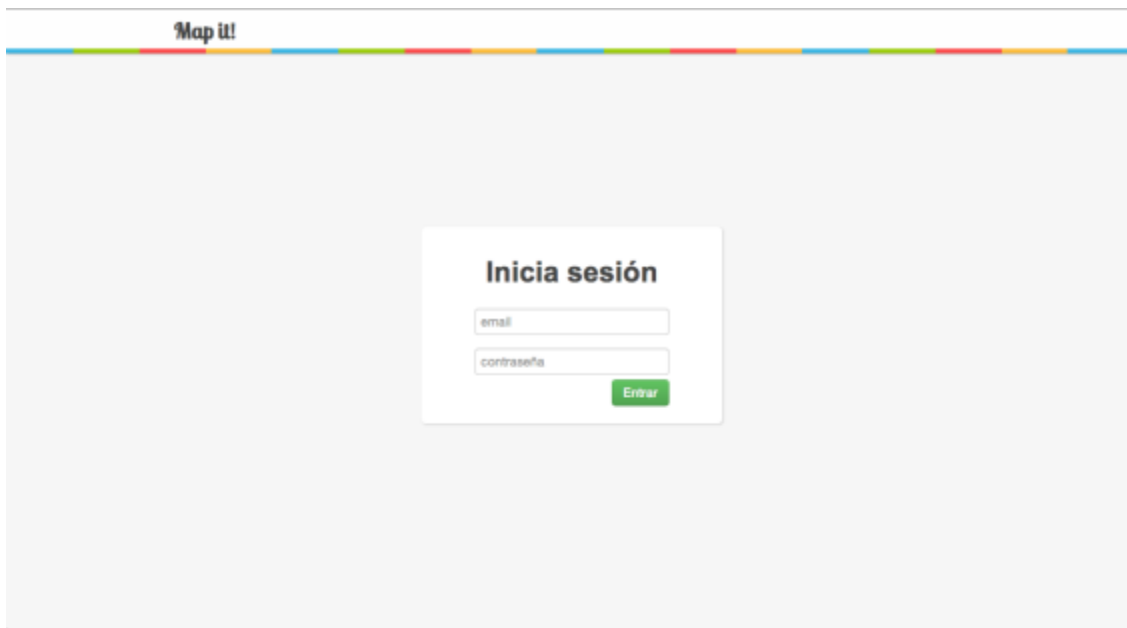


6.6.2 Aplicación web

En la primera pantalla podemos ver la home de la página en la que se muestra el registro directamente en ella para agilizar el proceso de registro de un usuario y un acceso al login.



La siguiente pantalla muestra el inicio de sesión.



El panel de control del usuario al iniciar sesión la primera vez estará vacío y se mostrará de la siguiente manera:



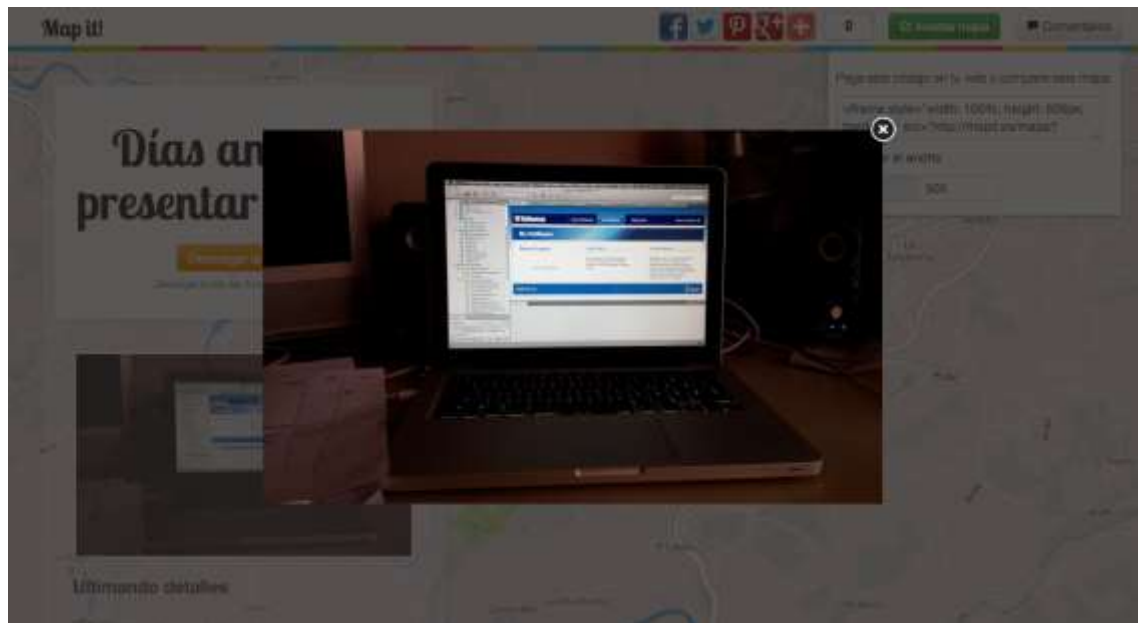
Cuando el usuario exporte mapas irán apareciendo en este panel:



Se puede ver que cada mapa tiene las opciones asociadas en su mismo recuadro. Al pulsar en ver mapa se abrirá la vista del mapa seleccionado:







6.7 Especificación Técnica del Plan de Pruebas

6.7.1 Pruebas Unitarias

<u>Registro de usuario</u>	
Estas pruebas se aplicarán al cambiar la base de datos, o cualquier cambio en el subsistema de seguridad	
Prueba	Resultado Esperado
Añadir un usuario no existente	Se añadirá un usuario que no exista al sistema, este usuario ha de insertarse en la base de datos y generar correctamente el hash para su contraseña añadiendo un usuario con un id distinto a todos los que se encuentran en la base de datos ya que este dato es autoincremental.
Prueba	Resultado Esperado
Añadir un usuario que ya existe	Al insertar en la base de datos un usuario ya existente, por la propia configuración de la base de datos, saltará una excepción indicando que el campo email es único y por lo tanto no se añadirá el usuario.
Prueba	Resultado Esperado
El email no es valido	Se validará antes de enviar los datos a la base de datos que el email introducido tiene formato de email valido, al añadir un email no valido estos datos no se enviarán a la base de datos y por tanto no se añadirá ningún usuario.
Prueba	Resultado Esperado
Las contraseñas no coinciden	Al igual que en la prueba anterior se validará este caso antes de enviarse a la base de datos por tanto al no coincidir no se creará ningún usuario en el sistema.

<u>Inicio de sesión en aplicación web</u>	
Estas pruebas se aplicarán al cambiar la base de datos, o cualquier cambio en el subsistema de seguridad	
Prueba	Resultado Esperado
Entrar con usuario existente	Al introducir un usuario existente se busca este usuario en la base de datos y se extraen sus datos, al estos no ser nulos se devuelve el usuario correctamente y por tanto se puede cargar su panel de control.
Prueba	Resultado Esperado
Entrar con usuario no existente	El usuario no se encuentra en la base de datos y por tanto no se puede acceder a su panel de control. Se vuelve a mostrar el login informando al usuario después de esta acción.

Ver mapa exportado en la aplicación web

Estas pruebas se aplicarán al cambiar en los css, js de la aplicación web. Se deberán realizar las pruebas en distintos navegadores para comprobar que se ve de la misma forma y se puede utilizar la aplicación web y por tanto ver el mapa.

Prueba	Resultado Esperado
Hacer scroll	Las imágenes irán apareciendo desde abajo y se seleccionará automáticamente la imagen superior en la pantalla. El mapa se moverá a la localización de esta imagen.
Prueba	Resultado Esperado
Pulsar en imagen	La imagen en la que se ha pulsado se abre en un popup en el centro de la pantalla.

Eliminar mapa

Estas pruebas se aplicarán al cambiar la base de datos, o cualquier cambio en el subsistema de seguridad

Prueba	Resultado Esperado
Eliminar mapa y confirmar petición de eliminar	Se mostrará un mensaje de confirmación para eliminar, al confirmarlo el mapa se eliminará el mapa seleccionado de la base de datos. Después de esta acción la base de datos contendrá un mapa menos y en el listado el mapa eliminado no aparecerá.
Prueba	Resultado Esperado
Eliminar mapa y cancelar	Se mostrará un mensaje de confirmación para eliminar, al cancelarlo el mapa no se eliminará ya que no se enviará la petición

Crear nuevo mapa (aplicación móvil)

Estas pruebas se aplicará al cambiar en la base de datos la tabla proyectos

Prueba	Resultado Esperado
Se introduce el nombre del mapa y aceptar	Se crea el nuevo mapa en la base de datos.
Prueba	Resultado Esperado
Se introduce el nombre del mapa y se pulsa cancelar	No se crea ningún mapa nuevo ya que no se llama siquiera al método de crear proyecto de GestorBaseDatos

Eliminar mapa (aplicación móvil)

Estas pruebas se aplicará al cambiar en la base de datos la tabla proyectos

Prueba	Resultado Esperado
Eliminar mapa	Se eliminará el mapa directamente de la base de datos de la aplicación móvil. Todos los archivos de este proyecto en la tabla archivo también se eliminarán de la base de datos.

Ver mapa de fotos (aplicación móvil)

Estas pruebas se aplicarán al cambiar en la base de datos o las Activities que gestionan la creación de fotos ya que de ellas puede depender que se vean o no las imágenes correctamente.

Prueba	Resultado Esperado
Ver mapa	Se extraerán de la base de datos todos los archivos relacionados con el proyecto. Después estos se mostrarán en

	el mapa.
Pulsar en marcado	Se verá la miniatura de la imagen tomada en el lugar del marcador indicado.
Pulsar en imagen del marcador	Se abrirá la vista de detalle de la imagen pulsada. Se extraerán todos los datos de este archivo de la base de datos correctamente.

Ver listado de fotos (aplicación móvil)

Estas pruebas se aplicará al cambiar en la base de datos o las Activities que gestionan la vista de las fotos del sistema

Prueba	Resultado Esperado
Ver listado	Se mostrarán todas las fotos añadidas al mapa y no se mostrarán las fotos que se hayan eliminado del mismo.

Añadir foto desde la cámara (aplicación móvil)

Estas pruebas se deberán aplicar en diferentes dispositivos ya que dependiendo del hardware dependerá el resultado de las mismas.

Prueba	Resultado Esperado
Sacar foto y elegir	Se mostrará la cámara se sacara la foto y al aprobar la foto el sistema devolverá a la aplicación una dirección en la que se almacenó esta imagen. Se guardará esta dirección junto con el resto de información en la base de datos y esta imagen se podrá acceder más tarde desde su vista de detalle, accediendo a la ruta proporcionada por el sistema de cámara.
Sacar foto y descartar	Se mostrará la cámara se sacara la foto y al descartar la foto esta no se añadirá a la vista de detalle de foto permitiéndose volver a abrir la cámara para añadir la foto. No se enviará nada a la base de datos después de esta tarea.
No se introduce nombre de foto	Se mostrará un mensaje indicando que este campo es obligatorio y no se almacenará la foto. No se enviará nada a la base de datos después de esta tarea.
No se introduce lugar de foto	Se mostrará un mensaje indicando que este campo es obligatorio y no se almacenará la foto. No se enviará nada a la base de datos después de esta tarea.
No se encuentra la localización el usuario	Se mostrará un mensaje indicando que el error y no se almacenará la foto. No se enviará nada a la base de datos después de esta tarea.

Añadir foto desde la galería de imágenes (aplicación móvil)

Estas pruebas se deberán aplicar en diferentes dispositivos ya que dependiendo del hardware dependerá el resultado de las mismas.

Prueba	Resultado Esperado
Elegir foto	Se mostrará la galería de imágenes se erigirá una foto , el

	<p>sistema devolverá a la aplicación una dirección en la que se almacenó esta imagen.</p> <p>Se guardará esta dirección junto con el resto de información en la base de datos y esta imagen se podrá acceder más tarde desde su vista de detalle, accediendo a la ruta proporcionada por el sistema de cámara.</p>
No se introduce nombre de foto	<p>Se mostrará un mensaje indicando que este campo es obligatorio y no se almacenará la foto.</p> <p>No se enviará nada a la base de datos después de esta tarea.</p>
No se introduce lugar de foto	<p>Se mostrará un mensaje indicando que este campo es obligatorio y no se almacenará la foto.</p> <p>No se enviará nada a la base de datos después de esta tarea.</p>
No se encuentra la localización el usuario	<p>Se mostrará un mensaje indicando que el error y no se almacenará la foto.</p> <p>No se enviará nada a la base de datos después de esta tarea.</p>

Eliminar foto (aplicación móvil)

Se aplicarán estas pruebas al modificar la base de datos o el sistema de exportación de modo que se compruebe que la imagen no sale ni en la aplicación móvil ni en el servidor una vez actualizado.

Prueba	Resultado Esperado
Eliminar foto	Se actualizara el archivo en la base de datos poniendo su atributo borrado a 1. De esta forma a la hora de exportar el mapa se podrá informar al servidor de que debe de eliminar esta foto en el servidor.

Ver indicaciones al lugar en el que se sacó una foto (aplicación móvil)

Estas pruebas se deberán aplicar en diferentes dispositivos y la configuración de estos dispositivos de distinta forma, es decir, apagando el gps, localizando con redes wifi...

Prueba	Resultado Esperado
No se encuentra ubicación	Se podrá elegir una ubicación desde la que empezar las ubicaciones
Ubicación encontrada	Se mostrarán las indicaciones desde la ubicación actual del usuario.

Compartir foto (aplicación móvil)

Estas pruebas se deberán aplicar en diferentes dispositivos

Prueba	Resultado Esperado
Compartir foto	Se abrirá un selector de aplicaciones con las que compartir la foto y esta aparecerá en el servicio de compartición seleccionado.

Exportar mapa (aplicación móvil)

Estas pruebas se deberán aplicar cada vez que se modifique la base de datos y el Activity

ExportarMapa o el task relacionado con esta acción.	
Prueba	Resultado Esperado
Exportar por primera vez	Se enviará la petición para crear un proyecto en el servidor. Se enviarán uno por uno los archivos del proyecto. Se enviará la petición de url del proyecto que el servidor devolverá a la aplicación.
Exportar mapa privado	La url de retorno del mapa accederá a un mapa privado en el que habrá que introducir una contraseña
Exportar mapa publico	La url de retorno del mapa accederá a un mapa publico que podrá ser visto sin necesidad de introducir ninguna contraseña y sin estar registrado en el sistema

<u>Actualizar mapa (aplicación móvil)</u>	
Estas pruebas se deberán aplicar cada vez que se modifique la base de datos y el Activity ExportarMapa o el task relacionado con esta acción.	
Prueba	Resultado Esperado
Exportar por segunda vez el mismo mapa (actualizar mapa)	En la actualización de un mapa no se enviará la petición para crear un nuevo proyecto. Se utilizará el id que el servidor tiene asignado al proyecto anteriormente exportado para enviar o borrar los archivos del mapa. Al igual que en el caso anterior se devolver la url del mapa y este se habrá actualizado añadiendo las nuevas fotos en orden o borrando las que tenían marcadas el atributo borrado a 1.

<u>Compartir mapa exportado en las redes sociales (aplicación móvil)</u>	
Estas pruebas se deberán aplicar en diferentes dispositivos	
Prueba	Resultado Esperado
Compartir	Se abrirá un selector de aplicaciones con las que compartir la url del mapa y esta aparecerá en el servicio de compartición seleccionado.

<u>Comentar mapa exportado</u>	
Estas pruebas se deberán comprobar al redespargar la aplicación web y en varios navegadores. Al utilizar un plugin de Facebook estas pruebas dependen en su gran mayoría del correcto funcionamiento del mismo por parte de Facebook.	
Prueba	Resultado Esperado
Comentar	El comentario se añadirá al mapa en el que se ha comentado y únicamente a este mapa.
Eliminar comentario	El comentario se eliminará del mapa.

<u>Generar código para insertar mapa en otra web</u>	
Estas pruebas se deberán comprobar en varios navegadores, además se deberán comprobar	

al cambiar los js de la aplicación web.	
Prueba	Resultado Esperado
Cambiar ancho	En el momento en el que se cambia el ancho del mapa se cambia el código generado
Cambiar alto	En el momento en el que se cambia el ancho del mapa se cambia el código generado
Ajustar al ancho	En el momento en el que se cambia el ancho del mapa se cambia el código generado

<i>Descargar todas las fotos de un mapa</i>	
Estas pruebas se deberán comprobar al cambiar los servicios de ArchivoServices ya que de estos depende la generación del zip.	
Prueba	Resultado Esperado
Descargar zip	Se descargará un único zip con todas y cada una de las fotos del mapa

6.7.2 Pruebas de Integración y del Sistema

<i>Comunicación aplicación móvil - API</i>	
Estas pruebas se deberán comprobar al cambiar tanto la API del servidor como la parte de exportación de la aplicación móvil, comprobando que los datos que se envían entre el cliente y el servidor son correctos.	
Estas pruebas se deberán realizar antes de desplegar la aplicación en el servidor o de generar el apk en la aplicación móvil.	
Prueba	Resultado Esperado
Iniciar sesión	Después de crear un usuario se comprobará que se puede iniciar sesión desde la aplicación móvil y que se guarda correctamente la sesión en esta. Tanto el servidor como la aplicación móvil deberán crear el mismo token para que el inicio de sesión sea correcto.
Exportar mapa	Se exportará un mapa comprobando que este se añade correctamente en la base de datos del servidor. Además se comprobará que las imágenes se corresponden con el proyecto añadido en base de datos.
Borrar foto y actualizar mapa	Se eliminará una foto de la aplicación, se comprobará que se marca el atributo borrado a 1 y que no aparece en la aplicación. Después de esto se actualizará el mapa (volver a exportar) y se comprobará que en la base de datos del servidor se ha eliminado el archivo correspondiente a la foto y que no se muestra en el mapa.
Añadir foto y actualizar mapa	Se añadirá una foto desde la aplicación móvil, se exportará el proyecto y se comprobará que se ha añadido esta foto en la

	base de datos del servidor, además de en el nuevo zip descargable.
Descargar zip	Se comprobará que después de exportar un mapa y actualizar este en el servidor el zip generado se corresponda con el contenido de este mapa y no contenga archivos ya eliminados o carezca de nuevos archivos.

6.7.3 Pruebas de Usabilidad y Accesibilidad

6.7.3.1 Actividades de las Pruebas de Usabilidad

Como se ha comentado en el análisis las pruebas de usabilidad de la aplicación se realizarán con personas de diversos perfiles. Para ello se acogerán y asignaran personas a los siguientes grupos:

- Joven
- Adulto
- Anciano

A cada uno de estos grupos se les pasarán las pruebas de usabilidad explicadas a continuación de forma que después de la finalización de estas pruebas se pueda obtener un informe que muestre unos resultados medios de la usabilidad del sistema.

A continuación se detallan los cuestionarios que se utilizarán para las pruebas de usabilidad.

6.7.3.1.1 Preguntas de carácter general

¿Usa un ordenador frecuentemente?
<ol style="list-style-type: none"> 1. Todos los días 2. Varias veces a la semana 3. Ocasionalmente 4. Nunca o casi nunca
¿Qué tipo de actividades realiza con el ordenador?
<ol style="list-style-type: none"> 1. Es parte de mi trabajo o profesión 2. Lo uso básicamente para ocio 3. Solo empleo aplicaciones estilo Office 4. Únicamente leo el correo y navego ocasionalmente
¿Ha usado alguna vez software como el de esta prueba?

<ol style="list-style-type: none"> 1. Sí, he empleado software similar 2. No, aunque si empleo otros programas que me ayudan a realizar tareas similares 3. No, nunca
¿Qué busca Vd. Principalmente en un programa?
<ol style="list-style-type: none"> 1. Que sea fácil de usar 2. Que sea intuitivo 3. Que sea rápido 4. Que tenga todas las funciones necesarias

6.7.3.1.2 Actividades guiadas

Cada uno de los usuarios asignado a los grupos de pruebas deberá realizar las siguientes acciones, siendo estas vigiladas por el coordinador de las pruebas rellenando el siguiente formulario.

Tarea	Tiempo en realizar la tarea	Errores	Observaciones
<i>Crear mapa</i>			
<i>Añadir foto al mapa desde la galería</i>			
<i>Añadir foto al mapa desde la cámara</i>			
<i>Eliminar foto del mapa</i>			
<i>Ver listado de mapa</i>			
<i>Compartir imagen en las redes sociales</i>			
<i>Indicaciones a una imagen</i>			
<i>Exportar mapa</i>			
<i>Registro en la página web</i>			
<i>Iniciar sesión</i>			
<i>Ver mapa exportado</i>			
<i>Compartir mapa exportado</i>			
<i>Comentar mapa</i>			

6.7.3.1.3 Preguntas Cortas sobre la Aplicación y Observaciones

Un posible cuestionario de preguntas cortas (a desarrollar más en cada proyecto) es el siguiente:

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Sabe donde está dentro de la aplicación?</i>				
<i>¿Le resulta sencillo el uso de la aplicación?</i>				
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Funciona cada tarea como Vd. Espera?</i>				
<i>Echa de menos alguna funcionalidad</i>				
<i>Ve necesaria otra vista alternativa al mapa</i>				
<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>				
Calidad del Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
<i>El tipo y tamaño de letra es</i>				
<i>Los iconos e imágenes usados son</i>				
<i>Los colores empleados son</i>				
Diseño de la Interfaz	Si		No	A veces
<i>¿Le resulta fácil de usar?</i>				
<i>¿El diseño de las pantallas es claro y atractivo?</i>				
<i>¿Cree que el programa está bien estructurado?</i>				
Observaciones				
Cualquier comentario del usuario				

Los resultados de las pruebas se analizarán conjuntamente para y se realizará un informe que muestre:

1. **Facilidad de aprendizaje:** Capacidad para aprender la funcionalidad de la aplicación desarrollada y desarrollar las tareas de manera adecuada, midiendo cuanto se tarda en hacer las distintas tareas.
2. **Eficiencia:** Cuanto mejora la labor de los usuarios por usar la aplicación respecto a lo que se hacía anteriormente.

3. **Errores:** Cuantos errores cometen los usuarios en las distintas tareas, lo que decrementa la usabilidad del mismo.
4. **Satisfacción del usuario:** Impresión general de los usuarios al usar la aplicación.

6.7.3.2 Pruebas de Accesibilidad

En la realización de las pruebas de accesibilidad podremos determinar el nivel de accesibilidad de la aplicación web.

Para ello se realizarán las siguientes revisiones:

- Revisión preliminar: en esta revisión se comprobarán manualmente que la aplicación web funciona en diversos exploradores de internet, distintas resoluciones, distintos sistemas operativos....
- Checklist WCAG: se revisará el checklist de las WCAG 1 comprobando las reglas que se cumplen de accesibilidad y las que no.

Se intentará cumplir lo más posible los estándares de accesibilidad teniendo en cuenta que la página se ha creado con el fin de investigar nuevas formas de mostrar la información sobre un mapa.

6.7.4 Pruebas de Rendimiento

El sistema no depende de un alto rendimiento ya que la mayoría de tareas se realizan localmente en el dispositivo móvil, lo que relaja el trabajo del servidor.

A la hora de utilizar la aplicación web también se realiza la mayoría de operaciones en el propio cliente por medio de Javascript lo que hace que el servidor atienda únicamente la primera petición para devolver la información y el resto de interacción sea tratada en el cliente.

El rendimiento se medirá en la aplicación web del servidor y en la exportación desde el cliente móvil ya que son los puntos más delicados del sistema.

Para comprobar el rendimiento se medirán los siguientes aspectos

Acción	Tiempo máximo	Tiempo obtenido
<i>Cargar pagina home</i>	2-4 segundos	
<i>Cargar mapa</i>	4-7 segundos	
<i>Comentar mapa</i>	2 segundos	
<i>Exportar una única foto</i>	10 segundos (depende de la conexión)	
<i>Mostrar mapa aplicación móvil</i>	1 segundo	
<i>Añadir foto aplicación móvil</i>	1 segundo	

Capítulo 7. Implementación del Sistema

7.1 Lenguajes de Programación

7.1.1 Java

Tanto la aplicación móvil Android como la lógica de la aplicación web están desarrolladas en Java. La razón por la que se ha elegido este lenguaje de programación es la amplia experiencia con él y debido a que en el máster de ingeniería web se constituía como lenguaje de programación base.

El lenguaje de programación Java fue originalmente desarrollado por James Gosling de Sun Microsystems (la cual fue adquirida por la compañía Oracle) y publicado en el 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva mucho de C y C++, pero tiene menos facilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente. Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir del 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados.

7.1.1.1 Orientado a objetos

La primera característica, orientado a objetos ("OO"), se refiere a un método de programación y al diseño del lenguaje. Aunque hay muchas interpretaciones para OO, una primera idea es diseñar el software de forma que los distintos tipos de datos que usen estén unidos a sus operaciones. Así, los datos y el código (funciones o métodos) se combinan en entidades llamadas objetos. Un objeto puede verse como un paquete que contiene el "comportamiento" (el código) y el "estado" (datos). El principio es separar aquello que cambia de las cosas que permanecen inalterables. Frecuentemente, cambiar una estructura de datos implica un cambio en el código que opera sobre los mismos, o viceversa. Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de un sistema

software. El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando como consecuencia su calidad y reduciendo el número de proyectos fallidos. Otra de las grandes promesas de la programación orientada a objetos es la creación de entidades más genéricas (objetos) que permitan la reutilización del software entre proyectos, una de las premisas fundamentales de la Ingeniería del Software. Un objeto genérico “cliente”, por ejemplo, debería en teoría tener el mismo conjunto de comportamiento en diferentes proyectos, sobre todo cuando estos coinciden en cierta medida, algo que suele suceder en las grandes organizaciones. En este sentido, los objetos podrían verse como piezas reutilizables que pueden emplearse en múltiples proyectos distintos, posibilitando así a la industria del software a construir proyectos de envergadura empleando componentes ya existentes y de comprobada calidad; conduciendo esto finalmente a una reducción drástica del tiempo de desarrollo. Podemos usar como ejemplo de objeto el aluminio. Una vez definidos datos (peso, maleabilidad, etc.), y su “comportamiento” (soldar dos piezas, etc.), el objeto “aluminio” puede ser reutilizado en el campo de la construcción, del automóvil, de la aviación, etc.

La reutilización del software ha experimentado resultados dispares, encontrando dos dificultades principales: el diseño de objetos realmente genéricos es pobremente comprendido, y falta una metodología para la amplia comunicación de oportunidades de reutilización. Algunas comunidades de “código abierto” (open source) quieren ayudar en este problema dando medios a los desarrolladores para diseminar la información sobre el uso y versatilidad de objetos reutilizables y bibliotecas de objetos.

7.1.1.2 Independencia de la plataforma

La segunda característica, la independencia de la plataforma, significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Este es el significado de ser capaz de escribir un programa una vez y que pueda ejecutarse en cualquier dispositivo, tal como reza el axioma de Java, “write once, run anywhere”.

Para ello, se compila el código fuente escrito en lenguaje Java, para generar un código conocido como “bytecode” (específicamente Java bytecode)—instrucciones máquina simplificadas específicas de la plataforma Java. Esta pieza está “a medio camino” entre el código fuente y el código máquina que entiende el dispositivo destino. El bytecode es ejecutado entonces en la máquina virtual (JVM), un programa escrito en código nativo de la plataforma destino (que es el que entiende su hardware), que interpreta y ejecuta el código. Además, se suministran bibliotecas adicionales para acceder a las características de cada dispositivo (como los gráficos, ejecución mediante hebras o threads, la interfaz de red) de forma unificada. Se debe tener presente que, aunque hay una etapa explícita de compilación, el bytecode generado es interpretado o convertido a instrucciones máquina del código nativo por el compilador JIT (Just In Time).

7.1.1.3 Recolector de basura

En Java el problema fugas de memoria se evita en gran medida gracias a la recolección de basura (o automatic garbage collector). El programador determina cuándo se crean los objetos y el entorno en tiempo de ejecución de Java (Java runtime) es el responsable de gestionar el

ciclo de vida de los objetos. El programa, u otros objetos pueden tener localizado un objeto mediante una referencia a éste. Cuando no quedan referencias a un objeto, el recolector de basura de Java borra el objeto, liberando así la memoria que ocupaba previniendo posibles fugas (ejemplo: un objeto creado y únicamente usado dentro de un método sólo tiene entidad dentro de éste; al salir del método el objeto es eliminado). Aun así, es posible que se produzcan fugas de memoria si el código almacena referencias a objetos que ya no son necesarios—es decir, pueden aún ocurrir, pero en un nivel conceptual superior. En definitiva, el recolector de basura de Java permite una fácil creación y eliminación de objetos y mayor seguridad.

7.1.2 JavaScript

JavaScript ha sido utilizado en la aplicación web para la interacción del mapa y para los componentes dinámicos que existen en la página.

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,³ basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas⁴ aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Una cuarta edición está en desarrollo e incluirá nuevas características tales como paquetes, espacio de nombres y definición explícita de clases.

JavaScript fue desarrollado originalmente por Brendan Eich de Netscape con el nombre de Mocha, el cual fue renombrado posteriormente a LiveScript, para finalmente quedar como JavaScript. El cambio de nombre coincidió aproximadamente con el momento en que Netscape agregó soporte para la tecnología Java en su navegador web Netscape Navigator en la versión 2.002 en diciembre de 1995. La denominación produjo confusión, dando la impresión de que el lenguaje es una prolongación de Java, y se ha caracterizado por muchos

como una estrategia de mercadotecnia de Netscape para obtener prestigio e innovar en lo que eran los nuevos lenguajes de programación web.

7.2 Herramientas y Programas Usados para el Desarrollo

7.2.1 Eclipse

El IDE utilizado para desarrollar la aplicación de Android ha sido Eclipse Indigo. Se ha utilizado este IDE debido a que dispone de un plugin para el desarrollo de aplicaciones Android.

Eclipse es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent o Azureus.

Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project, cubriendo casi todas las áreas de Model Driven Engineering.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

Eclipse fue liberado originalmente bajo la Common Public License, pero después fue relicenciado bajo la Eclipse Public License. La Free Software Foundation ha dicho que ambas licencias son licencias de software libre, pero son incompatibles con Licencia pública general de GNU (GNU GPL).

7.2.2 Netbeans

Para desarrollar la aplicación web se ha utilizado el IDE de programación Netbeans. Se ha utilizado este IDE debido a su fácil integración con un Tomcat ya instalado en el ordenador y la facilidad y potencia de las herramientas de refactorización.

El IDE NetBeans es un entorno de desarrollo integrado - una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número

importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso.

El NetBeans IDE es un IDE de código abierto escrito completamente en Java usando la plataforma NetBeans. El NetBeans IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring.

NetBeans IDE 6.5, la cual fue publicada el 19 de noviembre de 2008, extiende las características existentes del Java EE (incluyendo Soporte a Persistencia, EJB 3 y JAX-WS). Adicionalmente, el NetBeans Enterprise Pack soporta el desarrollo de Aplicaciones empresariales con Java EE 5, incluyendo herramientas de desarrollo visuales de SOA, herramientas de esquemas XML, orientación a web servicios (for BPEL), y modelado UML. El NetBeans C/C++ Pack soporta proyectos de C/C++, mientras el PHP Pack, soporta PHP 5.

Modularidad. Todas las funciones del IDE son provistas por módulos. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permitiéndole al usuario comenzar a trabajar inmediatamente.

Sun Studio, Sun Java Studio Enterprise, y Sun Java Studio Creator de Sun Microsystems han sido todos basados en el IDE NetBeans.

Desde julio de 2006, NetBeans IDE es licenciado bajo la Common Development and Distribution License (CDDL), una licencia basada en la Mozilla Public License (MPL). En octubre de 2007, Sun anunció que NetBeans desde entonces se ofrecerá bajo licenciamiento dual de Licencia CDDL y la GPL versión 2.

7.2.3 Android Development Tools

Junto a Eclipse se utilizó el Android Development Tools (ADT). El ADT es un conjunto de herramientas que ofrece Android para el desarrollo de nuevas aplicaciones. Estas herramientas se pueden integrar con nuestro entorno de desarrollo.

Junto a Eclipse se utilizó el Android Development Tools (ADT). El ADT es un conjunto de herramientas que ofrece Android para el desarrollo de nuevas aplicaciones. Estas herramientas se pueden integrar con el entorno de desarrollo que estemos utilizando, en este caso Eclipse.

Las herramientas que se ofrecen en el ADT se clasifican en dos tipos:

- **Herramientas SDK (SDK Tools):** Estas herramientas no dependen de la plataforma de Android para la que se esté programando.

- **Herramientas de Plataforma (Platform Tools):** Estas herramientas están actualizadas para ser usadas con la última plataforma de Android.

En cuanto a la interfaz gráfica y a la mayoría de recursos y archivos de Android se programan mediante un XML.

Las herramientas de desarrollo Android (ADT) plugin para Eclipse añade extensiones de gran alcance para desarrollar para Android. Permite crear y depurar aplicaciones de Android más fácil y más rápido. Algunas de las ayudas que ofrecen estas herramientas son:

- Da acceso a otras herramientas de desarrollo de Android desde dentro del IDE de Eclipse. Por ejemplo, ADT le permite acceder a las muchas capacidades de la herramienta “DDMS”: realizar capturas de pantalla, gestionar el reenvío de puerto, establecer puntos de interrupción... etc.
- Se proporciona un Asistente para nuevo proyecto, que te ayuda a crear rápidamente y crear todos los archivos básicos que se necesiten para una aplicación Android nueva.
- Automatiza y simplifica el proceso de construcción de una aplicación Android.
- Proporciona un editor de código de Android que ayuda a escribir código XML válido para los archivos de Android.
- Permite exportar un proyecto en un APK (archivo instalable de la aplicación en un dispositivo con Android) firmado, que puede ser distribuido a los usuarios.
- Estas herramientas incorporan también un emulador de teléfono móvil. Es muy útil en el caso de que no se posea un dispositivo con Android en el que probar la aplicación. Al principio del desarrollo comencé utilizando este emulador pero debido a grandes complicaciones de memoria y otros aspectos de los que éste carece termine depurando la aplicación en un teléfono con Android, cosa que es mucho más rápido y cómodo.

7.2.4 Photoshop

El Photoshop se ha utilizado durante la fase de diseño de la aplicación para dar formato a todos los botones, etiquetas, barras y fondos de la aplicación. En definitiva la función de esta herramienta fue la de dar un aspecto más atractivo y funcional a la aplicación.

La primera versión de Photoshop fue lanzada por primera vez en 1990; sin embargo, su primera versión vio la luz en 1988, cuando un estudiante de nombre Thomas Karroll lo produjo para Macintosh. En principio, el objetivo del programa era poder desarrollar la escala de grises en imágenes en pantallas monocromáticas. Dado el éxito que comenzó a tener, se ampliaron sus herramientas hasta convertirse en unos de los programas más importantes y reconocidos en el mundo para el retoque y variación de fotos. Tal fue la acogida, que ahora se utiliza la palabra photoshop para cualquier tipo de modificación en una foto, y el acto de modificarla se denomina photoshopear en nuestro idioma.

Este programa es elaborado por la compañía Adobe System y puede ser usado tanto en Windows como en computadoras Apple. Su nombre en español significa “Taller de fotos” y su complejidad varía de acuerdo al uso que le quieras dar: si bien cuenta con funciones altamente

complejas, utilizadas por profesionales, no necesitas ser un experto para poder utilizarlo, pues con saber emplear herramientas básicas puedes modificar las fotos que desees.

7.3 Creación del Sistema

7.3.1 Problemas Encontrados

7.3.1.1 Gestión de memoria en Android

La gestión de la memoria RAM en los dispositivos móviles es un tema delicado, y más al desarrollar aplicaciones que gestionan elementos multimedia (imágenes, videos, audio). El procesamiento de estas imágenes hace que dispositivos con poca memoria RAM no puedan completar las tareas necesarias para su fin.

La gestión de la cámara, la galería y la tarjeta SD ha dado muchos problemas durante el desarrollo de la aplicación. Se ha conseguido bajar la calidad de las imágenes y decodificar las mismas a tamaños más pequeños para utilizarlas en la aplicación, además bajar el tamaño de estas imágenes ha sido una buena solución que mejora los tiempos de exportación de las mismas al tener que transferir menos datos.

7.3.1.2 Orientación de las imágenes en Android

Dependiendo del dispositivo utilizado la forma en la que estos tratan las imágenes y su orientación varia. En el caso de que se saque una foto con la pantalla en horizontal unos dispositivos giran la imagen después de sacarla para que esta quede bien orientada mientras que otros mantienen la imagen en vertical y dejan una marca en ella (Exif data) que indica que se ha de girar al mostrar.

Para hacer que la aplicación funcione correctamente en todos los dispositivos la solución fue crear un botón que permita al usuario girar la imagen en caso de que esta no esté en la orientación deseada, de este modo es el usuario el que puede dejar la imagen como desee sin depender del dispositivo.

7.3.1.3 Actualizaciones SDK Android y plugin Eclipse

No es la primera vez que tengo problemas con este tema. En muchas ocasiones actualizar el SDK de Android se puede convertir en un tormento. Al actualizar este el plugin de eclipse deja de funcionar, al actualizar el plugin de eclipse se dejan de reconocer las librerías utilizadas en el proyecto de Android...

Tras varios días buscando solución en foros y compartiendo información con otras personas en la misma situación se ha conseguido solucionar.

7.3.1.4 Google Maps v1 obsoleto

La aplicación de Android fue el primer subsistema en desarrollarse. Cuando este se desarrollo la api de Google Maps v2 no se encontraba todavía demasiado probada y no era muy recomendable desarrollar la aplicación con esta librería.

Se utilizo una librería externa para poder mostrar las fotos en el mapa de forma “atractiva” que funcionaba con la versión de Google Maps v1. Al querer firmar la aplicación meses después me encuentro con que la versión de Google Maps v1 está obsoleta y que se ha de usar la API de Google Maps v2. Esto no sería un gran cambio si no se hubiera desarrollado todo el sistema de mapas con la librería externa que gestiona los bodadillos ya que Google Maps v1 no permitía esto.

Al final no se tuvo que cambiar a la versión de Google Maps v2 ya que se tenía almacenada una clave para el mapa desde otro ordenador, pero en caso de querer firmar la aplicación para subirla a Google Play se deberá cambiar esta parte de la aplicación.

7.3.1.5 Facebook comment box

La caja de comentarios social de Facebook no funciona si la llamada no viene desde un dominio real, por lo que se tuvo que comparar un dominio y apuntar al servidor en el que desplegó la aplicación. Una vez que la aplicación estaba referenciada por el dominio la caja de comentarios funcionó perfectamente.

7.3.2 Descripción Detallada de las Clases

Capítulo 8. Desarrollo de las Pruebas

8.1 Pruebas Unitarias

<i>Registro de usuario</i>	
Prueba	Resultado Obtenido
Añadir un usuario no existente	Se añade el usuario al sistema correctamente
Prueba	Resultado Obtenido
Añadir un usuario que ya existe	Salta una excepción en la base de datos y se muestra una traza de error en el navegador Solución: Capturar la excepción y añadir un parámetro a la url para mostrar un mensaje de error en el formulario
Prueba	Resultado Obtenido
El email no es valido	Se muestra un error debajo del campo de email hasta que se introduzca bien el email.
Prueba	Resultado Obtenido
Las contraseñas no coinciden	Se muestra un error en el campo de la primera contraseña.

<i>Inicio de sesión en aplicación web</i>	
Prueba	Resultado Obtenido
Entrar con usuario existente	Se inicia sesión correctamente y se entra en el panel de control del usuario.
Prueba	Resultado Obtenido
Entrar con usuario no existente	No se inicia sesión y se vuelve a la página de inicio de sesión, pero no se muestra ningún error. Solución: Añadir un parámetro a la url para mostrar un mensaje de error en el formulario.

<i>Ver mapa exportado en la aplicación web</i>	
Prueba	Resultado Obtenido
Hacer scroll	Las imágenes aparecen correctamente y el mapa se mueve a la posición de las imágenes pero no tiene fluided. Solución: Revisión del JavaScript y el procesamiento de eventos para hacer una mejor gestión de los mismos y no sobrecargar el navegador con cálculos innecesarios.

Prueba	Resultado Obtenido
Pulsar en imagen	Se abre un popup en el centro de la imagen. El popup no tiene ancho máximo por lo que la imagen tiene scroll. Solución: Añadir ancho y alto máximos a la imagen y al popup que la contiene.

<u>Eliminar mapa</u>	
Prueba	Resultado Obtenido
Eliminar mapa y confirmar petición de eliminar	Se elimina el mapa correctamente.
Prueba	Resultado Obtenido
Eliminar mapa y cancelar	No se elimina el mapa y se cierra el popup.

<u>Crear nuevo mapa (aplicación móvil)</u>	
Prueba	Resultado Obtenido
Se introduce el nombre del mapa y aceptar	Se crea el mapa correctamente
Prueba	Resultado Obtenido
Se introduce el nombre del mapa y se pulsa cancelar	No se crea ningún mapa nuevo.

<u>Eliminar mapa (aplicación móvil)</u>	
Estas pruebas se aplicará al cambiar en la base de datos la tabla proyectos	
Prueba	Resultado Obtenido
Eliminar mapa	Se eliminará el mapa directamente de la base de datos de la aplicación móvil. Todos los archivos de este proyecto en la tabla archivo también se eliminarán de la base de datos.

<u>Ver mapa de fotos (aplicación móvil)</u>	
Prueba	Resultado Obtenido
Ver mapa	Se muestra el mapa correctamente pero no se centra en una posición buena para ver el contenido. Solución: Centrar el mapa en la última imagen que se añadió al mapa o en la ubicación del usuario. Se añadió un botón para ir a la ubicación del usuario.
Pulsar en marcador	Se abre correctamente la imagen encima del marcador con el nombre. La imagen tarda mucho en cargarse. Solución:

	Decodificar la imagen y reducir su tamaño ya que no es necesario el tamaño original para mostrarla tan pequeña.
Pulsar en imagen del marcador	Se abre la vista de detalle de la imagen.

Ver listado de fotos (aplicación móvil)

Prueba	Resultado Obtenido
Ver listado	Se muestran las imágenes del mapa pero no se encuentran ordenadas por fecha. Solución: Ordenar las imágenes por fecha.

Añadir foto desde la cámara (aplicación móvil)

Prueba	Resultado Obtenido
Sacar foto y elegir	Se muestra la foto correctamente pero si se saca en formato vertical en algunos dispositivos la imagen no se encuentra orientada correctamente. Solución: Añadir un botón en la edición de la imagen para girar esta según el gusto del usuario.
Sacar foto y descartar	No se añade ninguna foto y se muestra un botón para volver a mostrar la cámara o la galería de imágenes.
No se introduce nombre de foto	Se muestra un mensaje indicando que el nombre es obligatorio.
No se introduce lugar de foto	Se muestra un mensaje indicando que el lugar es obligatorio.
No se encuentra la localización el usuario	Se muestra un mensaje indicando que no se encuentra la localización del usuario.

Añadir foto desde la galería de imágenes (aplicación móvil)

Prueba	Resultado Obtenido
Elegir foto	Se muestra correctamente la foto seleccionada.
No se introduce nombre de foto	Se mostrará un mensaje indicando que este campo es obligatorio y no se almacenará la foto. No se enviará nada a la base de datos después de esta tarea.
No se introduce nombre de foto	Se muestra un mensaje indicando que el nombre es obligatorio.
No se introduce lugar de foto	Se muestra un mensaje indicando que el lugar es obligatorio.

No se encuentra la localización el usuario	Se muestra un mensaje indicando que no se encuentra la localización del usuario.
--	--

Eliminar foto (aplicación móvil)	
Prueba	Resultado Obtenido
Eliminar foto	Se deja de mostrar la foto correctamente, ya que no se elimina de la base de datos sino que se pone el borrado a 1

Ver indicaciones al lugar en el que se sacó una foto (aplicación móvil)	
Prueba	Resultado Obtenido
No se encuentra ubicación	La aplicación deja seleccionar punto de origen.
Ubicación encontrada	Se muestran las indicaciones al lugar de la foto correctamente.

Compartir foto (aplicación móvil)	
Prueba	Resultado Obtenido
Compartir foto	Se abre el selector de aplicaciones de Android y al seleccionar un servicio que permita compartir una foto se comparte correctamente.

Exportar mapa (aplicación móvil)	
Prueba	Resultado Obtenido
Exportar por primera vez	La aplicación crea el mapa correctamente pero las imágenes no se suben correctamente, salta un timeout en el servidor. Solución: Bajar el tamaño de las imágenes a exportar, reduciendo mínimamente su calidad para preservar esta.
Exportar mapa privado	En caso de que se marque el mapa como privado y no se introduzca contraseña el mapa no es accesible de ninguna manera. Solución: Se añade un código de contraseña vacía que se introducirá en el caso de que el mapa no sea privado. Si no se introduce contraseña el mapa será público.
Exportar mapa publico	El mapa se exporta correctamente.

Actualizar mapa (aplicación móvil)	
Prueba	Resultado Obtenido
Exportar por segunda vez el mismo mapa (actualizar mapa)	Se actualiza correctamente el mapa pero también se crea un mapa nuevo en la base de datos.

	<p>Solución:</p> <p>Revisar el método de exportación y comprobar correctamente que se trata de una actualización y no enviar la petición de crear el mapa.</p>
--	---

Compartir mapa exportado en las redes sociales (aplicación móvil)

Prueba	Resultado Obtenido
Compartir	Se abre el menú de compartir y se comparte correctamente la url del mapa.

Comentar mapa exportado

Prueba	Resultado Obtenido
Comentar	<p>El mapa no se puede comentar ya que la caja de comentarios da un error diciendo que no se reconoce el dominio de la pagina (localhost). Se prueba con una dirección estática y no se muestra este error pero tampoco deja comentar.</p> <p>Solución:</p> <p>Se compra un dominio para la aplicación web y se apunta a la dirección estática antes comentada.</p>
Eliminar comentario	El comentario se eliminará del mapa.

Generar código para insertar mapa en otra web

Prueba	Resultado Obtenido
Cambiar ancho	El código generado se cambia automáticamente al cambiar este valor.
Cambiar alto	El código generado se cambia automáticamente al cambiar este valor.
Ajustar al ancho	El código generado se cambia automáticamente al cambiar este valor.

Descargar todas las fotos de un mapa

Prueba	Resultado Obtenido
Descargar zip	Se descarga el zip correctamente con las imágenes del mapa.

8.2 Pruebas de Integración y del Sistema

Comunicación aplicación móvil - API	
Prueba	Resultado Esperado
Iniciar sesión	<p>Se iniciar sesión con algunos usuarios pero con otros no.</p> <p>Solución:</p> <p>Para iniciar sesión se genera un token en el cliente y otro en el servidor que se comparan. La forma de generar un md5 en el servidor no es la misma que en Android, y estas no coinciden. Se modifica el método que genera el md5 en Android y se consigue que funcione.</p> <p>Además algunas veces tampoco deja iniciar sesión ya que el token se genera utilizando el minuto del día, para que no sea valido más de un minuto, en el caso de que la petición se haga muy cerca de que acabe un minuto el token no coincidirá por lo que en el servidor se generan dos tokens para comparar, el del minuto actual y el del anterior.</p>
Exportar mapa	El mapa se exporta correctamente.
Borrar foto y actualizar mapa	La foto se borra correctamente del mapa al actualizar este, pero no se borra de la carpeta de imágenes del servidor.
Añadir foto y actualizar mapa	La nueva imagen se añade al mapa y este se actualiza correctamente, la nueva foto también se añade al zip.
Descargar zip	<p>El zip no elimina las fotos eliminadas y las sigue incluyendo pero sí que añade las nuevas fotos actualizadas.</p> <p>Solución:</p> <p>Al eliminar una imagen en el servidor actualizará el contenido del zip.</p>

8.3 Pruebas de Usabilidad y Accesibilidad

8.3.1 Pruebas de Usabilidad

Para la realización de las pruebas de usabilidad se ha contado con la colaboración de 16 personas que se distribuyen en los siguientes perfiles de la siguiente manera:

- Joven: 6
- Adulto: 6
- Anciano: 4

A todos y cada uno de estas personas se les ha hecho realizar un conjunto de tareas en el sistema para ver su reacción y si la aplicación es usable e intuitiva. A continuación se incluyen las tablas desarrolladas en el diseño de las pruebas con los resultados medios de las pruebas aplicadas.

8.3.1.1 Preguntas de carácter general

¿Usa un ordenador frecuentemente?	
5. Todos los días	62,5%
6. Varias veces a la semana	25%
7. Ocasionalmente	
8. Nunca o casi nunca	12,5%
¿Qué tipo de actividades realiza con el ordenador?	
5. Es parte de mi trabajo o profesión	31,25%
6. Lo uso básicamente para ocio	56,6%
7. No lo utilizo	12,5%
8. Únicamente leo el correo y navego ocasionalmente	
¿Ha usado alguna vez software como el de esta prueba?	
4. Sí, he empleado software similar	68,75%
5. No, aunque si empleo otros programas que me ayudan a realizar tareas similares	18,75%

6. No, nunca	12,5%
¿Qué busca Vd. Principalmente en un programa?	
5. Que sea fácil de usar	43,95%
6. Que sea intuitivo	18,75%
7. Que sea rápido	12,3%
8. Que tenga todas las funciones necesarias	25%

8.3.1.1.2 Actividades guiadas

A continuación se muestra la tabla que refleja los resultados medios y los errores cometidos por los usuarios

Tarea	Tiempo medio	Errores	Observaciones
Crear mapa	5 s	No	
Añadir foto al mapa desde la galería	12 s	No	
Añadir foto al mapa desde la cámara	9 s	No	Después de haber añadido a desde la galería añadir desde la cámara no supone un problema ya que se encuentra en el mismo sitio.
Eliminar foto del mapa	23 s	<ul style="list-style-type: none"> Vuelta al listado de mapas Búsqueda de botón en el mapa de imágenes 	Si se pide realizar esta tarea desde el mapa de imágenes se intenta buscar el botón de borrar en este mismo mapa, poco después al cambiar de pantalla no hay problema para eliminar una imagen.
Ver listado de mapa	6 s	No	
Compartir imagen en las redes sociales	7 s	Las personas mayores no saben qué servicio utilizar para compartir una imagen ya que no disponen de cuenta en ninguno de ellos.	No hay problemas en encontrar esta opción en la vista de una imagen.
Indicaciones a una imagen	13 s	Desorientación.	A algunos de los usuarios les desconcierta que se abra otra aplicación para indicar la ruta
Exportar mapa	20 s	No	Se tarda en encontrar la opción de exportar

Registro en la página web	13 s	No	El formulario es muy fácil de encontrar y rápido de rellenar
Iniciar sesión	9 s	No	
Ver mapa exportado	5 s	No	Una vez iniciado sesión ver un mapa es muy intuitivo y rápido
Compartir mapa exportado	10 s	Las personas mayores no identifican la simbología de las aplicaciones para compartir	
Comentar mapa	12 s	No	Algunas personas no tienen cuenta de las necesarias.

No se encuentran mayores problemas en la usabilidad del sistema ya que en su mayor partes se parece a las aplicaciones que se utilizan hoy en día. Se utilizan los mismos iconos estándar y la misma forma de hacer las cosas para que resulte familiar al usuario.

8.3.1.1.3 Preguntas Cortas sobre la Aplicación y Observaciones

A continuación se muestran los resultados medios de las preguntas cortas sobre el sistema

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Sabe donde está dentro de la aplicación?		81,25%	18,75%	
¿Le resulta sencillo el uso de la aplicación?	68,75%	31,25%		
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Funciona cada tarea como Vd. Espera?	81,25%	18,75%		
Echa de menos alguna funcionalidad		37,5%	12,5%	50%
Ve necesaria otra vista alternativa al mapa				100%
¿El tiempo de respuesta de la aplicación es muy grande?				100%
Calidad del Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
El tipo y tamaño de letra es	18,75%	81,25		
Los iconos e imágenes usados son	31,25%	68,75%		
Los colores empleados son	25%	75%		
Diseño de la Interfaz	Si	No	A veces	
¿Le resulta fácil de usar?		87,5%		
¿El diseño de las pantallas es claro y atractivo?		100%		
¿Cree que el programa está bien estructurado?		100%		

A partir de las pruebas realizadas se puede observar que la aplicación es atractiva y fácilmente usable aunque muchos de los usuarios echan en falta más funcionalidades, como añadir videos por ejemplo.

Como esto se excede del alcance del proyecto se añadirá en mejoras del sistema y aplicaciones.

8.3.2 Pruebas de Accesibilidad

8.3.2.1 Descripción de las pruebas

Se desea conseguir un nivel de accesibilidad óptimo teniendo en cuenta que la página generada para la compartición de mapas se ha construido utilizando un diseño novedoso y distinto de lo que se encuentra normalmente en una página web.

Antes de pasar las pruebas se sabe de antemano que no se podrá conseguir un nivel de accesibilidad AA debido al diseño requerido de la aplicación y a los frameworks css y js utilizados.

Para realizar el diseño y funcionalidad de la aplicación, en general la interacción, se ha utilizado mucho javascript y css lo que hace que si se desactivan no se permitirá utilizar de forma correcta la aplicación, mas en concreto la vista de mapas.

8.3.2.2 Equipamiento para las pruebas

Las pruebas de accesibilidad del sistema se realizarán con el siguiente equipamiento:

- Macbook pro 10''
- MacOSx
- Windows 7
- Google Chrome
- Mozilla Firefox
- Safari
- Internet Explorer

8.3.2.3 Páginas a analizar

Las paginas a analizar en las pruebas de accesibilidad del sistema han sido escogidas teniendo en cuenta los elementos que estas contienen y haciendo que entre las páginas seleccionadas se obtengan todos los elementos que se deben evaluar en estas pruebas.

Las páginas elegidas para estas pruebas son:

- Home
- Vista de mapa

8.3.2.4 Aspectos positivos en la accesibilidad

Se ha realizado una revisión preliminar el sitio en el que se verifican las siguientes pruebas:

- **Cambio del tamaño de texto de la aplicación:** al cambiar el tamaño de texto desde el navegador la aplicación sigue siendo usable. La página se adapta fácilmente a su contenido.
- **Distintas resoluciones:** se ha comprobado que la aplicación se ve bien en distintas resoluciones ya que con el diseño responsivo esta se adapta al ancho de la página generando así una vista especial para cada ancho.
- **Distintas pantallas:** la aplicación web se ha probado en distintos tamaños de pantalla (13, 15, 17, 20 pulgadas). El contenido de la página se ve bien en todas ellas y es posible su uso.
- **Utilización de la página con el teclado:** La página funciona correctamente utilizándose con el teclado, con las flechas podremos elegir las fotos que queremos ver y con el tabulador nos moveremos entre los diferentes enlaces de la aplicación.
- **Diferentes navegadores:** Se ha comprobado el correcto funcionamiento en los siguientes exploradores:
 - **IE 9**
 - **Google Chrome**
 - **Firefox**
 - **Safari**

8.3.2.5 Problemas encontrados

- Desactivando el css y el JavaScript en la página de vista de mapa se pierden funcionalidades, esto es normal ya que esta página ha sido diseñada con este fin, con el fin de aplicar un concepto nuevo a la compartición de mapas y un diseño novedoso. No se ha encontrado la manera de que se pueda utilizar sin css y JavaScript.
- Se han corregido muchos fallos de la aplicación para que se muestre correctamente en Internet Explorer 8, pero no se ha conseguido su funcionalidad completa.
- No se ha podido pasar corregir alguna regla de accesibilidad como la de no utilizar elementos de diseño en el html ya que bootstrap (el framework css utilizado en la página web) utiliza el elemento `<i>` para los iconos de los botones.
- Se han detectado fallos en la interacción utilizando el teclado que se han corregido modificando la estructura del html de forma que el propio navegador consigue utilizar la web sin necesidad de ratón.
- Tamaños fijos o posicionamiento absoluto: para conseguir el diseño deseado en la vista de un mapa es necesario el posicionamiento absoluto de algunos elementos. Como anteriormente se ha mencionado, lo que se busca en este proyecto es un diseño distinto, una nueva forma de presentar la información y esto no puede ser posible sin el formateo de esta información.
- Validación css: el propio css de bootstrap no pasa la validación de css del W3C ya que tiene filtros y hacks para lidiar con errores de Internet Explorer.
- Desactivar CSS: el css en la pantalla de vista de mapa es muy importante para la aplicación. Si este se desactiva no es posible ver correctamente el mapa debido a que la vista de mapa ocupa todo el fondo de la aplicación, las imágenes subidas por el usuario no controlan su tamaño... Esta aplicación se ha desarrollado para generar una vista atractiva del mapa y así compartirla con otras personas, no tiene mucho sentido acceder a la vista de un mapa sin css.

8.3.2.6 Checklist del WCAG 1.0

En general (Prioridad 1)	Sí	No	N/A
1.1 Proporcione un texto equivalente para todo elemento no textual (Por ejemplo, a través de "alt", "longdesc" o en el contenido del elemento). <i>Esto incluye:</i> imágenes, representaciones gráficas del texto, mapas de imagen, animaciones (Por ejemplo, <i>GIFs</i> animados), "applets" y objetos programados, "ascii art", marcos, scripts, imágenes usadas como viñetas en las listas, espaciadores, botones gráficos, sonidos (ejecutados con o sin interacción del usuario), archivos exclusivamente auditivos, banda sonora del vídeo y vídeos.	X		
2.1 Asegúrese de que toda la información transmitida a través de los colores también esté disponible sin color, por ejemplo mediante el contexto o por marcadores.	X		
4.1 Identifique claramente los cambios en el idioma del texto del documento y en cualquier texto equivalente (por ejemplo, leyendas).			X
6.1 Organice el documento de forma que pueda ser leído sin hoja de estilo. Por ejemplo, cuando un documento HTML es interpretado sin asociarlo a una hoja de estilo, tiene que ser posible leerlo.		X	
6.2 Asegúrese de que los equivalentes de un contenido dinámico son actualizados cuando cambia el contenido dinámico.	X		
7.1 Hasta que las aplicaciones de usuario permitan controlarlo, evite provocar destellos en la pantalla.			X
14.1 Utilice el lenguaje apropiado más claro y simple para el contenido de un sitio.	X		
Y si utiliza imágenes y mapas de imagen (Prioridad 1)	Sí	No	N/A
1.2 Proporcione vínculos redundantes en formato texto para cada zona activa de un mapa de imagen del servidor.	X		
9.1 Proporcione mapas de imagen controlados por el cliente en lugar de por el servidor, excepto donde las zonas sensibles no puedan ser definidas con una forma geométrica.	X		
Y si utiliza tablas (Prioridad 1)	Sí	No	N/A
5.1 En las tablas de datos, identifique los encabezamientos de fila y columna.			X
5.2 Para las tablas de datos que tienen dos o más niveles lógicos de encabezamientos de fila o columna, utilice marcadores para asociar las celdas de encabezamiento y las celdas de datos.			X
Y si utiliza marcos ("frames") (Prioridad 1)	Sí	No	N/A
12.1 Titule cada marco para facilitar su identificación y navegación.	X		
Y si utiliza "applets" y "scripts" (Prioridad 1)	Sí	No	N/A
6.3 Asegure que las páginas sigan siendo utilizables cuando se desconecten o no se soporten los scripts, <i>applets</i> u otros objetos programados. Si esto no es posible, proporcione información		X	

equivalente en una página alternativa accesible.			
Y si utiliza multimedia (Prioridad 1)	Sí	No	N/A
1.3 Hasta que las aplicaciones de usuario puedan leer en voz alta automáticamente el texto equivalente de la banda visual, proporcione una descripción auditiva de la información importante de la banda visual de una presentación multimedia.			X
1.4 Para toda presentación multimedia dependiente del tiempo (por ejemplo, una película o animación) sincronice alternativas equivalentes (por ejemplo, subtítulos o descripciones de la banda visual) con la presentación.			X
Y si todo lo demás falla (Prioridad 1)	Sí	No	N/A
11.4 Si, después de los mayores esfuerzos, no puede crear una página accesible, proporcione un vínculo a una página alternativa que use tecnologías W3C, sea accesible, tenga información (o funcionalidad) equivalente y sea actualizada tan a menudo como la página (original) inaccesible.		X	

Puntos de verificación Prioridad 2:

En general (Prioridad 2)	Sí	No	N/A
2.2 Asegúrese de que las combinaciones de los colores de fondo y primer plano tengan el suficiente contraste para que sean percibidas por personas con deficiencias de percepción de color o en pantallas en blanco y negro [Prioridad 2 para las imágenes. Prioridad 3 para los textos].	X		
3.1 Cuando exista un marcador apropiado, use marcadores en vez de imágenes para transmitir la información.	X		
3.2 Cree documentos que estén validados por las gramáticas formales publicadas.			X
3.3 Utilice hojas de estilo para controlar la maquetación y la presentación.	X		
3.4 Utilice unidades relativas en lugar de absolutas al especificar los valores en los atributos de los marcadores de lenguaje y en los valores de las propiedades de las hojas de estilo.	X		
3.5 Utilice elementos de encabezado para transmitir la estructura lógica y utilícelos de acuerdo con la especificación.	X		
3.6 Marque correctamente las listas y los ítems de las listas.	X		
3.7 Marque las citas. No utilice el marcador de citas para efectos de formato tales como sangrías.	X		
6.5 Asegúrese de que los contenidos dinámicos son accesibles o proporcione una página o presentación alternativa.		X	
7.2 Hasta que las aplicaciones de usuario permitan controlarlo, evite el parpadeo del contenido (por ejemplo, cambio de presentación en periodos regulares, así como el encendido y apagado).	X		
7.4 Hasta que las aplicaciones de usuario proporcionen la posibilidad de detener las actualizaciones, no cree páginas que se actualicen automáticamente de forma periódica.	X		
7.5 Hasta que las aplicaciones de usuario proporcionen la posibilidad de detener el redireccionamiento automático, no utilice marcadores para redirigir las páginas automáticamente. En su lugar, configure el servidor para que ejecute esta posibilidad.	X		

10.1 Hasta que las aplicaciones de usuario permitan desconectar la apertura de nuevas ventanas, no provoque apariciones repentinas de nuevas ventanas y no cambie la ventana actual sin informar al usuario.	X		
11.1 Utilice tecnologías W3C cuando estén disponibles y sean apropiadas para la tarea y use las últimas versiones que sean soportadas.	X		
11.2 Evite características desaconsejadas por las tecnologías W3C.	X		
12.3 Divida los bloques largos de información en grupos más manejables cuando sea natural y apropiado.	X		
13.1 Identifique claramente el objetivo de cada vínculo.	X		
13.2 Proporcione metadatos para añadir información semántica a las páginas y sitios.	X		
13.3 Proporcione información sobre la maquetación general de un sitio (por ejemplo, mapa del sitio o tabla de contenidos).		X	
13.4 Utilice los mecanismos de navegación de forma coherente.	X		
Y si utiliza tablas (Prioridad 2)	Sí	No	N/A
5.3 No utilice tablas para maquetar, a menos que la tabla tenga sentido cuando se alinee. Por otro lado, si la tabla no tiene sentido, proporcione una alternativa equivalente (la cual debe ser una versión alineada).	X		
5.4 Si se utiliza una tabla para maquetar, no utilice marcadores estructurales para realizar un efecto visual de formato.	X		
Y si utiliza marcos ("frames") (Prioridad 2)	Sí	No	N/A
12.2 Describa el propósito de los marcos y cómo éstos se relacionan entre sí, si no resulta obvio solamente con el título del marco.	X		
Y si utiliza formularios (Prioridad 2)	Sí	No	N/A
10.2 Hasta que las aplicaciones de usuario soporten explícitamente la asociación entre control de formulario y etiqueta, para todos los controles de formularios con etiquetas asociadas implícitamente, asegúrese de que la etiqueta está colocada adecuadamente.	X		
12.4 Asocie explícitamente las etiquetas con sus controles.	X		
Y si utiliza "applets" y "scripts" (Prioridad 2)	Sí	No	N/A
6.4 Para los <i>scripts</i> y <i>applets</i> , asegúrese de que los manejadores de eventos sean independientes del dispositivo de entrada.	X		
7.3 Hasta que las aplicaciones de usuario permitan congelar el movimiento de los contenidos, evite los movimientos en las páginas.		X	
8.1 Haga los elementos de programación, tales como <i>scripts</i> y <i>applets</i> , directamente accesibles o compatibles con las ayudas técnicas [Prioridad 1 si la funcionalidad es importante y no se presenta en otro lugar; de otra manera, Prioridad 2].			X
9.2 Asegúrese de que cualquier elemento que tiene su propia interfaz pueda manejarse de forma independiente del dispositivo.		X	
9.3 Para los "scripts", especifique manejadores de evento lógicos mejor que manejadores de eventos dependientes de dispositivos.	X		

Puntos de verificación Prioridad 3:

En general (Prioridad 3)	Sí	No	N/A
4.2 Especifique la expansión de cada abreviatura o acrónimo cuando aparezcan por primera vez en el documento.			X

4.3 Identifique el idioma principal de un documento.	X		
9.4 Cree un orden lógico para navegar con el tabulador a través de vínculos, controles de formulario y objetos.	X		
9.5 Proporcione atajos de teclado para los vínculos más importantes (incluidos los de los mapas de imagen de cliente), los controles de formulario y los grupos de controles de formulario.		X	
10.5 Hasta que las aplicaciones de usuario (incluidas las ayudas técnicas) interpreten claramente los vínculos contiguos, incluya caracteres imprimibles (rodeados de espacios), que no sirvan como vínculo, entre los vínculos contiguos.	X		
11.3 Proporcione la información de modo que los usuarios puedan recibir los documentos según sus preferencias (por ejemplo, idioma, tipo de contenido, etc.).		X	
13.5 Proporcione barras de navegación para destacar y dar acceso al mecanismo de navegación.	X		
13.6 Agrupe los vínculos relacionados, identifique el grupo (para las aplicaciones de usuario) y, hasta que las aplicaciones de usuario lo hagan, proporcione una manera de evitar el grupo.	X		
13.7 Si proporciona funciones de búsqueda, permita diferentes tipos de búsquedas para diversos niveles de habilidad y preferencias.			X
13.8 Localice la información destacada al principio de los encabezamientos, párrafos, listas, etc.	X		
13.9 Proporcione información sobre las colecciones de documentos (por ejemplo, los documentos que comprendan múltiples páginas).	X		
13.10 Proporcione un medio para saltar sobre un <i>ASCII art</i> de varias líneas.			X
14.2 Complemente el texto con presentaciones gráficas o auditivas cuando ello facilite la comprensión de la página.			
14.3 Cree un estilo de presentación que sea coherente para todas las páginas.	X		
Y si utiliza imágenes o mapas de imagen (Prioridad 3)	Sí	No	N/A
1.5 Hasta que las aplicaciones de usuario interpreten el texto equivalente para los vínculos de los mapas de imagen de cliente, proporcione vínculos de texto redundantes para cada zona activa del mapa de imagen de cliente.	X		
Y si utiliza tablas (Prioridad 3)	Sí	No	N/A
5.5 Proporcione resúmenes de las tablas.			X
5.6 Proporcione abreviaturas para las etiquetas de encabezamiento.			X
10.3 Hasta que las aplicaciones de usuario (incluidas las ayudas técnicas) interpreten correctamente los textos contiguos, proporcione un texto lineal alternativo (en la página actual o en alguna otra) para <i>todas</i> las tablas que maquetan texto en paralelo, en columnas de palabras.			X
Y si utiliza formularios (Prioridad 3)	Sí	No	N/A
10.4 Hasta que las aplicaciones de usuario manejen correctamente los controles vacíos, incluya caracteres por defecto en los cuadros de edición y áreas de texto.	X		

8.4 Pruebas de Rendimiento

Para realizar las pruebas de rendimiento en la pagina se ha utilizado la pagina web pingdom que realiza la medida de tiempos de carga.

Accion	Tiempo máximo	Tiempo obtenido
<i>Cargar pagina home (Chrome)</i>	2-4 segundos	<i>1,62 segundos</i>
<i>Cargar mapa</i>	4-7 segundos	<i>7,35 segundos</i>
<i>Comentar mapa</i>	2 segundos	<i>2,6 segundos</i>
<i>Exportar una única foto</i>	10 segundos (depende de la conexión)	<i>5,6 segundos</i>
<i>Mostrar mapa aplicación móvil</i>	1 segundo	<i>1,3 segundos</i>
<i>Añadir foto aplicación móvil</i>	1 segundo	<i>1,6 segundos</i>

La home se carga rápidamente, aunque el mapa insertado en ella tarda más, pero mientras este se carga la home es totalmente usable.

El tiempo de carga de un mapa depende de la cantidad de imágenes que este tenga, pero mientras se cargan las imágenes el mapa es usable a partir de los 3,2 segundos.

La pagina tiene un rendimiento aceptable teniendo en cuenta que se encuentra en el ordenador de clase y no en un servidor con más recursos.

Capítulo 9. Manuales del Sistema

9.1 Manual de Instalación

9.1.1 Instalación de la aplicación móvil

Para la instalación de la aplicación móvil únicamente es necesario instalar el apk de Map it!

Enviando este apk al dispositivo por medio del correo electrónico, cable, dropbox o cualquier otra forma únicamente se tendrá que hacer click en la aplicación y el dispositivo la instalará automáticamente.

En el caso de que no deje instalar la aplicación se deberá a un problema con la firma de aplicaciones y habrá que hacer lo siguiente:

Ir a ajustes > aplicaciones y marcar la opción de permitir aplicaciones no firmadas para google play. Esto puede variar dependiendo del sistema operativo del dispositivo.

En el caso de que se desee instalar la aplicación móvil para funcionar con la aplicación web que se encuentre en otro servidor se deberá cambiar la clase WController para cambiar la dirección del servidor. Se cambiará el atributo API_IP.

9.1.2 Instalación de la aplicación web

Para instalar la aplicación web será necesario instalar en un servidor el apache Tomcat.

Para ello podremos descargarlo en la siguiente página:

<http://tomcat.apache.org/download-70.cgi>

Instalaremos tomcat como servicio en nuestro servidor.

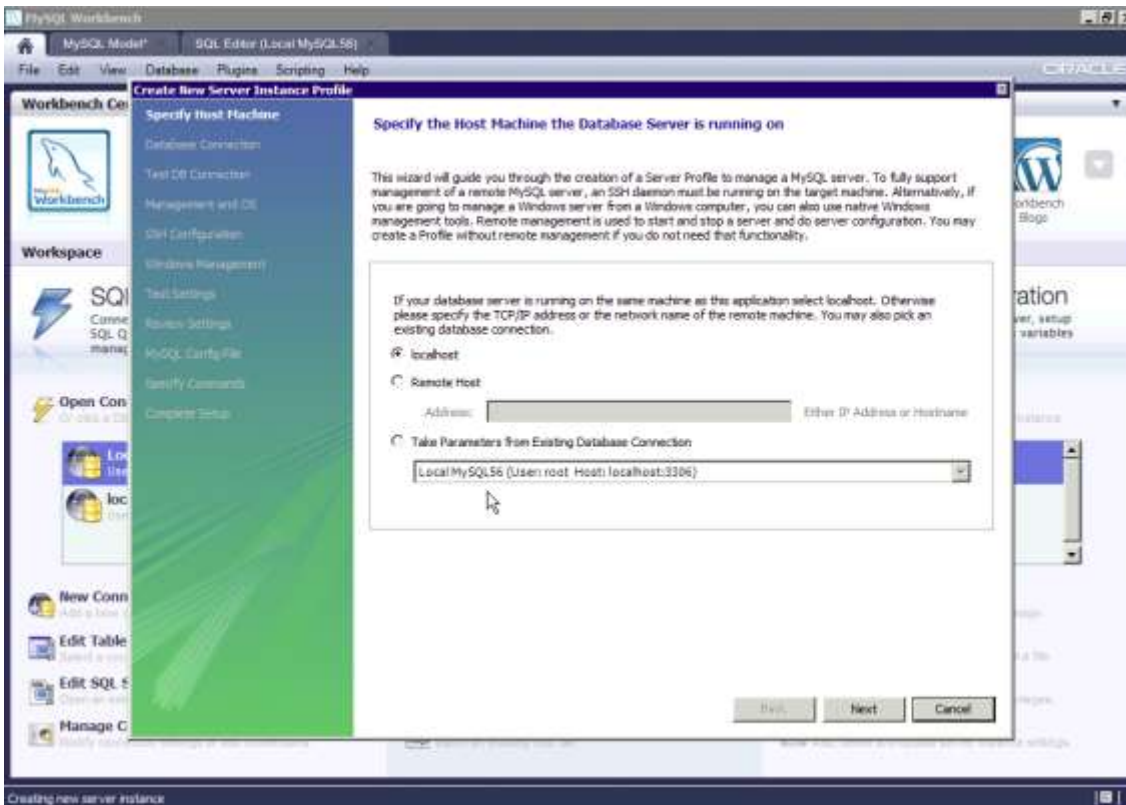
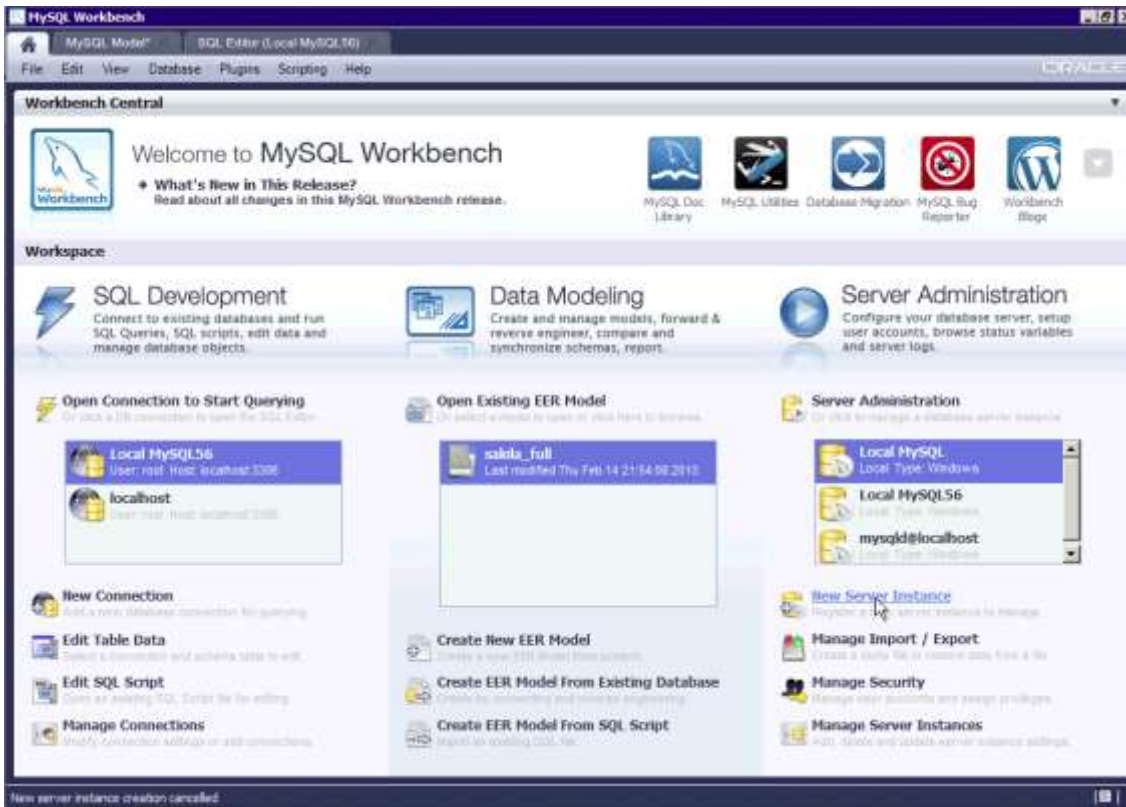
Además tendremos que instalar el servidor de base de datos:

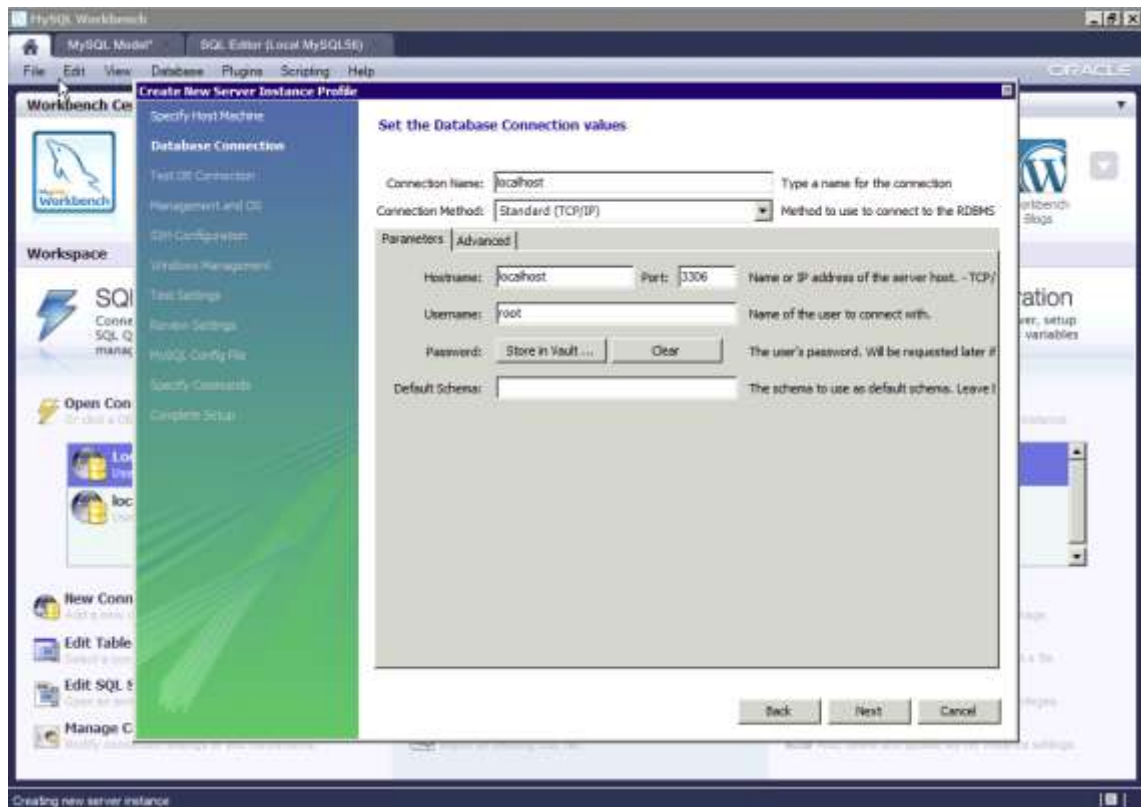
<http://dev.mysql.com/downloads/mysql/>

Podemos instalar todas las funcionalidades y así tendremos instalado el Mysql workbench que nos permitirá gestionar las bases de datos.

Deberemos crear una nueva conexión a la base de datos y cargar el archivo .sql que se adjunta con la documentación para crear la base de datos en el servidor.

Pulsamos en New Server Instance y seguimos los pasos de las capturas de pantalla.

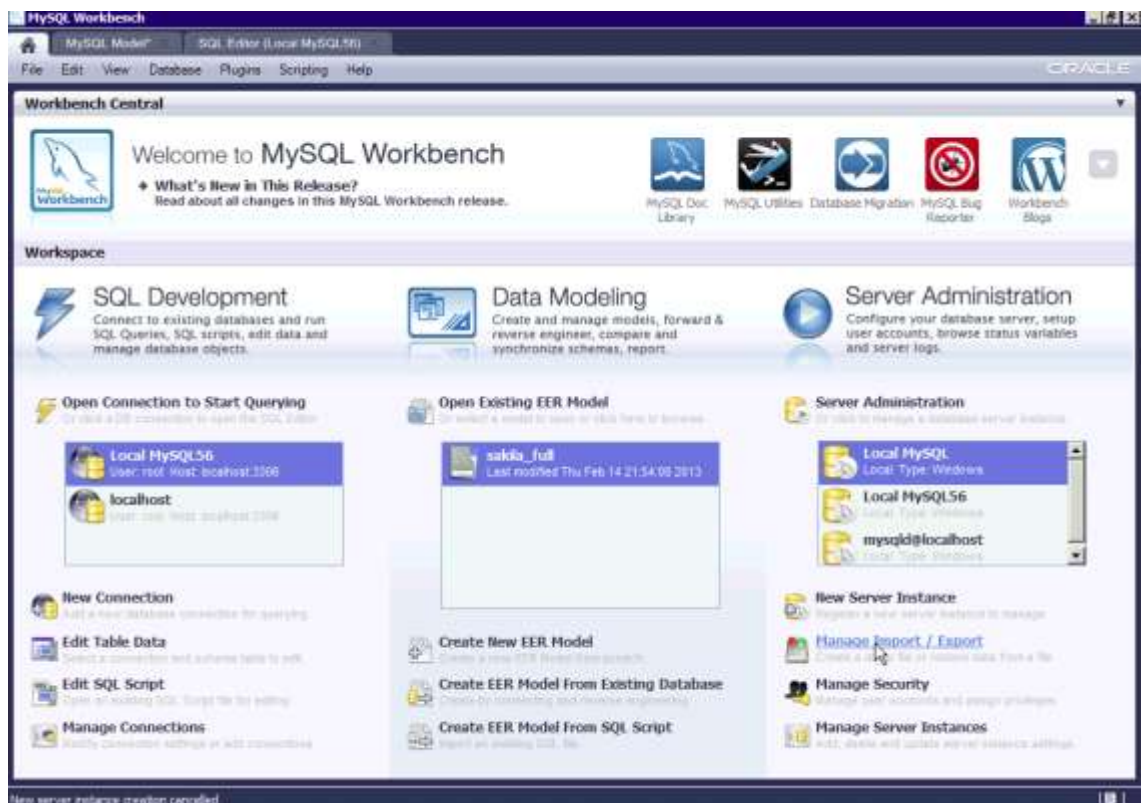




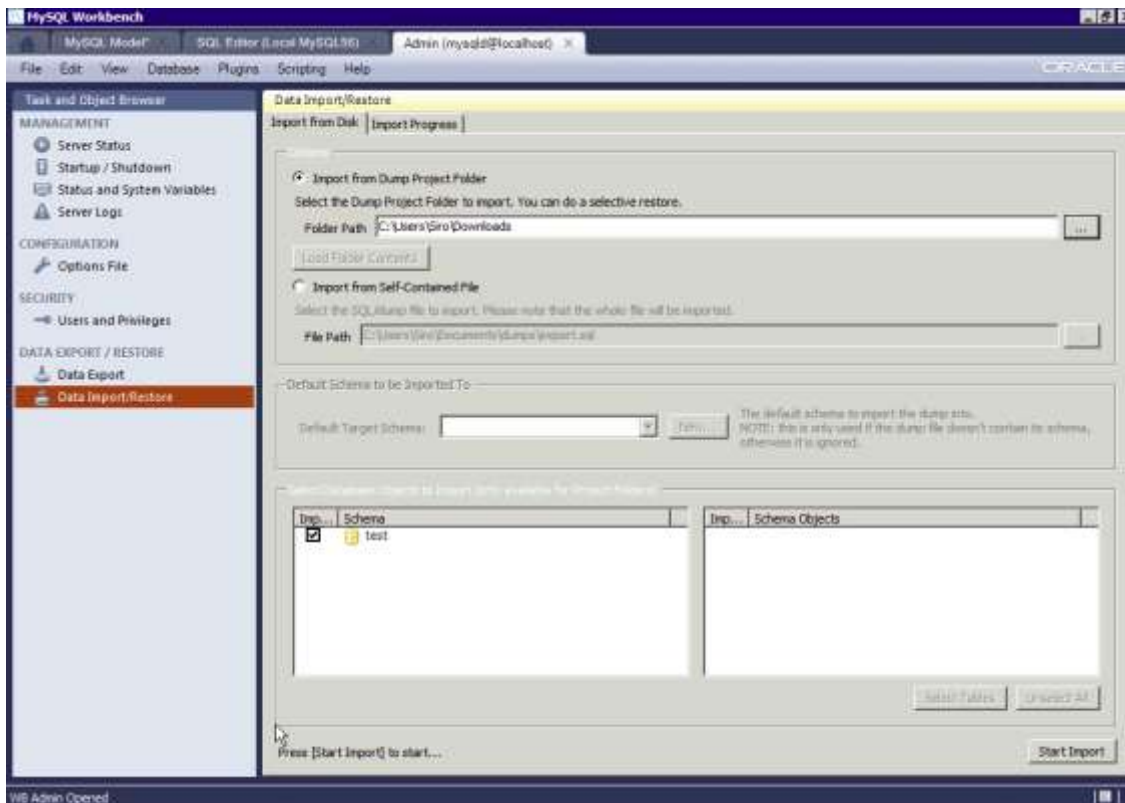
Automáticamente se nos crea una conexión a la base de datos que utilizaremos para conectarnos desde el mysql workbench y desde la aplicación web.

Para cargar la copia de la estructura de la base de datos haremos lo siguiente:

Pulsamos en Manage Import / Export:



En la columna de la izquierda seleccionamos Data Import y buscamos la ruta en la que tengamos el archivo sql. Al seleccionar la carpeta ya nos encontrará la base de datos. Únicamente tenemos que pulsar en importar.



Después de instalar la base de datos únicamente tendremos que localizar la carpeta webapps de nuestro Tomcat (en el lugar que lo hayamos instalado) y arrastrar a esta carpeta el archivo ROOT.war. Automáticamente el tomcat desplegará nuestra aplicación y la podremos ver en <http://localhost>

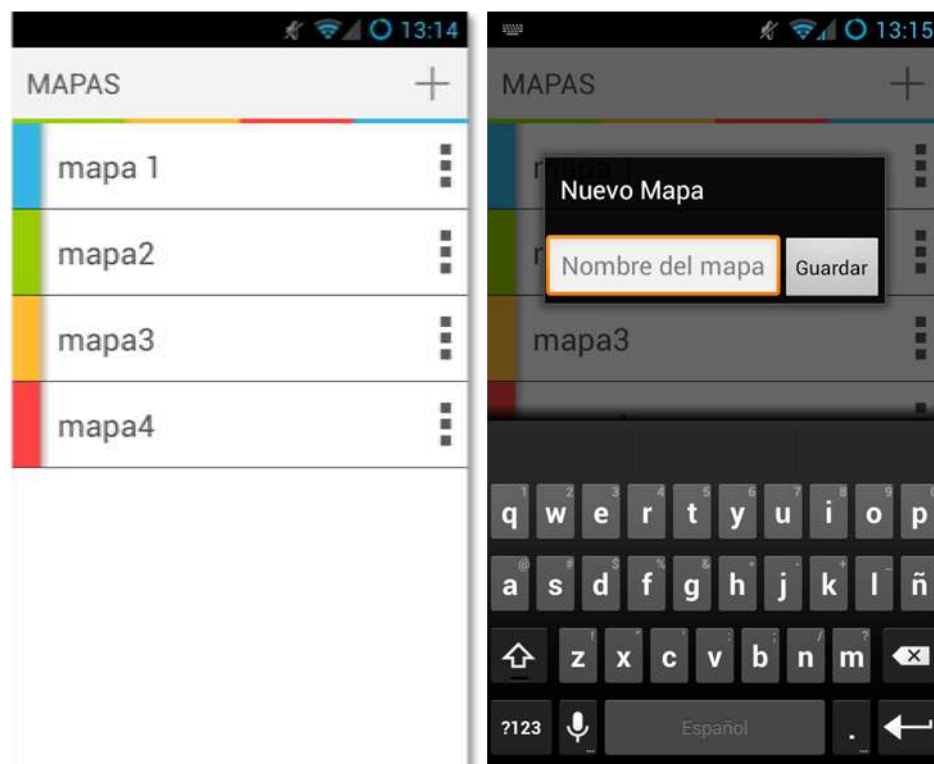
9.2 Manual de Usuario

En esta apartado se explicará cómo realizar cada una de las tareas posibles en el sistema.

9.2.1 Aplicación móvil

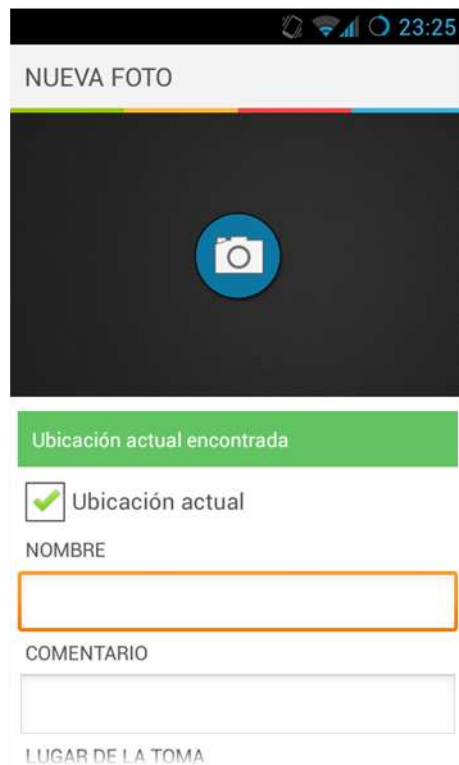
9.2.1.1 Crear mapa

En la pantalla principal de la aplicación al pulsar en el botón de la esquina superior derecha se nos abrirá un popup en el que introduciremos el nombre del mapa. Al pulsar aceptar este se añadirá a la lista y podremos acceder al mapa pulsando sobre él.



9.2.1.2 Añadir foto a mapa

Dentro de la vista de un mapa podremos ver un botón en la parte inferior de la pantalla con el símbolo de una cámara. Pulsando este botón iremos al formulario de creación de una nueva foto para el mapa seleccionado.



Se nos abra un popup en el que elijeremos si sacar una foto con la camara o elegirla de la galeria y una vez elegida la foto se nos cargará en el formulario. Despues deberemos rellenar el resto de datos del formulario.



Al guardar la foto veremos la vista de detalle de la misma.



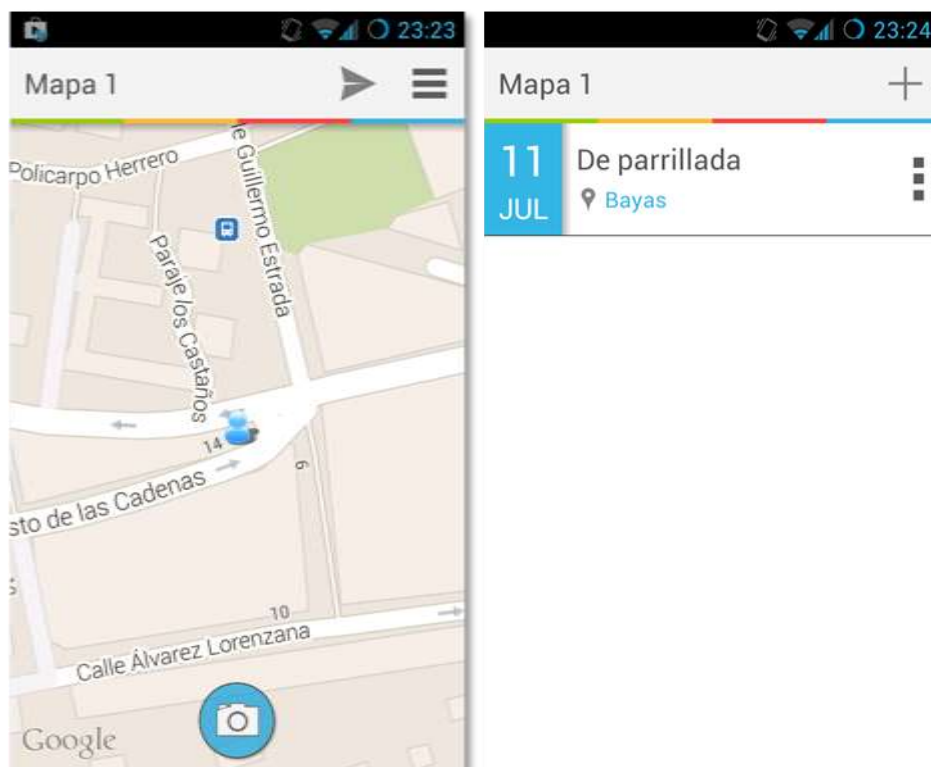
9.2.1.3 Indicaciones al lugar de una imagen y compartir imagen

Las opciones de indicaciones al lugar de una imagen y compartir una imagen están situadas en la barra superior de la vista de detalle de una imagen.



9.2.1.4 Ver listado de imágenes

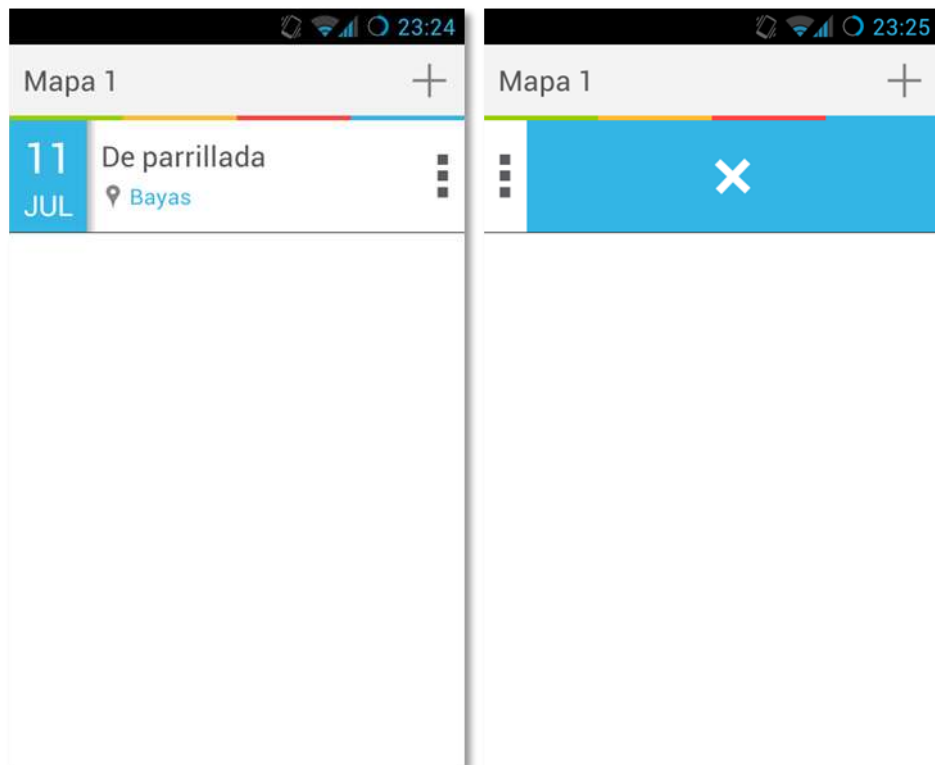
Para ver el listado de imágenes desde la vista de mapa se pulsará el botón de la esquina superior derecha.



Desde la vista de listado también será posible eliminar una imagen.

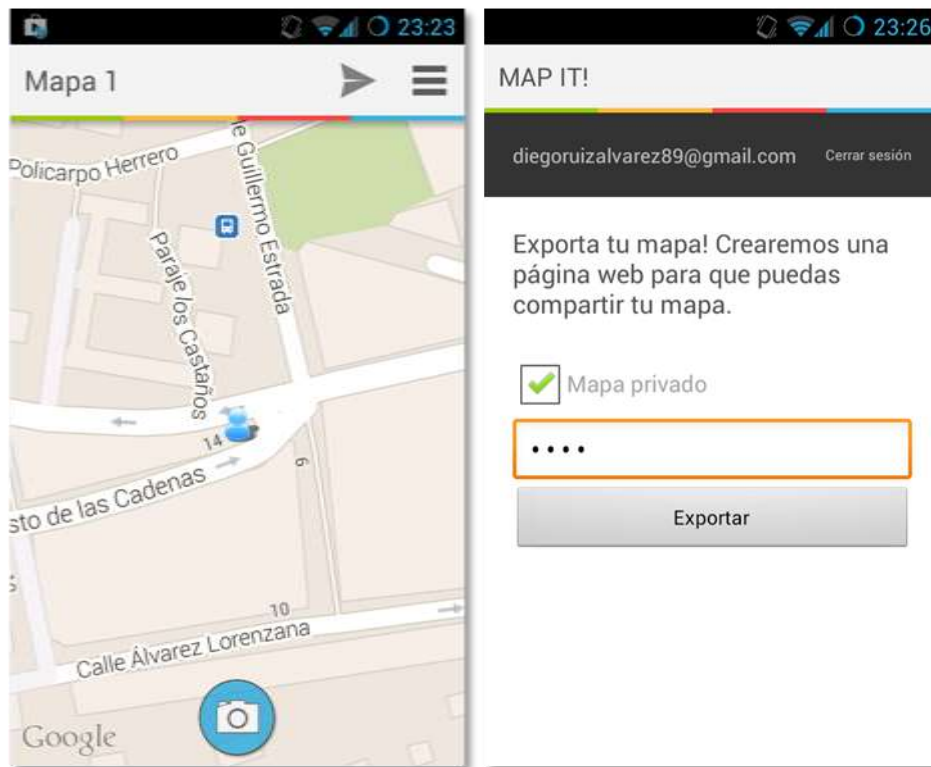
9.2.1.5 Eliminar fotos

Desde la vista de listado de fotos pulsando en el botón a la derecha de un elemento del listado se abrirán las opciones para este elemento. Pulsando en la cruz eliminaremos la foto.



9.2.1.6 Exportar mapa

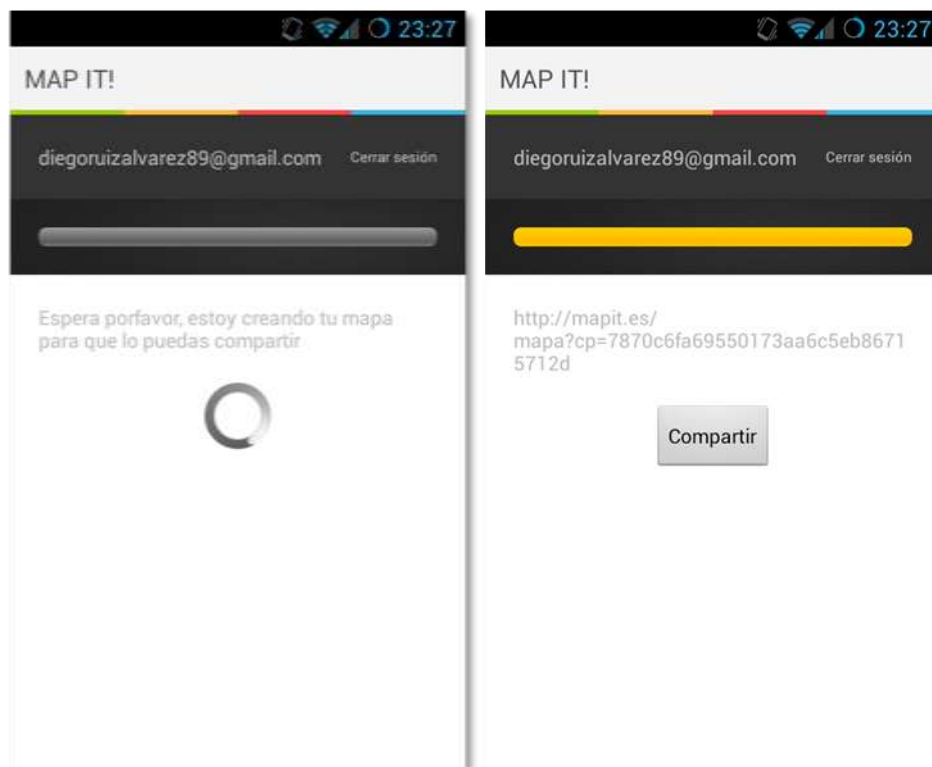
Para exportar un mapa desde la vista de mapa pulsando en el botón del “avión de papel” se entrará en el panel de exportación.



En esta pantalla seleccionaremos si queremos exportar un mapa privado o no marcando o desmarcando el check que pone “mapa privado”, en caso de marcarlo aparecerá un campo para que introduzcamos una contraseña para el mapa.

Al pulsar en el botón exportar comenzara la exportación.

En la barra superior de progreso se mostrará el % de imágenes exportadas, cuando esta barra se llene completamente se mostrará la url del mapa exportado.



Pulsando en el botón compartir podremos compartir el enlace generado por la aplicación.

9.2.1.7 Registro

Para registrarse en el sistema y por tanto poder iniciar sesión es necesario registrarse en el sistema.

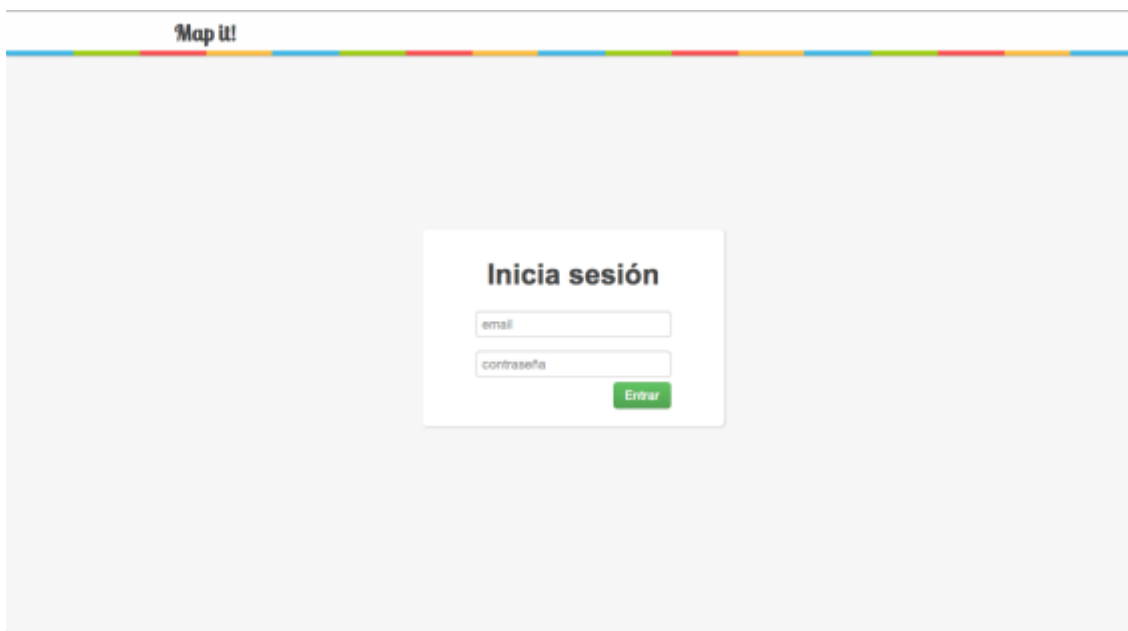
Para ello en la página <http://mapit.es> se rellenara el formulario de registro.



9.2.1.8 Ver mapas exportados

Para ver los mapas exportados de un usuario nos dirigiremos a la página web de la aplicación. En este caso <http://mapit.es>

Aquí iniciaremos sesión en web y accederemos al panel de control del usuario que haya iniciado sesión.



Aquí pulsando sobre ver mapa iremos a la vista de ver mapa.



9.2.1.9 Comentar mapa

En la vista de mapa en la esquina superior derecha se puede ver un botón que pone “comentar”. Pulsando este botón se desplegará una caja de comentarios con la que comentar este mapa.



9.2.1.10 Generar código para insertar mapa en otra web

En la vista del mapa que se desea insertar en otra web pulsando en el botón insertar mapa se desplegará una caja en la que se especificaran las medidas del mapa a insertar. Copiando el código generado en tu pagina web podrás ver el mapa desde ella.



9.2.1.11 Descargar zip

Para descargar el zip desde la vista de mapa en la primera tarjeta de la izquierda donde se muestra el nombre del mapa pulsaremos en el botón de descargar archivos.

9.3 Manual del Programador

9.3.1 Configuración de la aplicación web

La aplicación web sigue un el Modelo Vista Controlador. De forma que se estructura el código en capas.

En la fase de diseño se explica la estructura de paquetes de la aplicación que se deberá seguir durante el desarrollo.

Como se comentó en el diseño se tiene una capa de servicios y una capa de persistencia (DAOS). Estas capas se relacionan entre sí mediante la inyección de dependencias de Spring Beans. Por eso no es necesario crear instancias de los servicios o los DAOs ya que la inyección de dependencias crea un singleton de estos objetos.

Para configurar la ruta en la que se almacenan las imágenes recogidas por el servidor y la ruta base de toda la aplicación se modificará el fichero Configuration.properties que se encuentra en el paquete default junto al archivo struts.xml que gestiona el enrutado de la aplicación web.

La configuración de la base de datos se hará en la implementación del Helper de Mysql aquí podremos cambiar la ruta de la base de datos y al usuario o contraseña de la misma. Debemos modificar el archivo MysqlHelper.java

9.3.2 Configuración aplicación móvil

Para hacer cualquier cambio que esté relacionado con la comunicación con el servidor se deberá cambiar el archivo WSController ya que en este se indica la dirección al servidor web central.

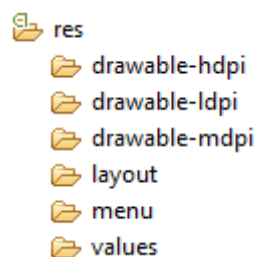
En el diseño se puede observar la estructura de paquetes en la que se indica que tipo de clases se incluyen en ellos, se deberá seguir esta estructura de organización alrededor de toda la aplicación.

A continuación se explican los pasos básicos para añadir recursos y nuevas actividades a la aplicación.

9.3.2.1 Recursos y Android Manifest

En toda aplicación para Android se tiene una carpeta para los recursos de la misma. Esta carpeta se llama “res” y se encuentra en el directorio raíz de nuestro proyecto.

La estructura de esta carpeta es la siguiente:



- Drawable-hdpi: En esta carpeta se introducen las imágenes que se utilizarán en la aplicación cuando se puedan mostrar en alta resolución. En esta carpeta están almacenadas las imágenes utilizadas tanto para los botones como para el fondo de la aplicación.
- Drawable-ldpi: En esta carpeta se introducen las imágenes que se utilizarán en la aplicación cuando se puedan mostrar en baja resolución.
- Drawable-mdpi: En esta carpeta se introducen las imágenes que se utilizarán en la aplicación cuando se puedan mostrar en una resolución intermedia.
- Layout: En esta carpeta se almacenan los layouts de las pantallas de la aplicación en formato XML
- Menú: En la carpeta se guardan los XML de configuración de los menús de la aplicación.
- Values: La carpeta values de un proyecto sirve para guardar XML de configuración de determinados componentes. En el caso de este proyecto se guarda el fichero arrays.xml para guardar los idiomas del desplegable y el fichero strings.xml que viene por defecto en todos los proyectos para guardar los nombres de los componentes.

En el directorio raíz del proyecto también se encuentra el fichero **AndroidManifest.xml**. Este fichero es el más importante de la aplicación, puesto que en él se especifican todas las

características del proyecto. Para añadir una Activity se debe reflejar en este fichero, así como la versión de la aplicación, la clase que se va a ejecutar al iniciar la aplicación, los permisos de la aplicación (en este caso permisos de cámara)....etc.

Capítulo 10. Conclusiones y Ampliaciones

y

10.1 Conclusiones

Se ha desarrollado un sistema que permite la compartición de fotos geolocalizadas para realizar un seguimiento completo de un viaje. Además se ha conseguido que la información recolectada por el usuario pueda ser actualizada a tiempo real, lo que facilita, al compartir un mapa, el seguimiento por parte de otras personas de este mapa. Se ha integrado el sistema con las redes sociales actuales de forma que la compartición de los elementos de un mapa o la compartición del mismo mapa es una tarea sencilla.

Se ha conseguido construir un sistema que cumpla con los requisitos iniciales y por tanto que realice todas las funcionalidades descritas, además se ha provisto a todo este sistema de un diseño atractivo e intuitivo que facilita su uso.

En cuanto a la elección de las tecnologías utilizadas en este trabajo creo que han sido las correctas, ya que en vistas de ser ampliado el proyecto Java es un lenguaje de programación fácil de aprender y de entender y Android es uno de los principales sistemas operativos móviles del mercado cuyo uso aumenta constantemente. Se ha intentado estructurar lo más posible el código para que su ampliación sea fácil y ordenada.

El sistema construido es completamente usable y podría utilizarse en un servidor real para que cualquier persona lo usase ya que dispone de todas las funcionalidades necesarias para su uso, sin depender de desarrollos de terceros, o suposiciones.

10.2 Ampliaciones

10.2.1 Introducción de videos

Se contempla la introducción de videos en el mapa. Esta funcionalidad no es tan parecida a la de añadir fotos como puede parecer. Utilizando la propia cámara del dispositivo los videos pueden ser demasiado largos o con demasiada calidad. Para desarrollar esta funcionalidad sería necesario desarrollar una cámara especializada para la aplicación de modo que se controle la duración máxima del video y su calidad.

10.2.2 Viajes gestionados por varios usuario

Lo que se espera de esta funcionalidad es que un usuario al crear un mapa pueda invitar a los usuarios que desee pudiendo gestionar el mapa entre todos ellos, es decir, subiendo fotos al mismo mapa y exportándolas en este. De esta forma al viajar varias personas juntas pueden crear un mapa conjunto.

Para ello se debería crear una especie de sistema de amigos lo que haría al sistema más social que era lo que no se pretendía. Podría buscarse otra solución que no implique tener amigos en el sistema pero si en otras redes sociales, como por ejemplo utilizar la API de Facebook para encontrar tus amigos que utilizan Map it!.

10.2.3 Comentarios en fotos

Cada una de las fotos de un mapa (mapa exportado) debería poder ser comentada por los visitantes a este mapa. De esta forma se separarían los comentarios globales del mapa con los de cada foto.

10.2.4 Sistema de valoraciones de mapas y top de mapas

Se desarrollará un sistema de valoraciones por el cual los usuarios puntuen los mapas, de modo que esta valoración indique lo útil que ha podido ser un mapa. Los usuarios pueden utilizar los mapas para crear guías turísticas de una determinada zona del mundo por lo que esta valoración haría que los mejores mapas salieran en un top de mapas que se incluiría en la página web.

10.2.5 Tags y búsqueda

Se proveerá al sistema de una gestión de tags para cada mapa. El usuario al crear el mapa podrá añadir etiquetas (tags) a este de forma que estas ayuden a encontrar este mapa en un buscador. Así otros usuarios podrán encontrar un mapa si esta etiquetado con alguno de los criterios que están buscando.

10.2.6 Aplicación para IOs

Se desarrollará la misma aplicación de Android pero para el sistema operativo móvil IOs. IOs es otro de los principales sistemas operativos del mercado actual y de esta forma se ampliaría el rango de personas que podrían utilizar la aplicación.

Capítulo 11. Presupuesto

11.1 Costes de desarrollo del proyecto

Los costes de desarrollo del proyecto incluyen tanto el coste del hardware utilizado para el mismo como el del software empleado.

Las herramientas software utilizadas para el desarrollo de este proyecto son las siguientes:

- Microsoft Office 2007: Documentación
- Eclipse: Desarrollo de la aplicación móvil
- Android ADT: Desarrollo de la aplicación móvil
- Enterprise Architect: Diagramas para la documentación.
- Netbeans: Desarrollo de la aplicación web.
- OmniPlan: Diagramas de Gantt
- Mockflow: Wireframes
- Adobe Photoshop CS3: Diseños del sistema.
- Tomcat: Servidor de aplicaciones web.
- Mysql Server: Servidor de base de datos.

En cuanto al Hardware necesario para el desarrollo del proyecto fueron necesarios:

- Equipo de desarrollo
- Servidor (para despliegue de la aplicación web)
- Dispositivo móvil para pruebas de la aplicación móvil Android (Samsung Galaxy S2)

Tanto a nivel de software como a nivel de hardware se ha llevado a cabo un estudio de la amortización de estos recursos de forma que se le cobre al cliente únicamente el porcentaje de uso de software y hardware empleado a este proyecto.

Se ha estimado un tiempo de vida de estos recursos de 4 años (1152 días). El tiempo empleado en la realización de este proyecto es de 183 días. Por tanto si dividimos los días que se ha tardado en realizar este proyecto entre el número de días de vida de cada recurso y lo multiplicamos por 100 tendremos el porcentaje de amortización.

$183/1152 = 0.15885416666 * 100 = 0.15885416666$, aplicaremos por tanto un 16% de amortización.

Ítem	SubItem	Concepto	Precio Unitario	Cantidad	Coste Total	Coste Ítem
01		Costes Software		(Unidades)		179 €
	001	S.O. Windows 7 (*)	0,00 €	0,16	0,00 €	
	002	Microsoft Office 2007 Professional	267,00 €	0,16	32,04 €	

Presupuesto

003	Eclipse	0,00 €	0,16	0,00 €
004	Netbeans	0,00€	0,16	0,00€
005	Android ADT	0,00 €	0,16	0,00 €
006	Enterprise Architect	135,00 €	0,16	21,26 €
007	GanttProyect	0,00 €	0,16	0,00 €
008	Adobe Photoshop Cs3	985,90 €	0,16	157,74 €
02	Costes Hardware		(Unidades)	376.84 €
001	Equipo	920,00 €	0,16	147,20 €
002	Servidor	1020,00€	0,16	163,20 €
003	Samsung Galaxy s2	415 €	0,16	66,44 €
			TOTAL:	555,84 €

11.2 Costes para el cliente

Además de los costes de material y herramientas para el desarrollo del proyecto es necesario que el cliente abone un coste conforme al trabajo realizado durante todo el desarrollo.

El esfuerzo se ve reflejado en horas ya que no todos los días se han invertido el mismo número de horas de trabajo y eso sería injusto para el cliente.

El precio por cada hora trabajada depende de la tarea que se realice, ya que no cobra lo mismo un programador que un analista o un diseñador. En la propia tabla se especifica el precio que se paga por cada función a la hora, variando desde 15€ a 30€ la hora.

A continuación se muestran los costes del trabajo que se ha desarrollado a lo largo del proyecto:

Ítem	SubItem	Concepto	Precio Unitario	Cantidad	Coste Total	Coste Ítem
03		Recursos Humanos		(Horas)		17.174,00 €
	001	Análisis del sistema	30,00 €	62	1.860,00 €	
	002	Diseño del sistema	30,00 €	154	4.620,00 €	
	003	Implementación del sistema	27,00 €	292	7.884,00 €	
	004	Realización de pruebas	15,00 €	94	1.410,00 €	
	005	Documentación	20,00 €	70	1.400,00 €	
	006	Preparación de presentación	20,00 €	8	160,00 €	
04		Otros				1.080,00 €
	001	Extra por trabajo en fin de semana o festivo	15,00 €	72	1.080,00 €	
					Subtotal	18.254,00€
					IVA (21%)	3.833,34€
					TOTAL	22.087,34€

11.2.1 Presupuesto final

Ítem	SubItem	Concepto	Precio Unitario	Cantidad	Coste Total	Coste Ítem
01		Costes Software		(Unidades)		179 €
	001	S.O. Windows 7 (*)	0,00 €	0,16	0,00 €	
	002	Microsoft Office 2007 Professional	267,00 €	0,16	32,04 €	

Presupuesto

003	Eclipse	0,00 €	0,16	0,00 €	
004	Netbeans	0,00€	0,16	0,00€	
005	Android ADT	0,00 €	0,16	0,00 €	
006	Enterprise Architect	135,00 €	0,16	21,26 €	
007	GanttProyect	0,00 €	0,16	0,00 €	
008	Adobe Photoshop Cs3	985,90 €	0,16	157,74 €	
02	Costes Hardware		(Unidades)	376,84 €	
001	Equipo	920,00 €	0,16	147,20 €	
002	Servidor	1020,00€	0,16	163,20 €	
003	Samsung Galaxy s2	415 €	0,16	66,44 €	
03	Recursos Humanos		(Horas)	17.174,00 €	
001	Análisis del sistema	30,00 €	62	1.860,00 €	
002	Diseño del sistema	30,00 €	154	4.620,00 €	
003	Implementación del sistema	27,00 €	292	7.884,00 €	
004	Realización de pruebas	15,00 €	94	1.410,00 €	
005	Documentación	20,00 €	70	1.400,00 €	
006	Preparación de presentación	20,00 €	8	160,00 €	
04	Otros			1.080,00 €	
001	Extra por trabajo en fin de semana o festivo	15,00 €	72	1.080,00 €	
				Subtotal	18.809,84€
				IVA (21%)	3.950,06€
				TOTAL	22.759,9€

Capítulo 12. Referencias Bibliográficas

12.1 Referencias en Internet

[The jQuery Foundation] “jQuery” . <http://jquery.com> 2013.

[Apache Struts 2] “The Apache Software Foundation, About Apache Struts2” . <http://struts.apache.org/release/2.0.x/docs/home.html> 2000-2013.

[MapBox] “MapBox Developers API” . <http://www.mapbox.com/developers/> 2013.

[Android Developers] “Google Inc” . <http://developer.android.com/develop/index.html> 2013.

[Bootstrap] “@mdo, @fat” . <http://twitter.github.io/bootstrap/> 2013.

[AddThis] “AddThis” . <http://www.addthis.com/> 2013.

[Facebook Developers] “Facebook” . <https://developers.facebook.com/> 2013.

[Icon Finder] <http://www.iconfinder.com/> 2013.

Capítulo 13. Apéndices

13.1 Glosario y Diccionario de Datos

- **Android:** Sistema operativo orientado a dispositivos móviles basado en una versión modificada del núcleo Linux.
- **API:** Conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
- **Base de datos:** es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.
- **Diagrama:** Forma visual de representar diversos tipos de información, ya sea de un componente u objeto concreto del sistema, como de su funcionamiento o iteración con el resto de componentes.
- **Framework:** Conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.
- **Iconografía:** Descripción de las temáticas de las imágenes y también el tratado o colección de éstas.
- **Java:** Lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90.
- **Linux:** Sistema operativo libre tipo Unix. Es usualmente utilizado junto a las herramientas GNU como interfaz entre los dispositivos de hardware y los programas usados por el usuario para manejar un computador.
- **Modelo vista controlador:** patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.
- **OpenGL:** Conjunto de especificaciones estándar que definen una API multilenguaje y multiplataforma para escribir aplicaciones o juegos que producen gráficos en 3D.
- **Sistema de gestión de base de datos:** permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada.
- **Streaming:** Distribución de audio o vídeo por internet. La palabra "streaming" se refiere a que se trata de una corriente continua (sin interrupción). El usuario puede escuchar o ver en el momento que quiera. No es necesario estar suscrito para escuchar o ver.
- **Tablet PC:** Dispositivo mezcla entre un ordenador portátil y una PDA, en el que se puede escribir y manejar a través de una pantalla táctil.
- **Tethering:** El Tethering es el proceso por el cual un dispositivo móvil es la puerta de acceso para dar internet a otros dispositivos.

13.2 Contenido Entregado en el CD-ROM

13.2.1 Contenidos y estructura

Directorio	Contenido
<i>./ Directorio raíz del CD</i>	Contiene un fichero leeme.txt explicando toda esta estructura.
<i>./Mapit servidor</i>	Contiene toda la estructura de directorios del proyecto para desarrollo de la parte del servidor. Para ver la estructura interna de este directorio consultar la tabla del apartado “Estructura del directorio MapitServer”
<i>./Mapit cliente</i>	Contiene toda la estructura de directorios del proyecto para desarrollo de la parte del cliente (aplicación Android). Para ver la estructura interna de este directorio consultar la tabla del apartado “Estructura del directorio Mapit”
<i>./instalacion</i>	Ficheros utilizados para la instalación del proyecto.
<i>./documentacion</i>	Contiene toda la documentación asociada al proyecto en formato docx y PDF.
<i>./documentacion/img</i>	Directorio que contiene las imágenes utilizadas en la documentación.
<i>./documentacion/diagramas</i>	Ficheros que genera la herramienta Enterprise Architect para la creación de los diagramas pres.
<i>./presentacion</i>	Directorio que contiene la presentación de la defensa del proyecto.

13.2.1.1 Estructura del directorio Mapit

Directorio	Contenido
<i>./ Directorio raíz de “desarrollo”</i>	Contiene los ficheros de proyecto del IDE utilizado.
<i>./src</i>	Ficheros Java del proyecto.
<i>./bin</i>	Directorio donde se guardan los ficheros compilados .class.
<i>./gen</i>	Directorio guardan los archivos generados por la aplicación, como el archivo de recursos.

13.2.1.2 Estructura del directorio MapitServer

Directorio	Contenido
./ Directorio raíz de "desarrollo"	Contiene los ficheros de proyecto del IDE utilizado.
./src	Ficheros Java del proyecto.
./bin	Directorio donde se guardan los ficheros compilados .class.
./dist	Directorio donde se guardan los archivos de despliegue de la aplicación. Aquí se encontrará el fichero .war para el despliegue de la aplicación en un servidor web.
./web	Contiene los ficheros relacionados con la presentación de la web: css, js, jsp, imágenes...

13.2.2 Código Ejecutable e Instalación

Para instalar correctamente el sistema consultar con el manual de instalación.

Para desplegar la aplicación web una vez que se ha configurado correctamente el servidor, se deberá copiar el archivo ROOT.war que se encuentra en la carpeta MapitServer/dist en la carpeta webapps del Tomcat. Esto desplegará automáticamente la aplicación en el servidor. Para acceder a la página web únicamente se deberá acceder a la página <http://localhost>

Si se desea acceder a la aplicación web de pruebas, esta se encuentra instalada en <http://mapit.es>

Para instalar la aplicación de android el fichero .apk se encuentra en la carpeta Mapit/bin . Únicamente se deberá enviar al dispositivo en el que se quiera instalar y hacer click sobre este archivo.

13.2.3 Ficheros de Configuración

Para configurar la aplicación, como se indica en el manual del programador, se deberán configurar los siguientes ficheros:

- Aplicación móvil: WSController; en esta clase se encuentra la ruta al servidor para poder comunicarse con él y enviar las fotos de un mapa.
- Aplicación web: Configuration.properties; en este fichero se encuentran los datos relativos a la url de la aplicación y a la carpeta donde se guardarán las imágenes recibidas de la aplicación móvil.

13.3 Código Fuente

En este apartado se expondrán las partes del sistema más importantes, ya que de no ser así sería demasiado extenso.

13.3.1 Aplicación móvil:

13.3.1.1 WSController:

```

package controller;

import java.io.File;
import java.nio.charset.Charset;
import java.util.ArrayList;
import java.util.List;

import model.Archivo;
import model.Proyecto;

import org.springframework.core.io.FileSystemResource;
import org.springframework.core.io.Resource;
import org.springframework.http.HttpEntity;
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpMethod;
import org.springframework.http.MediaType;
import org.springframework.http.client.HttpComponentsClientHttpRequestFactory;
import org.springframework.http.converter.HttpMessageConverter;
import org.springframework.http.converter.json.GsonHttpMessageConverter;
import org.springframework.util.LinkedMultiValueMap;
import org.springframework.util.MultiValueMap;
import org.springframework.web.client.RestTemplate;

import android.app.Activity;
import android.content.Context;
import android.database.Cursor;
import android.net.Uri;
import android.provider.MediaStore;

import com.google.gson.Gson;
import com.google.gson.GsonBuilder;

public class WSController {

    static WSController CONTROLLER;
    static String API_IP = "http://mapit.es/api/";
    // static String API_IP = "http://192.168.1.17:8080/api/";

    /**
     * Inicializa la plantilla utilizada para los servicios web a√+adiendole los
     * adapters necesarios de algunos atributos.
     * @param rest Plantilla RestTemplate
     * @return Gson utilizado para convertir JSON en objetos
     */
    public Gson inicialiceTemplate(RestTemplate rest) {

        GsonHttpMessageConverter converter = new GsonHttpMessageConverter();
        Charset utf8 = Charset.forName("UTF-8");

        List<MediaType> supportedMediaTypes = new ArrayList<MediaType>();
        supportedMediaTypes.add(new MediaType("application", "json", utf8));
    }
}

```

```

        List<HttpMessageConverter<?>> mc = rest.getMessageConverters();

        converter.setSupportedMediaTypes(supportedMediaTypes);

        GsonBuilder gsonBuilder = new GsonBuilder();
//        gsonBuilder.registerTypeAdapter(AttributeImpl.class, new
GsonAttributeAdapter());
//        gsonBuilder.registerTypeAdapter(Date.class, new GsonDateAdapter());
//        gsonBuilder.registerTypeAdapter(Restriction.class, new
GsonRestrictionAdapter());
//        gsonBuilder.registerTypeAdapter(TypeItem.class, new
GsonTypeItemAdapter());
//

        Gson gson = gsonBuilder.create();

        mc.add(converter);
        converter.setGson(gson);

        rest.setMessageConverters(mc);

        return gson;
    }

    public RestTemplate getDefaultRestTemplate(){
        HttpComponentsClientHttpRequestFactory requestFactory = new
HttpComponentsClientHttpRequestFactory();

//        requestFactory.setConnectTimeout(20000);
        return new RestTemplate(requestFactory);

//
    }

    public String crearProyecto(Context context, Proyecto p){
        long idUsuario = UserController.getController().getIdUsuario(context);
        String password = p.getPassword();
        if(password.equals(""))
            password = "***VACIA**";
        String url =
API_IP+"user/"+idUsuario+"/create/project/"+p.getNombre()+"/"+password;

        RestTemplate rest = getDefaultRestTemplate();
        Gson gson = inicialiceTemplate(rest);

        try{

            String response = rest.getForObject(url, String.class);

            return response;

        }catch(UnsupportedOperationException ex){
            System.err.println(ex.getLocalizedMessage());
            return null;
        }catch (Exception e) {
            System.err.println(e.getLocalizedMessage());
            return null;
        }
    }

    public long iniciarSesion(Context context, String email, String password){
        String token = UserController.getController().generarToken(email,
password);
        String url = API_IP+"user/"+email+"/login/"+token;
        System.out.println("INICIAR SESION");

        RestTemplate rest = getDefaultRestTemplate();
        Gson gson = inicialiceTemplate(rest);

        try{

```

```

        String response = rest.getForObject(url, String.class);

        long id = Long.parseLong(response);

        return id;

    } catch (UnsupportedOperationException ex) {
        System.err.println(ex.getLocalizedMessage());
        return -1;
    } catch (Exception e) {
        System.err.println(e.getLocalizedMessage());
        return -1;
    }
}

public String getUrlProyecto(Context context, Proyecto p) {
    String url = API_IP+"project/"+p.getServer_id()+"/link";

    RestTemplate rest = getDefaultRestTemplate();
    Gson gson = inicialiceTemplate(rest);

    try {

        String response = rest.getForObject(url, String.class);

        return response;

    } catch (UnsupportedOperationException ex) {
        System.err.println(ex.getLocalizedMessage());
        return null;
    } catch (Exception e) {
        System.err.println(e.getLocalizedMessage());
        return null;
    }
}

public String eliminarArchivo(Context context, Archivo a) {
    String url =
API_IP+"user/"+UserController.getController().getIdUsuario(context)+
        "/project/"+a.getId_proyecto()+"/delete/"+a.getServerId();

    RestTemplate rest = getDefaultRestTemplate();
    Gson gson = inicialiceTemplate(rest);

    try {

        String response = rest.getForObject(url, String.class);
        return response;

    } catch (UnsupportedOperationException ex) {
        System.err.println(ex.getLocalizedMessage());
        return null;
    } catch (Exception e) {
        System.err.println(e.getLocalizedMessage());
        return null;
    }
}

public Long enviarArchivo(Context context, Archivo archivo) {
    long idUsuario = UserController.getController().getIdUsuario(context);
    String url =
API_IP+"user/"+idUsuario+"/project/"+archivo.getId_proyecto()+"/media/attach";

    RestTemplate rest = getDefaultRestTemplate();
    Gson gson = inicialiceTemplate(rest);

    File cacheDirectory = context.getFilesDir();

    // String path = getPathForArchivo((Activity)context,
    archivo.getArchivo(), archivo.getTipo());

    String path = archivo.getArchivo();

```

```

        if(!path.equals("")){
            File tmpFile = new File(path);
            Resource file = new FileSystemResource(tmpFile);

            System.out.println("FILE: "+path);
            String[] pathtokens = path.split("/");
            pathtokens = pathtokens[pathtokens.length-1].split("\\.");
            String ext = pathtokens[pathtokens.length-1];

            System.out.println("EXTENSION: "+ext);
            MultiValueMap<String, Object> formData = new
LinkedMultiValueMap<String, Object>();
            formData.add("file", file);
            formData.add("nombre", archivo.getNombre());
            formData.add("latitud", archivo.getLatitud()+"");
            formData.add("longitud", archivo.getLongitud()+"");
            formData.add("direccion", archivo.getDireccion());
            formData.add("fecha", archivo.getFecha().getTime()+"");
            formData.add("descripcion", archivo.getDescripcion());
            formData.add("tipo", archivo.getTipo()+"");
            formData.add("ext", ext);

            HttpHeaders requestHeaders = new HttpHeaders();
            requestHeaders.setContentType(MediaType.MULTIPART_FORM_DATA);
            HttpEntity<MultiValueMap<String, Object>> requestEntity = new
HttpEntity<MultiValueMap<String, Object>>(formData, requestHeaders);

            try{

//                String responseString = rest.postForObject(url, user,
String.class);

                HttpEntity<String> response = rest.exchange(url,
HttpMethod.POST, requestEntity, String.class);

                String responseString = response.getBody();
                System.out.println("RESPONSE: "+responseString);

                return Long.parseLong(responseString);

            }catch(UnsupportedOperationException ex){
                System.err.println(ex.getLocalizedMessage());
                return null;
            }catch (Exception e) {
                System.err.println(e.getLocalizedMessage());
                return null;
            }
        }else{
            return null;
        }
    }

    public String getPathForArchivo(Activity a, String file,int tipo){
        String path="";
        try{
            Uri contentUri = Uri.parse(file);
            String[] proj = { MediaStore.Images.Media.DATA };
            Cursor cursor = a.managedQuery(contentUri, proj, null, null, null);
            int column_index = 0;
            if(tipo == Archivo.FOTO)
                cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
            else if(tipo == Archivo.VIDEO)
                cursor.getColumnIndexOrThrow(MediaStore.Video.Media.DATA);
            else
                cursor.getColumnIndexOrThrow(MediaStore.Audio.Media.DATA);

            cursor.moveToFirst();
            path=cursor.getString(column_index);
        }catch(Exception e){

        }

        return path;
    }
}

```

```
public static WSController getController(){
    if(CONTROLLER==null){
        CONTROLLER = new WSController();
    }
    return CONTROLLER;
}
}
```

13.3.1.2 UserController

```
package controller;

import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Calendar;

import android.content.Context;
import android.content.SharedPreferences;
import android.content.SharedPreferences.Editor;

public class UserController {

    private static UserController CONTROLLER;

    public Boolean iniciarSesion(Context c, String email, String password) {
        SharedPreferences settings = c.getSharedPreferences("preferencias",
            Context.MODE_PRIVATE);
        Boolean autenticado = false;

        long idUsuario = WController.getController().iniciarSesion(c, email,
password);

        if (idUsuario != -1) {
            autenticado = true;
            Editor editor = settings.edit();
            editor.putString("email", email);
            editor.putString("pass", password);
            editor.putLong("idUsuario", idUsuario);
            editor.commit();
        }

        return autenticado;
    }

    public void cerrarSesion(Context c) {
        SharedPreferences settings = c.getSharedPreferences("preferencias",
            Context.MODE_PRIVATE);

        Editor editor = settings.edit();
        editor.putString("email", null);
        editor.putString("pass", null);
        editor.putLong("idUsuario", -1);
        editor.commit();
    }

    public long getIdUsuario(Context c) {
        SharedPreferences settings = c.getSharedPreferences("preferencias",
            Context.MODE_PRIVATE);
        return settings.getLong("idUsuario", -1);
    }

    public String getEmailUsuario(Context c) {
        SharedPreferences settings = c.getSharedPreferences("preferencias",
            Context.MODE_PRIVATE);
        return settings.getString("email", null);
    }

    public Boolean usuarioLogueado(Context c) {
        long id = CONTROLLER.getIdUsuario(c);

        if (id != -1)
            return true;
        else
            return false;
    }

    public String generarToken(String email, String password) {
        String token = "";

        Calendar c = Calendar.getInstance();
```

```

        int dayOfMonth = c.get(Calendar.DAY_OF_MONTH);
        int hourOfDay = c.get(Calendar.HOUR_OF_DAY);
        int minuteOfDay = c.get(Calendar.MINUTE);

        token = email + md5(password) + dayOfMonth + "/" + hourOfDay + "/"
                + minuteOfDay;
        token = md5(token);

        return token;
    }

    public static final String md5(final String s) {
        try {
            // Create MD5 Hash
            MessageDigest digest = java.security.MessageDigest
                .getInstance("MD5");
            digest.update(s.getBytes());
            byte messageDigest[] = digest.digest();

            // Create Hex String
            StringBuffer hexString = new StringBuffer();
            for (int i = 0; i < messageDigest.length; i++) {
                String h = Integer.toHexString(0xFF & messageDigest[i]);
                while (h.length() < 2)
                    h = "0" + h;
                hexString.append(h);
            }
            return hexString.toString();

        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        return "";
    }

    public static UserController getController() {
        if (CONTROLLER == null) {
            CONTROLLER = new UserController();
        }

        return CONTROLLER;
    }
}

```

13.3.1.3 GestorBaseDatos

```

package persistence;

import java.util.ArrayList;
import java.util.Date;
import java.util.LinkedList;
import java.util.List;

import model.Archivo;
import model.Proyecto;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteQueryBuilder;

public class GestorBaseDatos {

    private final static String BD_NOMBRE = "mytravels";
    private final static String BD_TABLA_PROYECTO = "proyectos";
    private final static String BD_TABLA_ARCHIVOS = "archivos";

    ArrayList<String> salida = new ArrayList<String>();

    public static void create(Context context){

```

```

        SQLiteDatabase myDB = context.openOrCreateDatabase(BD_NOMBRE, 1, null);

        /* Creamos la tabla de usuarios en la base de datos.
        * En caso de que existiera previamente no da error ('IF NOT EXISTS'). */

        myDB.execSQL("CREATE TABLE IF NOT EXISTS "
            + BD_TABLA_ARCHIVOS + "(id INTEGER PRIMARY KEY autoincrement,
serverId INTEGER, idProyecto INTEGER, nombre VARCHAR, descripcion VARCHAR,
            + "direccion VARCHAR, latitud VARCHAR, longitud VARCHAR, fecha INTEGER,
imagen VARCHAR, tipo INTEGER, borrado INTEGER);");

        myDB.execSQL("CREATE TABLE IF NOT EXISTS "
            + BD_TABLA_PROYECTO + "(id INTEGER PRIMARY KEY
autoincrement, serverId INTEGER, nombre VARCHAR, color INTEGER);");

        if (myDB != null)
            myDB.close();

        System.out.println("CREADA BASE DE DATOS");
    }

    public static Long addArchivoToProyect(Context context, Long id_Proyecto,
Archivo foto){
        SQLiteDatabase myDB = context.openOrCreateDatabase(BD_NOMBRE, 1, null);
        myDB.execSQL("PRAGMA foreign_keys = ON;");

        String tabla=BD_TABLA_ARCHIVOS;

        ContentValues values = new ContentValues();
            values.put("idProyecto", id_Proyecto);
        values.put("nombre", foto.getNombre());
        values.put("descripcion", foto.getDescripcion());
        values.put("direccion", foto.getDireccion());
        values.put("latitud", foto.getLatitud()+"");
        values.put("longitud", foto.getLongitud()+"");
        values.put("fecha", foto.getFecha().getTime());
        values.put("imagen", foto.getArchivo());
        values.put("tipo", foto.getTipo());
        values.put("borrado", 0);
        values.put("serverId", 0);

        Long id=myDB.insert(tabla, null, values);

        if (myDB != null)
            myDB.close();

        return id;
    }

    public static Long addProyecto(Context context, Proyecto p){

        SQLiteDatabase myDB = context.openOrCreateDatabase(BD_NOMBRE, 1, null);
        myDB.execSQL("PRAGMA foreign_keys = ON;");

        String tabla=BD_TABLA_PROYECTO;

        ContentValues values = new ContentValues();
        values.put("nombre", p.getNombre());
        values.put("color", p.getColor());
        values.put("serverId", 0);

        Long id=myDB.insert(tabla, null, values);

        if (myDB != null)
            myDB.close();
    }

```



```

        return id;
    }

    public static List<Archivo> getArchivosByProyecto(Context context, Long id_Proyecto) {

        SQLiteDatabase myDB = context.openOrCreateDatabase(BD_NOMBRE, 1, null);
        myDB.execSQL("PRAGMA foreign_keys = ON;");
        String tabla=BD_TABLA_ARCHIVOS;

        List<Archivo> fotos=new LinkedList<Archivo>();

        /* Ahora hacemos una select sobre las columnas 'nombre' y 'edad'
        * para todos los usuarios mayores de 18 años.
        * Sólo mostraremos los primeros 7 usuarios. */

        String[] columns =
        {"id","idProyecto","nombre","descripcion","direccion","latitud","longitud","fecha","imagen",
        "en","tipo","serverId","borrado"};
        Cursor c = myDB.query(tabla, columns, "idProyecto = '"+id_Proyecto+"' and
        borrado = 0", null, null, null, null, null);

        /* Nos aseguramos de que se haya creado el cursor. */
        if (c != null) {
            /* Obtenemos el índice de las columnas que vamos a utilizar. */
            int columnaIdIndice = c.getColumnIndexOrThrow("id");
            int columnaIdProyecto = c.getColumnIndexOrThrow("idProyecto");
            int columnaNombreIndice = c.getColumnIndexOrThrow("nombre");
            int columnaDescripcionIndice = c.getColumnIndexOrThrow("descripcion");
            int columnaDireccionIndice = c.getColumnIndexOrThrow("direccion");
            int columnaLatitudIndice = c.getColumnIndexOrThrow("latitud");
            int columnaLongitudIndice = c.getColumnIndexOrThrow("longitud");
            int columnaFechaIndice = c.getColumnIndexOrThrow("fecha");
            int columnaImagenIndice = c.getColumnIndexOrThrow("imagen");
            int columnaTipoIndice = c.getColumnIndexOrThrow("tipo");
            int columnaBorradoIndice = c.getColumnIndexOrThrow("borrado");
            int columnaServerIdIndice = c.getColumnIndexOrThrow("serverId");

            /* Creamos un entero a modo de contador
            * para listar los nombres por pantalla. */
            int i = 1;
            /* El cursor devuelto est. posicionado antes de la primera entrada.
            * Avanzamos una posición. */
            while (c.moveToNext()) {
                /* Obtenemos el valor de la columna nombre */

                Archivo foto=new Archivo(c.getString(columnaNombreIndice),
                c.getString(columnaDescripcionIndice),
                c.getString(columnaDireccionIndice),
                Double.valueOf(c.getString(columnaLatitudIndice)),
                Double.valueOf(c.getString(columnaLongitudIndice)),
                new Date(c.getLong(columnaFechaIndice)),
                c.getString(columnaImagenIndice),
                c.getInt(columnaTipoIndice));

                foto.setId((long) c.getInt(columnaIdIndice));
                foto.setId_proyecto((long) c.getInt(columnaIdProyecto));
                foto.setServerId((long) c.getInt(columnaServerIdIndice));
                foto.setBorrado(c.getInt(columnaBorradoIndice));
                fotos.add(foto);

            }

            if (myDB != null)
                myDB.close();

            return fotos;
        }

        public static List<Archivo> getArchivosByProyectoToSincro(Context context, Long
        id_Proyecto) {

```

```

    SQLiteDatabase myDB = context.openOrCreateDatabase(BD_NOMBRE, 1, null);
    myDB.execSQL("PRAGMA foreign_keys = ON;");
    String tabla=BD_TABLA_ARCHIVOS;

    List<Archivo> fotos=new LinkedList<Archivo>();

    /* Ahora hacemos una select sobre las columnas 'nombre' y 'edad'
    * para todos los usuarios mayores de 18 años.
    * Sólo mostraremos los primeros 7 usuarios. */

    String[] columns =
    {"id","idProyecto","nombre","descripcion","direccion","latitud","longitud","fecha","imagen",
    "tipo","serverId","borrado"};
    Cursor c = myDB.query(tabla, columns, "idProyecto = '"+id_Proyecto+"'", null,
    null, null, null, null);

    /* Nos aseguramos de que se haya creado el cursor. */
    if (c != null) {
        /* Obtenemos el índice de las columnas que vamos a utilizar. */
        int columnaIdIndice = c.getColumnIndexOrThrow("id");
        int columnaIdProyecto = c.getColumnIndexOrThrow("idProyecto");
        int columnaNombreIndice = c.getColumnIndexOrThrow("nombre");
        int columnaDescripcionIndice = c.getColumnIndexOrThrow("descripcion");
        int columnaDireccionIndice = c.getColumnIndexOrThrow("direccion");
        int columnaLatitudIndice = c.getColumnIndexOrThrow("latitud");
        int columnaLongitudIndice = c.getColumnIndexOrThrow("longitud");
        int columnaFechaIndice = c.getColumnIndexOrThrow("fecha");
        int columnaImagenIndice = c.getColumnIndexOrThrow("imagen");
        int columnaTipoIndice = c.getColumnIndexOrThrow("tipo");
        int columnaBorradoIndice = c.getColumnIndexOrThrow("borrado");
        int columnaServerIdIndice = c.getColumnIndexOrThrow("serverId");

        /* Creamos un entero a modo de contador
        * para listar los nombres por pantalla. */
        int i = 1;
        /* El cursor devuelto est. posicionado antes de la primera entrada.
        * Avanzamos una posición. */
        while (c.moveToNext()){
            /* Obtenemos el valor de la columna nombre */

            Archivo foto=new Archivo(c.getString(columnaNombreIndice),
                c.getString(columnaDescripcionIndice),
                c.getString(columnaDireccionIndice),
                Double.valueOf(c.getString(columnaLatitudIndice)),
                Double.valueOf(c.getString(columnaLongitudIndice)),
                new Date(c.getLong(columnaFechaIndice)),
                c.getString(columnaImagenIndice),
                c.getInt(columnaTipoIndice));

            foto.setId((long) c.getInt(columnaIdIndice));
            foto.setId_proyecto((long) c.getInt(columnaIdProyecto));
            foto.setServerId((long) c.getInt(columnaServerIdIndice));
            foto.setBorrado(c.getInt(columnaBorradoIndice));
            fotos.add(foto);

        }
    }

    if (myDB != null)
        myDB.close();

    return fotos;
}

public static Archivo getArchivoById(Context context,Long id){

    SQLiteDatabase myDB = context.openOrCreateDatabase(BD_NOMBRE, 1, null);
    myDB.execSQL("PRAGMA foreign_keys = ON;");
    String tabla=BD_TABLA_ARCHIVOS;

    Archivo foto=null;

    /* Ahora hacemos una select sobre las columnas 'nombre' y 'edad'
    * para todos los usuarios mayores de 18 años.
    * Sólo mostraremos los primeros 7 usuarios. */

```

```

String[] columns =
{"id","idProyecto","nombre","descripcion","direccion","latitud","longitud","fecha","imagen",
"tipo","serverId","borrado"};
Cursor c = myDB.query(tabla, columns, "id = '"+id+"' ", null, null, null, null,
null);

/* Nos aseguramos de que se haya creado el cursor. */
if (c != null) {
/* Obtenemos el índice de las columnas que vamos a utilizar. */
int columnaIdIndice = c.getColumnIndexOrThrow("id");
int columnaIdProyecto = c.getColumnIndexOrThrow("idProyecto");
int columnaNombreIndice = c.getColumnIndexOrThrow("nombre");
int columnaDescripcionIndice = c.getColumnIndexOrThrow("descripcion");
int columnaDireccionIndice = c.getColumnIndexOrThrow("direccion");
int columnaLatitudIndice = c.getColumnIndexOrThrow("latitud");
int columnaLongitudIndice = c.getColumnIndexOrThrow("longitud");
int columnaFechaIndice = c.getColumnIndexOrThrow("fecha");
int columnaImagenIndice = c.getColumnIndexOrThrow("imagen");
int columnaTipoIndice = c.getColumnIndexOrThrow("tipo");
int columnaBorradoIndice = c.getColumnIndexOrThrow("borrado");
int columnaServerIdIndice = c.getColumnIndexOrThrow("serverId");

/* Creamos un entero a modo de contador
* para listar los nombres por pantalla. */
int i = 1;
/* El cursor devuelto est. posicionado antes de la primera entrada.
* Avanzamos una posición. */
while (c.moveToNext()){
/* Obtenemos el valor de la columna nombre */

foto=new Archivo(c.getString(columnaNombreIndice),
c.getString(columnaDescripcionIndice),
c.getString(columnaDireccionIndice),
Double.valueOf(c.getString(columnaLatitudIndice)),
Double.valueOf(c.getString(columnaLongitudIndice)),
new Date(c.getLong(columnaFechaIndice)),
c.getString(columnaImagenIndice),
c.getInt(columnaTipoIndice));

foto.setId((long) c.getInt(columnaIdIndice));
foto.setId_proyecto((long) c.getInt(columnaIdProyecto));

}
}

if (myDB != null)
myDB.close();

return foto;
}

public static void deleteArchivo(Context context, Long id){
SQLiteDatabase myDB = context.openOrCreateDatabase(BD_NOMBRE, 1, null);
myDB.execSQL("PRAGMA foreign_keys = ON;");

String tabla=BD_TABLA_ARCHIVOS;

ContentValues dataToUpdate = new ContentValues();
dataToUpdate.put("borrado", "1");

// myDB.delete(tabla,"id=?", new String[]{id+""});
myDB.update(tabla, dataToUpdate, "id=?", new String[]{id+""});
if (myDB != null)
myDB.close();
}

public static void deleteArchivoFromDB(Context context, Long id){
SQLiteDatabase myDB = context.openOrCreateDatabase(BD_NOMBRE, 1, null);
myDB.execSQL("PRAGMA foreign_keys = ON;");

```

```

String tabla=BD_TABLA_ARCHIVOS;

ContentValues dataToUpdate = new ContentValues();
dataToUpdate.put("borrado", "1");

myDB.delete(tabla,"id=?", new String[]{id+""});

if (myDB != null)
    myDB.close();
}

public static void deleteProyecto(Context context, Long id){
    SQLiteDatabase myDB = context.openOrCreateDatabase(BD_NOMBRE, 1, null);
    myDB.execSQL("PRAGMA foreign_keys = ON;");

    String tabla=BD_TABLA_PROYECTO;

    myDB.delete(tabla,"id=?", new String[]{id+""});

    if (myDB != null)
        myDB.close();

    deleteArchivosProyecto(context, id);
}

public static void updateProyectoServerId(Context context, Long id, Long server_id){
    SQLiteDatabase myDB = context.openOrCreateDatabase(BD_NOMBRE, 1, null);
    myDB.execSQL("PRAGMA foreign_keys = ON;");

    String tabla=BD_TABLA_PROYECTO;

    ContentValues dataToUpdate = new ContentValues();
    dataToUpdate.put("serverId", server_id);

    myDB.update(tabla, dataToUpdate, "id=?", new String[]{id+""});
    if (myDB != null)
        myDB.close();
}

public static void updateArchivoServerId(Context context, Long id, Long server_id){
    SQLiteDatabase myDB = context.openOrCreateDatabase(BD_NOMBRE, 1, null);
    myDB.execSQL("PRAGMA foreign_keys = ON;");

    String tabla=BD_TABLA_ARCHIVOS;

    ContentValues dataToUpdate = new ContentValues();
    dataToUpdate.put("serverId", server_id);

    myDB.update(tabla, dataToUpdate, "id=?", new String[]{id+""});
    if (myDB != null)
        myDB.close();
}

public static void deleteArchivosProyecto(Context context, Long id){
    SQLiteDatabase myDB = context.openOrCreateDatabase(BD_NOMBRE, 1, null);
    myDB.execSQL("PRAGMA foreign_keys = ON;");

    String tabla=BD_TABLA_ARCHIVOS;

    myDB.delete(tabla,"idProyecto=?", new String[]{id+""});

    if (myDB != null)
        myDB.close();
}

public static List<Proyecto> getProyectos(Context context){

```

```

        SQLiteDatabase myDB = context.openOrCreateDatabase(BD_NOMBRE, 1, null);
        myDB.execSQL("PRAGMA foreign_keys = ON;");
        String tabla=BD_TABLA_PROYECTO;

        List<Proyecto> proyectos=new LinkedList<Proyecto>();

        /* Ahora hacemos una select sobre las columnas 'nombre' y 'edad'
        * para todos los usuarios mayores de 18 años.
        * Sólo mostraremos los primeros 7 usuarios. */

        String[] columns = {"id","nombre","color","serverId"};
        Cursor c = myDB.query(tabla, columns, null, null, null, null, null, null);

        /* Nos aseguramos de que se haya creado el cursor. */
        if (c != null) {
            /* Obtenemos el índice de las columnas que vamos a utilizar. */
            int columnaIdIndice = c.getColumnIndexOrThrow("id");
            int columnaNombreIndice = c.getColumnIndexOrThrow("nombre");
            int columnaColorIndice = c.getColumnIndexOrThrow("color");
            int columnaServerIdIndice = c.getColumnIndexOrThrow("serverId");

            /* Creamos un entero a modo de contador
            * para listar los nombres por pantalla. */

            /* El cursor devuelto est. posicionado antes de la primera entrada.
            * Avanzamos una posición. */
            while (c.moveToNext()){
                /* Obtenemos el valor de la columna nombre */
                int valorColumnaId = c.getInt(columnaIdIndice);
                String valorColumnaNombre = c.getString(columnaNombreIndice);
                int valorColumnaColor = c.getInt(columnaColorIndice);
                Proyecto p=new Proyecto(valorColumnaNombre,valorColumnaColor);
                p.setId((long) valorColumnaId);
                p.setServer_id(c.getLong(columnaServerIdIndice));

                proyectos.add(p);

            }

        }

        if (myDB != null)
            myDB.close();

        return proyectos;
    }

    public static Proyecto getProyectoById(Context context, long id){

        SQLiteDatabase myDB = context.openOrCreateDatabase(BD_NOMBRE, 1, null);
        myDB.execSQL("PRAGMA foreign_keys = ON;");
        String tabla=BD_TABLA_PROYECTO;

        /* Ahora hacemos una select sobre las columnas 'nombre' y 'edad'
        * para todos los usuarios mayores de 18 años.
        * Sólo mostraremos los primeros 7 usuarios. */

        String[] columns = {"id","nombre","color","serverId"};
        Cursor c = myDB.query(tabla, columns, "id = "+id, null, null, null, null, null);

        /* Nos aseguramos de que se haya creado el cursor. */
        if (c != null) {
            /* Obtenemos el índice de las columnas que vamos a utilizar. */
            int columnaIdIndice = c.getColumnIndexOrThrow("id");
            int columnaNombreIndice = c.getColumnIndexOrThrow("nombre");
            int columnaColorIndice = c.getColumnIndexOrThrow("color");
            int columnaServerIdIndice = c.getColumnIndexOrThrow("serverId");

            /* Creamos un entero a modo de contador
            * para listar los nombres por pantalla. */

            /* El cursor devuelto est. posicionado antes de la primera entrada.
            * Avanzamos una posición. */
            while (c.moveToNext()){

```

```
        /* Obtenemos el valor de la columna nombre */
        int valorColumnaId = c.getInt(columnaIdIndice);
        String valorColumnaNombre = c.getString(columnaNombreIndice);
        int valorColumnaColor = c.getInt(columnaColorIndice);
        Proyecto p=new Proyecto(valorColumnaNombre,valorColumnaColor);
        p.setId((long) valorColumnaId);
        p.setServer_id(c.getLong(columnaServerIdIndice));

        return p;
    }
}

if (myDB != null)
    myDB.close();

return null;
}
}
```

13.3.1.4 NuevaFoto

```
package es.mapit;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Date;

import model.Archivo;
import persistence.GestorBaseDatos;
import util.ImageDecoder;
import util.TripHelper;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.content.res.Configuration;
import android.graphics.Bitmap;
import android.graphics.Matrix;
import android.location.Location;
```

```
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.CompoundButton;
import android.widget.CompoundButton.OnCheckedChangeListener;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.ImageView.ScaleType;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;
import controller.WSController;

public class NuevaFoto extends Activity{

    private EditText nombre;

    private EditText direccion;
    private EditText descripcion;

    private Button guardar;
    private Button sacarFoto;
    private ImageView foto;
    private LinearLayout ubicacionLayout;
    private CheckBox ubicacionCheckBox;
    private Button elegirUbicacionButton;
    private TextView ubicacionText;
    private Button rotarImagen;

    private Bitmap captureBmp;

    private int SELECT_IMAGE = 237487;
```

```
private int TAKE_PICTURE = 829038;

private int SELECT_LOCATION = 8292122;

private File _photoFile;
private Uri _fileUri;

private String imagen="";

private Long id;
private static final int TAKE_PHOTO_CODE = 1;

private Location ubicacion_manual;

@Override
protected void onCreate(Bundle savedInstanceState) {
    // TODO Auto-generated method stub
    super.onCreate(savedInstanceState);
    requestWindowFeature(Window.FEATURE_NO_TITLE);
    setContentView(R.layout.foto_form);

    getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_STATE_ALWAYS_HIDDEN);

    nombre=(EditText) findViewById(R.id.nombre);
    descripcion=(EditText) findViewById(R.id.descripcion);
    direccion=(EditText) findViewById(R.id.direccion);

    guardar=(Button) findViewById(R.id.guardarFoto);
    sacarFoto=(Button) findViewById(R.id.btFoto);
    foto=(ImageView) findViewById(R.id.imagen);
    ubicacionLayout = (LinearLayout) findViewById(R.id.ubicacion_layout);
    ubicacionCheckBox = (CheckBox) findViewById(R.id.tu_ubicacion_check);
    elegirUbicacionButton = (Button) findViewById(R.id.elegir_localizacion);
    ubicacionText = (TextView) findViewById(R.id.ubicacion_text);
    rotarImagen = (Button) findViewById(R.id.rotarImagen);

    Bundle b=this.getIntent().getExtras();
    id=b.getLong("id");
```



```
        dialogPhoto();

        ubicacionCheckBox.setChecked(true);

        ubicacionCheckBox.setOnCheckedChangeListener(new
OnCheckedChangeListener() {

            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {

                if(!isChecked)
                    elegirUbicacionButton.setVisibility(View.VISIBLE);
                else
                    elegirUbicacionButton.setVisibility(View.GONE);

                actualizarMensajeUbicacion();

            }
        });

        guardar.setOnClickListener(new OnClickListener() {

            public void onClick(View v) {

                Location ubicacion=TripHelper.getUbicacion();

                //                if(ubicacion == null || ubicacion_manual!=null)
                //                    ubicacion = ubicacion_manual;

                if(!ubicacionCheckBox.isChecked())
                    ubicacion = ubicacion_manual;

                if(imagen.equals("")){
                    Toast.makeText(NuevaFoto.this,
R.string.add_foto, Toast.LENGTH_LONG).show();
                    return;
                }
            }
        });
    }
}
```

```
        }  
        if(direccion.getText().toString().equals("")){  
            Toast.makeText(NuevaFoto.this,  
R.string.add_lugar, Toast.LENGTH_LONG).show();  
            return;  
        }  
        if(nombre.getText().toString().equals("")){  
            Toast.makeText(NuevaFoto.this,  
R.string.add_nombre, Toast.LENGTH_LONG).show();  
            return;  
        }  
        if(ubicacion!=null ){  
            Archivo foto=new  
R.string.add_nombre, Toast.LENGTH_LONG).show();  
            descripcion.getText().toString(),  
            direccion.getText().toString(),  
                ubicacion.getLatitude(),  
                ubicacion.getLongitude(),  
                new Date(),  
                imagen,  
                Archivo.FOTO);  
            Long  
idFoto=GestorBaseDatos.addArchivoToProyect(NuevaFoto.this, id, foto);  
            foto.setId(idFoto);  
            foto.setId_proyecto(id);  
            Bundle b=new Bundle();  
            b.putSerializable("foto", foto);  
            Intent intent=new Intent();  
            intent.setClass(NuevaFoto.this,  
ViewFoto.class);  
            intent.putExtras(b);  
            startActivity(intent);  
            finish();  
            TripHelper.setRecargarMapa(true);
```

```
                }else{
                    Toast.makeText(NuevaFoto.this,
R.string.no_ubicacion, Toast.LENGTH_LONG).show();
                }
            }
        });

        sacarFoto.setOnClickListener(new OnClickListener() {

            public void onClick(View v) {
                dialogPhoto();
            }
        });
    }

    private void dialogPhoto(){
        try{
            final CharSequence[] items = {"Seleccionar de la galería", "Hacer una foto"};

            AlertDialog.Builder builder = new AlertDialog.Builder(this);
            builder.setTitle("Seleccionar una foto");
            builder.setItems(items, new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int item) {
                    switch(item){
                        case 0:
                            Intent intent = new Intent(Intent.ACTION_PICK,
android.provider.MediaStore.Images.Media.INTERNAL_CONTENT_URI);
                            intent.setType("image/*");
                            startActivityForResult(intent, SELECT_IMAGE);
                            break;

                        case 1:
                            //                Intent cameraIntent = new
Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
                            ////                cameraIntent.putExtra(android.provider.MediaStore.EXTRA_OUTPUT,
android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
                            //
                            ////                cameraIntent.putExtra(MediaStore.EXTRA_SCREEN_ORIENTATION,
ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
```

```
//          startActivityForResult(cameraIntent, TAKE_PICTURE);

String storageState = Environment.getExternalStorageState();

if(storageState.equals(Environment.MEDIA_MOUNTED)) {

//          String path = Environment.getExternalStorageDirectory().getName()
+ File.separatorChar + "Android/data/" + NuevaFoto.this.getPackageName()+"/" + new
Date().getTime() + ".jpg";

String path =
Environment.getExternalStorageDirectory().getAbsolutePath()+"/MapIt/"+new
Date().getTime() + ".jpg";

    _photoFile = new File(path);

    try {

        if(!_photoFile.exists() == false) {

            _photoFile.getParentFile().mkdirs();

            _photoFile.createNewFile();

        }

    } catch (IOException e) {

//          e.printStackTrace();

        System.out.println("No se pudo crear el archivo:
"+e.getLocalizedMessage());

    }

    System.out.println(path);

    _fileUri = Uri.fromFile(_photoFile);

    Intent camIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE );

    camIntent.putExtra(MediaStore.EXTRA_SCREEN_ORIENTATION,
ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);

    camIntent.putExtra( MediaStore.EXTRA_OUTPUT, _fileUri);

    startActivityForResult(camIntent, TAKE_PICTURE);

} else {

    new AlertDialog.Builder(NuevaFoto.this)

        .setMessage("External Storage (SD Card) is required.\n\nCurrent
state: " + storageState)

        .setCancelable(true).create().show();

}

break;

}
```

```
    }

    });

    AlertDialog alert = builder.create();

    alert.show();

    } catch(Exception e){}

//          startActivityForResult(new
Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE), TAKE_PICTURE);
//
}

public boolean hasImageCaptureBug() {

    // list of known devices that have the bug
    ArrayList<String> devices = new ArrayList<String>();
    devices.add("android-devphone1/dream_devphone/dream");
    devices.add("generic/sdk/generic");
    devices.add("vodafone/vfpioneer/sapphire");
    devices.add("tmobile/kila/dream");
    devices.add("verizon/voles/sholes");
    devices.add("google_ion/google_ion/sapphire");

//          return devices.contains(android.os.Build.BRAND + "/" +
android.os.Build.PRODUCT + "/"
//          + android.os.Build.DEVICE);

    return true;

}

@Override

public void onActivityResult(int requestCode, int resultCode, Intent data) {

    super.onActivityResult(requestCode, resultCode, data);

    actualizarMensajeUbicacion();

    try{

        if(requestCode == SELECT_LOCATION && resultCode == Activity.RESULT_OK){

            Bundle b = data.getExtras();

            Boolean seleccionado = b.getBoolean("ok");

            if(seleccionado){
```

```
        int latitud = (Integer) b.get("lat");
        int longitud = (Integer) b.get("long");

        ubicacion_manual = new Location("loc");
        ubicacion_manual.setLatitude(latitud/ 1E6);
        ubicacion_manual.setLongitude(longitud/ 1E6);

    }

}

}else{
    if (resultCode == Activity.RESULT_OK) {

        Uri selectedImage;
        Bitmap myBitmap;

        if(requestCode == TAKE_PICTURE){
            selectedImage = _fileUri;
            imagen = selectedImage.getPath();
            reducirTamañoImagenGuardando();
        }else{
            selectedImage =
Uri.parse(WsController.getController().getPathForArchivo((Activity)NuevaFoto.this,
data.getDataString(), Archivo.FOTO));

            imagen = selectedImage.getPath();
        }

        myBitmap = ImageDecoder.decodeScaledBitmap(imagen, 400,
400);

        foto.setImageBitmap(myBitmap);
        foto.setScaleType(ScaleType.CENTER_CROP);

        sacarFoto.setVisibility(View.GONE);
        rotarImagen.setVisibility(View.VISIBLE);
    }
}

} catch(Exception e){}
```

```
    }

    public void elegirUbicacion(View v){
        Intent intent = new Intent(NuevaFoto.this, ElegirUbicacion.class);
        startActivityForResult(intent, SELECT_LOCATION);
    }

    @Override
    protected void onResume() {

        actualizarMensajeUbicacion();

        super.onResume();
    }

    public void rotarImagen(View v){
        Bitmap rotatedBitmap = rotate(-90,
        ImageDecoder.decodeScaledBitmap(imagen, 900, 900));
        saveBitmapSD(imagen, rotatedBitmap);
        foto.setImageBitmap(ImageDecoder.decodeScaledBitmap(imagen, 400, 400));
    }

    public Bitmap reducirTamañoImagenGuardando(){
        Bitmap bitmap = ImageDecoder.decodeScaledBitmap(imagen, 900, 900);
        saveBitmapSD(imagen, bitmap);
        return bitmap;
    }

    public Bitmap rotate(float x, Bitmap bitmapOrg) {

        int width = bitmapOrg.getWidth();

        int height = bitmapOrg.getHeight();

        Matrix matrix = new Matrix();
```

```
        matrix.postRotate(x);

        Bitmap resizedBitmap = Bitmap.createBitmap(bitmapOrg, 0, 0,width,
height, matrix, true);

        return resizedBitmap;
    }

    public void saveBitmapSD(String path, Bitmap bitmap){
        ByteArrayOutputStream bytes = new ByteArrayOutputStream();
        bitmap.compress(Bitmap.CompressFormat.JPEG, 80, bytes);

        File f = new File(path);
        if (f.exists()){
            f.delete(); //DELETE existing file
            f = new File(path);
        }
        try {
            f.createNewFile();
            //write the bytes in file
            FileOutputStream fo = new FileOutputStream(f);
            fo.write(bytes.toByteArray());
            // remember close de FileOutput
            fo.close();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    public void actualizarMensajeUbicacion(){
        Location ubicacion = TripHelper.getUbicacion();

        if(ubicacion_manual!=null){

            ubicacionLayout.setBackgroundColor(getResources().getColor(R.color.verde_ok));
            ubicacionText.setText(getString(R.string.ubicacion_manual_ok));
        }
    }
}
```



```

        }else if(ubicacion!=null){

ubicacionLayout.setBackgroundColor(getResources().getColor(R.color.verde_ok));
        ubicacionText.setText(getString(R.string.ubicacion_actual_ok));

        }else{

ubicacionLayout.setBackgroundColor(getResources().getColor(R.color.rojo));
        ubicacionText.setText(getString(R.string.ubicacion_actual_error));

        }

    }

@Override

public void onConfigurationChanged(Configuration newConfig) {

    actualizarMensajeUbicacion();

    super.onConfigurationChanged(newConfig);

}

}

```

13.3.1.5 ExportarProyectoTask

```

package task;

import java.util.List;

import persistence.GestorBaseDatos;

import model.Archivo;
import model.Proyecto;
import android.app.Activity;
import android.database.Cursor;
import android.net.Uri;
import android.os.AsyncTask;
import android.provider.MediaStore;
import controller.WSController;

public class ExportarProyectoTask extends AsyncTask<Object, Integer, String> {

    @Override
    protected String doInBackground(Object... arg0) {
        Activity activity = (Activity) arg0[0];
        Proyecto p = (Proyecto) arg0[1];
        List<Archivo> archivos = (List<Archivo>) arg0[2];

        int totalProgress = archivos.size();
        int count = 1;
        String url_proyecto=null;

        long proyecto_id = p.getServer_id();

        try{

            if(proyecto_id==0){
                proyecto_id
                =Long.parseLong(WSController.getController().crearProyecto(activity, p));

```

```
        GestorBaseDatos.updateProyectoServerId(activity, p.getId(),
proyecto_id);
    }
    p.setServer_id(proyecto_id);
    //Actualizar server id proyecto en BD
    for(Archivo a : archivos){
        a.setId_proyecto(proyecto_id);
        if(a.getBorrado()==0){
            if(a.getServerId()==0){
                Long serverId =
WSController.getController().enviarArchivo(activity,a);
                if(serverId != null){
                    GestorBaseDatos.updateArchivoServerId(activity, a.getId(), serverId);
                }
            }
            }else{
                //ELIMINAR DEL SERVIDOR Y DE LA BASE DE DATOS LOCAL
                WSController.getController().eliminarArchivo(activity, a);
                GestorBaseDatos.deleteArchivoFromDB(activity,
a.getId());
            }
            publishProgress((count*100)/totalProgress);
            count++;
        }
        url_proyecto =
WSController.getController().getUrlProyecto(activity, p);
    }catch(Exception e){
        e.printStackTrace();
    }
    return url_proyecto;
}
}
```

13.3.2 Aplicación web

13.3.2.1 ProyectoServicesImpl

```

package com.mapit.impl.services;

import com.mapit.dao.ProyectoDAO;
import com.mapit.model.Archivo;
import com.mapit.model.Proyecto;
import com.mapit.services.ProyectoServices;
import com.mapit.util.Configuration;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Date;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import java.util.zip.ZipEntry;
import java.util.zip.ZipOutputStream;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

/**
 *
 * @author Diego
 */
@Service
public class ProyectoServicesImpl implements ProyectoServices {

    @Autowired
    ProyectoDAO proyectoDao;

    public ProyectoServicesImpl() {
    }

    public long crearProyecto(Proyecto proyecto) throws Exception {
        String codigoProyecto = crearCodigoProyecto(proyecto);
        if (codigoProyecto != null) {
            proyecto.setCodigo_proyecto(codigoProyecto);
        }
        return proyectoDao.create(proyecto);
    }

    public Proyecto getProyectoByCodigo(String codigo) throws Exception {
        return proyectoDao.getProyectoByCodigo(codigo);
    }

    public String getCodigoProyecto(long id_proyecto) throws Exception {
        Proyecto p = proyectoDao.findbyId(id_proyecto);
        return p.getCodigo_proyecto();
    }

    public String getUrlProyecto(long id_proyecto) throws Exception {
        Proyecto p = proyectoDao.findbyId(id_proyecto);
        String baseUrl =
        Configuration.getInstance().getProperty(Configuration.BASE_URL);
        return baseUrl+"/mapa?cp="+p.getCodigo_proyecto();
    }

    public void addArchivoToProyecto(Archivo archivo, Proyecto proyecto) throws
    Exception {
        throw new UnsupportedOperationException("Not supported yet."); //To change body
    of generated methods, choose Tools | Templates.
    }
}

```

```

    }

    private String crearCodigoProyecto(Proyecto p) {
        Date now = new Date();

        String plaintext = p.getNombre() + p.getId() + p.getUsuario_id()+now.getTime();
        String hashtext = null;
        try {
            MessageDigest m = MessageDigest.getInstance("MD5");
            m.reset();
            m.update(plaintext.getBytes());
            byte[] digest = m.digest();
            BigInteger bigInt = new BigInteger(1, digest);
            hashtext = bigInt.toString(16);

        } catch (NoSuchAlgorithmException ex) {
            Logger.getLogger(ProyectoServicesImpl.class.getName()).log(Level.SEVERE,
null, ex);
        }
        return hashtext;
    }

    public void crearZipProyecto(Proyecto proyecto) throws Exception {

        String root =
Configuration.getInstance().getProperty(Configuration.MEDIA_FOLDER);

        String folder = root +"/project_resources/" + proyecto.getUsuario_id() + "_" +
proyecto.getId();

        final File f = new File(folder+"/mapit.zip");
        final ZipOutputStream out = new ZipOutputStream(new FileOutputStream(f));
        byte[] buffer = new byte[1024];

        for(Archivo a : proyecto.getArchivos()){

            FileInputStream fin = new FileInputStream(root+a.getArchivo());
            String[] trozos = a.getArchivo().split("/");

            out.putNextEntry(new ZipEntry(trozos[trozos.length-1]));

            int length;

            while((length = fin.read(buffer)) > 0)
            {
                out.write(buffer, 0, length);
            }

            out.closeEntry();

            fin.close();
        }

        out.close();
    }

    public Proyecto getProyectoById(long id_proyecto) throws Exception {
        return proyectoDao.findById(id_proyecto);
    }

    public List<Proyecto> getProyectosByUsuarioId(long id_usuario) throws Exception {
        return proyectoDao.getProyectosByUsuarioId(id_usuario);
    }

    public void deleteProyecto(Proyecto proyecto) throws Exception {
        proyectoDao.delete(proyecto);
    }
}

```

13.3.2.2 ArchivoServicesImpl

```
package com.mapit.impl.services;
```

```

import com.mapit.dao.ArchivoDAO;
import com.mapit.model.Archivo;
import com.mapit.model.Proyecto;
import com.mapit.services.ArchivoServices;
import com.mapit.util.Configuration;
import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.Date;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

/**
 *
 * @author Diego
 */
@Service
public class ArchivoServicesImpl implements ArchivoServices {

    @Autowired
    ArchivoDAO archivoDao;

    public long descargarArchivo(Archivo archivo, InputStream uploadedInputStream, long
user_id, long project_id, String extension) throws Exception {

        Date now = new Date();
        //String root = "/usr/local/apache-tomcat-7.0.37/webapps/ROOT";
        String root =
Configuration.getInstance().getProperty(Configuration.MEDIA_FOLDER);
        String folder = root + "/project_resources/" + user_id + "_" + project_id;
        String folderURL = "/project_resources/" + user_id + "_" + project_id;
        String archivoPath = folder + "/" + user_id + project_id + now.getTime() + "." +
extension;
        String archivoURL = folderURL + "/" + user_id + project_id + now.getTime() + "."
+ extension;

        archivo.setId_proyecto(project_id);
        archivo.setId_usuario(user_id);
        archivo.setArchivo(archivoURL);
        long id = crearArchivo(archivo);

        if (id != -1) {
            new File(folder).mkdir();
            File f = new File(archivoPath);
            OutputStream out = new FileOutputStream(f);
            byte buf[] = new byte[1024];
            int len;
            while ((len = uploadedInputStream.read(buf)) > 0) {
                out.write(buf, 0, len);
            }
            out.close();
            uploadedInputStream.close();
        }

        return id;
    }

    public long crearArchivo(Archivo archivo) throws Exception {
        return archivoDao.create(archivo);
    }

    public List<Archivo> getArchivosByProyecto(long id) throws Exception {
        return archivoDao.getArchivosByProyecto(id);
    }

    public void eliminarArchivo(long id, long id_usuario, long id_proyecto) throws
Exception {
        Archivo a = new Archivo();
        a.setId(id);
        a.setId_usuario(id_usuario);
        a.setId_proyecto(id_proyecto);
        archivoDao.delete(a);
    }
}

```

}

13.3.2.3 UsuarioServicesImpl

```

package com.mapit.impl.services;

import com.mapit.dao.UsuarioDAO;
import com.mapit.model.Usuario;
import com.mapit.services.UsuarioServices;
import com.mapit.util.Utils;
import java.util.Calendar;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

/**
 *
 * @author Diego
 */
@Service
public class UsuarioServicesImpl implements UsuarioServices {

    @Autowired
    UsuarioDAO usuarioDao;

    public long crearUsuario(Usuario usuario) throws Exception {
        usuario.setPassword(Utils.md5(usuario.getPassword()));
        return usuarioDao.create(usuario);
    }

    public Usuario getUsuarioByEmailAndPassword(String email, String password) throws
Exception {
        return usuarioDao.findByEmailAndPassword(email, Utils.md5(password));
    }

    public long getUserIdForToken(String email, String token) throws Exception {
        long id = -1;
        Usuario u = usuarioDao.findByEmail(email);

        if(u != null){
            Calendar c = Calendar.getInstance();
            int min = c.get(Calendar.MINUTE);
            String token1 = generarToken(u, min);
            String token2 = generarToken(u, min-1);

            System.out.println("VALIDANDO USUARIO");

            System.out.println("TOKEN Usuario: "+token);

            System.out.println("TOKEN SERVER: "+token1);
            System.out.println("TOKEN SERVER2: "+token2);

            if(token1.equals(token) || token2.equals(token))
                id = u.getId();
        }

        return id;
    }

    private String generarToken(Usuario u, int minuto) {
        String token = "";

        Calendar c = Calendar.getInstance();
        int dayOfMonth = c.get(Calendar.DAY_OF_MONTH);
        int hourOfDay = c.get(Calendar.HOUR_OF_DAY);
        token = u.getEmail() + u.getPassword() + dayOfMonth + "/" + hourOfDay + "/"
            + minuto;
        token = Utils.md5(token);

        return token;
    }
}

```

13.3.2.4 Map_historia.js

```

var map;
var markers = new Array();
var highestZIndex = 0;
var currentInfoBox;
var sections;
var setActive;
var currentActive = 0;

$('document').ready(function() {
    loadMapMarkers();
    loadMedia();
    loadNavigationControls();
    loadEvents();
    config();
});

function config() {

    if ($(window).width() <= 640)
        disable_scroll();
}

function loadMap(geojson) {

    var tiles = mapbox.layer().tilejson({
        tiles: ["http://a.tiles.mapbox.com/v3/fangui.map-ezeswlaz/{z}/{x}/{y}.png"]
    });
    var spots = mapbox.markers.layer()
        .features(geojson)
        .factory(function(f) {
            var el = document.createElement('div');
            el.className = 'marker marker-' + f.properties.id + ' tipo-' +
f.properties.tipo;

            MM.addEvent(el, 'click', function(e) {
                $('.archivo-data').each(function() {
                    var id = $(this).attr('data-id');
                    if (id == f.properties.id) {

                        var offset = $(this).offset().top - 15;
                        $(document).scrollTop(offset);

                    }
                });
            });

            return el;
        });

    map = mapbox.map('map', [tiles, spots]);
    map.eventHandlers[3].remove();
    map.ui.zoomer.add();
    map.ui.zoombox.add();

    sections = document.getElementsByClassName('archivo');

    $(sections).each(function(index, value) {
        var s = value;
        $(spots.markers()).each(function(index, value2) {
            var m = value2;
            var id = m.data.properties.id;
            var section_id = $(s).attr('data-id')
            if (id == section_id && (id != null && section_id != null))
                markers.push(m);
        });
    });
}

```

```

setActive = function(index, ease) {

    if (currentActive !== index) {
        currentActive = index;

        _(sections).each(function(s) {
            s.className = s.className.replace(' active', '');
        });
        _(markers).each(function(m) {
            m.element.className = m.element.className.replace(' active', '');
        });

        $(sections[index]).addClass('active');
        markers[index].element.className += ' active';

        document.body.className = 'section-' + index;

        console.log("POSITION:" + markers[index].location);
        // Ease map to active marker.
        if (index == 0) {
            map.ease.location(markers[index].location).zoom(13).optimal();
        } else if (!ease) {
            map.centerzoom(markers[index].location,
markers[index].data.properties.zoom || 15);
        } else {
            map.ease.location(markers[index].location).zoom(markers[index].data.properties.zoom ||
15).optimal();
        }

    }
    return true;
};

window.onscroll = _(function() {
    // IE 8
    if (window.pageYOffset === undefined) {
        var y = document.documentElement.scrollTop;
        var h = document.documentElement.clientHeight;
    } else {
        var y = window.pageYOffset;
        var h = window.innerHeight;
    }

    if (y === 0)
        return setActive(0, true);

    var memo = 0;
    var buffer = (h * 0.3333);
    var active = _(sections).any(function(el, index) {
        memo += el.offsetHeight;
        return y < (memo - buffer) ? setActive(index, true) : false;
    });

    if (!active)
        setActive(sections.length - 1, true);
}).debounce(10);

map.extent(spots.extent());
}

function setNavigationControls(seccionId) {
    var trozos = seccionId.split("-");
    var current = trozos[1];
    var prev = current;
    var next = current;
    if (current > 0) {
        prev--;
    }
    if (current < sections.length) {
        next++;
    }
}

```



```

    $('#prev_link').attr('href', "#archivo-" + prev);
    $('#next_link').attr('href', "#archivo-" + next);
}

function loadNavigationControls() {
    $('#next_link').click(function() {
        $(sections).each(function() {
            if ($(this).hasClass('active')) {
                var id = $(this).attr('id');
                var trozos = id.split("-");
                var current = trozos[1];
                if (current < sections.length - 1) {
                    current++;
                    setActive(current, true);
                }
                return false;
            }
        });
    });

    $('#prev_link').click(function() {
        $(sections).each(function() {
            if ($(this).hasClass('active')) {
                var id = $(this).attr('id');
                var trozos = id.split("-");
                var current = trozos[1];
                if (current > 0) {
                    current--;
                    setActive(current, true);
                }
                return false;
            }
        });
    });
}

function loadMapMarkers() {

    var geojson = [];

    var primera = true;

    $('.archivo-data').each(function() {

        var archivo = $(this).attr('data-archivo');
        var nombre = $(this).attr('data-nombre');
        var tipo = $(this).attr('data-tipo');
        var lat = $(this).attr('data-lat');
        var long = $(this).attr('data-long');
        var id = $(this).attr('data-id');

        if (primera) {
            geojson.push({"geometry": {"type": "Point", "coordinates": [long, lat]},
                "properties": {"id": 0, "zoom": 13, "tipo": 0}});
            primera = false;
        }

        geojson.push({"geometry": {"type": "Point", "coordinates": [long, lat]},
            "properties": {"id": id, "tipo": tipo}});

    });

    loadMap(geojson);

}

function loadMedia() {
    $(".open-info").fancybox({
        'padding': 0,
        'beforeLoad': function(links, index) {
            var self = $(this.element);
            var media = self.attr('data-media');
            var href = self.attr('href');
            if (href == "#info-popup-video") {
                $("#jquery_jplayer_video").jPlayer({

```

```
        ready: function() {
            $(this).jPlayer("setMedia", {
                m4v: media
            });
        },
        swfPath: "/jplayer",
        supplied: "m4v, oga"
    });
}

if (href == "#info-popup-audio") {
    var myCirclePlayer = new CirclePlayer("#jquery_jplayer_audio",
        {
            oga: media
        }, {
            cssSelectorAncestor: "#cp_container_1"
        });
}

if (href == "#info-popup-foto") {
    $('#image-popup').attr('src', media);
}

});
}

function preventDefault(e) {
    e = e || window.event;
    if (e.preventDefault)
        e.preventDefault();
    e.returnValue = false;
}

function wheel(e) {
    preventDefault(e);
}

function disable_scroll() {
    if (window.addEventListener) {
        window.addEventListener('DOMMouseScroll', wheel, false);
    }
    window.onmousewheel = document.onmousewheel = wheel;
}

function loadEvents() {
    $('#comment_button').click(function() {
        $('.widget-map').removeClass('active');
        $('.facebook-comments').toggleClass('active');

        $('#insert_button').removeClass('btn-success');
        $('#insert_button i').removeClass('icon-white');

        $('#comment_button').toggleClass('btn-success');
        $('#comment_button i').toggleClass('icon-white');
    });

    $('#insert_button').click(function() {
        updateWidgetCode();
        $('.facebook-comments').removeClass('active');
        $('.widget-map').toggleClass('active');

        $('#comment_button').removeClass('btn-success');
        $('#comment_button i').removeClass('icon-white');

        $('#insert_button').toggleClass('btn-success');
        $('#insert_button i').toggleClass('icon-white');
    });

    $('.widget-map-small-input').keyup(function() {
        updateWidgetCode();
    });

    $('.widget-map-check').change(function() {
        if ($(this).is(':checked')) {
```

```
        $('#widget-map-width').prop("disabled", true);
    } else {
        $('#widget-map-width').prop("disabled", false);
    }
    updateWidgetCode();
});

$('.copy-code').zclip({
    path: 'js/ZeroClipboard.swf',
    copy: $('.widget-map-code').html()
});

}

function updateWidgetCode() {

    var width = $('#widget-map-width').val() + "px";
    var height = $('#widget-map-height').val() + "px";

    if ($('#widget-map-check').is(':checked')) {
        width = "100%";
    }

    var code = '<iframe style="width: ' + width + '; height: ' + height + '; border:0;"
src="' + document.URL + '&mode=iframe" ></iframe>';

    $('.widget-map-code').html(code);
}
```