



**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA  
DE TELECOMUNICACIÓN**

INGENIERÍA DE TELECOMUNICACIÓN

+

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

**PROYECTO FIN DE CARRERA**

DISEÑO, DESARROLLO Y DESPLIEGUE DE UN  
SERVICIO WEB DE MONITORIZACIÓN DE  
ACTIVIDADES AL AIRE LIBRE PARA  
DISPOSITIVOS ANDROID

**Autor:** Javier Rodríguez Robles

**Tutor:** Gregorio Robles Martínez

Curso Académico 2012-2013



TÍTULO: *DISEÑO, DESARROLLO Y DESPLIEGUE DE UN SERVICIO WEB DE MONITORIZACIÓN DE ACTIVIDADES AL AIRE LIBRE PARA DISPOSITIVOS ANDROID*

AUTOR: *Javier Rodríguez Robles*

TUTOR: *Gregorio Robles Martínez*

La defensa del presente Proyecto Fin de Carrera se realizó el día      de Junio de 2013; siendo calificada por el siguiente tribunal:

PRESIDENTE:

SECRETARIO

VOCAL

Habiendo obtenido la siguiente calificación:

CALIFICACIÓN:

**Presidente**

**Secretario**

**Vocal**

Fuenlabrada,



## Agradecimientos

En primer lugar me gustaría dar las gracias a todos aquellos que han hecho de mí lo que soy, personas que a lo largo de 23 años han recorrido algún tramo del camino conmigo.

También me gustaría dar las gracias a los profesores, profesores que desde la escuela hasta hoy han sabido despertar en mí la curiosidad por saber más, por ver la razón de las cosas. Especialmente a Gregorio Robles, sin el cual éste proyecto no hubiera sido posible, gracias por hacerlo fácil.

A mi familia, a mis padres y a mis abuelos, y por supuesto a mi hermana Marta, que ahora empieza un camino que justo con estas líneas parece acabar para mí, suerte. Os quiero a todos, gracias por estar ahí.

Hace ya casi dos años, durante mi Erasmus conocí a gentes muy distintas y geniales. Especialmente a tí, Franka.

I would like to say just, thank you. With you, this year here, has been amazing. Now we have the freedom of going wherever we want, now we are going to visit unbelievable places. Thank you for supporting me this last year.

En fin, me despido no sin antes recordar a todos aquellos compañeros de universidad. Compañeros con los que he pasado más de un lustro realizando exámenes, prácticas y por supuesto pasando buenos momentos.

Gracias a todos.



*Siempre se ha creído que existe algo que se llama destino, pero siempre se ha creído también que hay otra cosa que se llama albedrío. Lo que califica al hombre es el equilibrio de esa contradicción.*

Gilbert Keith Chesterton



# Resumen

Hoy en día, gracias al avance tecnológico, una gran mayoría llevamos dispositivos inteligentes en nuestros bolsillos. Estos elementos son capaces de geolocalizarnos, de conectarnos con nuestro entorno y de procesar gran cantidad de información. Las posibilidades que ofrecen son inmensas, tanto desde el punto de vista hardware, como de acceso y compartición de datos en tiempo real.

Partiendo de este hecho, la propuesta es diseñar, desarrollar y posteriormente desplegar una aplicación Web orientada a Android capaz de monitorizar actividades al aire libre. Esta monitorización incluirá la protección inteligente del individuo. A su vez, toda esta información estará disponible en tiempo real para los diferentes usuarios de la aplicación a través de su interfaz Web. Con el fin de alcanzar este objetivo se diseñara una arquitectura Cliente-Servidor, es decir, estará compuesta por una interfaz Web, un servidor y una interfaz Android.

Enumerando brevemente algunas características, el usuario podrá actuar desde dos vertientes claramente diferenciadas. La primera, desde la interfaz Android, que proporcionara la localización, situación y diversos recursos útiles para el usuario y su integridad. La segunda vertiente es representada por la interfaz Web, donde dichos datos podrán ser compartidos y consultados por el propio usuario y por el resto de usuarios de la aplicación. De esta forma, no sólo controlamos la actividad sino que también incluimos un componente social que fomente la actividad entre los usuarios.

En definitiva, el proyecto se centra en crear una aplicación que monitorice y ayude al usuario en su actividad. Para conseguir este objetivo se utilizaran tanto recursos Web como recursos físicos de los SmartPhones (GPS, Sensores, Conexiones, etc).

La aplicación ha sido probada bajo diversas circunstancias con el fin de obtener información heterogénea e independiente sobre cualquier característica sensible de ser mejorada para ofrecer al usuario la mejor experiencia.



# Índice general

<b>Resumen</b>	<b>IX</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Objetivos . . . . .	2
1.2.1. Aspectos funcionales . . . . .	2
1.2.2. Aspectos tecnológicos . . . . .	3
1.2.3. Aspectos educativos . . . . .	4
1.3. Aplicaciones similares . . . . .	5
1.4. Experiencias previas . . . . .	5
1.5. Estimación de la planificación realizada . . . . .	5
1.6. Estructura de este documento . . . . .	6
<b>2. Estado del Arte</b>	<b>9</b>
2.1. Arquitectura de la aplicación . . . . .	9
2.2. Python . . . . .	10
2.3. Django - Framework de desarrollo Web . . . . .	11
2.4. Android . . . . .	12
2.4.1. El sistema operativo Android . . . . .	12
2.4.2. Desarrollo en Android . . . . .	13
2.4.3. Competidores y comunidades . . . . .	13
2.5. HTML Y CSS . . . . .	14

2.5.1.	Hojas de estilo CSS	15
2.5.2.	World Wide Web Consortium W3C	15
2.6.	sqlite	16
2.7.	Realidad Aumentada	16
2.7.1.	Tecnología para Realidad Aumentada	17
2.7.2.	Aplicaciones de la Realidad Aumentada	17
2.7.3.	Próximos avances en Realidad Aumentada	18
2.7.4.	Framework de Realidad Aumentada	18
2.8.	Geolocalización y sensores de movimiento	19
2.9.	Difusión y posicionamiento en buscadores	19
<b>3.</b>	<b>Diseño de la aplicación</b>	<b>21</b>
3.1.	Introducción	21
3.2.	Arquitectura REST en la aplicación	23
3.2.1.	Métodos HTTP y REST	24
3.3.	Aplicación Android	25
3.3.1.	Introducción	25
3.3.2.	Diseño y funcionalidades de la aplicación	25
3.3.3.	Implementación	26
3.4.	Aplicación Servidora	39
3.4.1.	Modelo de datos	40
3.4.2.	Servicio Web	41
3.4.3.	Características y puesta en marcha del servidor	42
3.5.	Interfaz Web	42
3.5.1.	Opciones de la interfaz Web	44
3.6.	Implementación de técnicas de difusión en buscadores	50
3.7.	Problemas afrontados	52
3.8.	Conclusiones y experiencias	53
<b>4.</b>	<b>Resultados</b>	<b>55</b>
4.1.	Primeras pruebas	56
4.2.	Pruebas de usuarios	56

4.3. Reflexiones y soluciones . . . . .	59
<b>5. Conclusiones</b>	<b>61</b>
5.1. Análisis de objetivos . . . . .	61
5.2. Autocrítica . . . . .	63
5.3. Futuras extensiones de la aplicación . . . . .	65
5.4. Recopilación y conclusiones finales . . . . .	66
<b>APÉNDICES</b>	<b>71</b>
<b>A. Manual de Usuario</b>	<b>71</b>
A.1. Introducción y funcionalidad . . . . .	71
A.2. Funcionamiento . . . . .	72
A.2.1. Funcionamiento en dispositivos Android . . . . .	72
A.2.2. Funcionamiento Interfaz Web . . . . .	74
A.3. Requisitos . . . . .	75
<b>B. Encuesta</b>	<b>77</b>



# Lista de Figuras

2.1. <i>Cuota de mercado móvil 2012</i> . . . . .	14
3.1. <i>Esquema Básico de la aplicación</i> . . . . .	22
3.2. <i>Recursos disponibles en la aplicación</i> . . . . .	24
3.3. <i>Estructura de clases por paquetes, Android</i> . . . . .	27
3.4. <i>Layout de la aplicación</i> . . . . .	29
3.5. <i>Layout del menú de la aplicación</i> . . . . .	30
3.6. <i>Layout Estado Mapa con PDIs</i> . . . . .	31
3.7. <i>Esquema de los ejes de aceleración</i> . . . . .	32
3.8. <i>Indicador de rumbo erroneo</i> . . . . .	33
3.9. <i>Indicador de rumbo correcto</i> . . . . .	34
3.10. <i>Layout preferencias de la aplicación</i> . . . . .	35
3.11. <i>Layout de logueo de usuario</i> . . . . .	36
3.12. <i>Diagrama de clases del seguimiento de actividad</i> . . . . .	37
3.13. <i>Layout Estado Seguimiento</i> . . . . .	38
3.14. <i>Esquema relacional del servidor</i> . . . . .	40
3.15. <i>Interfaz Web de la aplicación</i> . . . . .	43
3.16. <i>Interfaz Registro de la aplicación</i> . . . . .	44
3.17. <i>Perfil de usuario de la aplicación</i> . . . . .	45
3.18. <i>Interfaz de alertas de la aplicación</i> . . . . .	46
3.19. <i>Interfaz Mensajes de la aplicación</i> . . . . .	47
3.20. <i>Interfaz Rutas de la aplicación</i> . . . . .	48

3.21. <i>Interfaz Amigos de la aplicación</i> . . . . .	49
3.22. <i>Interfaz Envío Mensajes de la aplicación</i> . . . . .	50
3.23. <i>Estudio de palabras clave de negocio</i> . . . . .	51
3.24. <i>Optimización de metadatos HTML</i> . . . . .	51
A.1. <i>Esquema de la aplicación</i> . . . . .	72
A.2. <i>Menu de la aplicación</i> . . . . .	73
A.3. <i>Interfaz Web de la aplicación</i> . . . . .	75

# Introducción

En este apartado veremos una descripción lo más sencilla, concisa y explicativa posible sobre el proyecto fin de carrera que aquí se presenta. El fin de este primer capítulo es el de exponer algunos puntos que se consideren importantes o relevantes en lo relativo a este proyecto.

## 1.1. Motivación

En la actualidad, una gran mayoría de nosotros utilizamos dispositivos inteligentes de última generación en nuestro día a día. Estos dispositivos son capaces de mucho más que enviar y recibir llamadas o mensajes, hay luz más allá del Whatsapp. Estos nuevos teléfonos inteligentes cuentan con GPS, brújula, mapa y una serie de elementos o sensores de todo tipo que pueden ser extremadamente útiles bajo ciertas condiciones y que en muchos casos aún no han sido aplicados a según qué campos. Al igual que llevamos brújula, navaja o mapa físico, en nuestro macuto, el teléfono móvil también puede ser un gran aliado.

A mí, personalmente, me encantan los deportes al aire libre, como la escalada, montañismo y todo este tipo de deportes donde las características anteriormente expuestas aún no han sido explotadas. Estas características pueden ser de gran utilidad para los usuarios. ¿Qué utilidad podemos ofrecer? Usualmente, por desgracia, vemos en las noticias especializadas del sector información sobre personas que han sufrido accidentes en la práctica de actividades al aire libre.

La idea inicial es crear un servicio que nos permita monitorizar todo este tipo de situaciones, actividades al aire libre donde en ocasiones es posible sufrir accidentes y pueda ser necesaria ayuda externa. Aparte de cubrir las características de protección y monitorización comentadas, creo muy

interesante incluir una componente social en la aplicación. De esta forma conseguimos fomentar la utilización de la herramienta, y por tanto, facilitar la creación de una comunidad entre usuarios que cuenta con intereses comunes, en este caso las actividades al aire libre.

El ámbito de aplicación de este servicio es bastante extenso, puede ser usado por individuos particulares, clubs de deportistas o en la realización de carreras de montaña, entre otras posibles utilidades.

Al contar con una interfaz Web que proporciona información sobre la posición y la ruta de los usuarios esta aplicación, puede ser interesante en multitud de campos donde esta información pueda ser de utilidad.

Otro de los puntos que fomentó la creación y desarrollo de este proyecto ha sido la elección de la tecnología sobre la que esta aplicación va a estar desarrollada, Django (Framework sobre Python) y Android. Por un lado, como bien sabemos, ambos son software libre con todo lo que ello conlleva y la libertad de trabajo que nos aporta. Desde otra perspectiva, sobre todo con Android, por las posibilidades de futuro que aporta este sistema y que muy posiblemente en el futuro será el líder en cuanto a desarrollo móvil se trata. Además, Android es una tecnología sobre la que personalmente he seguido con interés desde sus comienzos allá por el año 2007.

Aunando todo lo anteriormente expuesto, sólo nos faltaba un dato, ya que siempre es necesario una denominación, un nombre para la aplicación, SOFTSPM (SOFTware de Seguimiento y Protección en Montaña).

## 1.2. Objetivos

En este punto vamos a exponer lo más claramente posible lo que queremos conseguir con el desarrollo de esta aplicación. Paso por paso detallaremos los objetivos desde el punto de vista tecnológico, de utilidad y de aprendizaje que la realización de este proyecto va a conllevar. Posteriormente volveremos sobre estos objetivos con el fin de analizar cuáles han sido posibles implementar.

### 1.2.1. Aspectos funcionales

Vamos a describir una serie de objetivos a cumplir en lo referente al funcionamiento que la aplicación debe ofrecer. Estos objetivos giran en torno a la monitorización de actividades al aire libre. Dentro de esta monitorización veamos más en profundidad algunas funcionalidades de este seguimiento:

- Medir aspectos relativos a la actividad desarrollada (velocidad, distancia, posición, etc.).
- Protección de la actividad ante accidentes o imprevistos.
- Uso de realidad aumentada para mostrar puntos de interés o el rumbo.
- Posibilidad de compartir elementos de la ruta realizada con otros usuarios a través de la interfaz Web de la aplicación.
- Fomento de la interactividad de los usuarios de la aplicación mediante la interfaz Web.

### 1.2.2. Aspectos tecnológicos

Dentro de los objetivos, en este apartado veremos los factores tecnológicos que tenemos que cumplir o esperamos cubrir para alcanzar el objetivo final de la aplicación, tal y como hemos expuesto anteriormente. Se integraran conjuntamente para la funcionalidad total de la aplicación los siguientes tres componentes elementales.

#### Servicios sobre Android

Android es uno de los elementos centrales en el desarrollo de este proyecto, ya que se trata de la principal fuente de captura de datos externos. En base a esta premisa hemos de ser capaces de controlar y entender, por un lado, la multitud de sensores con los que cuentan los dispositivos Android y por otro lado el ser capaz de enviar todos estos datos de forma fiable a la parte servidora.

Empecemos enumerando los sensores que es necesario controlar y evaluar para suplir todos los requisitos con los que cuenta la aplicación.

- Sensores GPS - Geolocalización
- Sensores Magnéticos
- Sensores Acelerómetros

En cuanto a las conexiones, se empleará el protocolo HTTP para comunicarse con la parte servidora de forma que se creará la trama adecuada y se adaptará la respuesta a cada caso particular. Con esto conseguimos un entendimiento entre el servidor y la aplicación para dispositivos Android.

Otro de los objetivos Android que serán expuestos y desarrollados más adelante pueden exponerse en líneas generales de la siguiente forma:

- Creación de actividades y servicios optimizados.
- Diseño de una interfaz agradable para proporcionar al usuario la mejor experiencia posible.
- Estudiar la compatibilidad con distintas versiones y servicios Android.

### **Interfaz Web**

La interfaz Web de esta aplicación nos proporciona una visión global de los datos obtenidos por la parte Android así como el componente social de la aplicación. Esta Interfaz Web trata de ser amigable e intuitiva a la vez que útil para optimizar la experiencia de usuario. Para ello se desarrollaran los diseños y estructuras HTML y CSS para conseguir una interfaz aceptable y sencilla.

### **Servidor**

La parte servidora está desarrollada en Python sobre el marco de desarrollo Django. Esta parte es la encargada de gestionar la información enviada por el dispositivo Android, almacenando toda esta información en una base de datos para mediante la tecnología sqlite. El servidor será el núcleo de la aplicación en su conjunto, conectando todos los demás servicios y proporcionando acceso a los usuarios de la misma.

Así mismo responderá y proporcionara el servicio requerido por el usuario sobre la interfaz Web respondiendo a sus requerimientos, este servidor será desplegado sobre un servidor online público para su acceso desde cualquier punto. Su lanzamiento sera presumiblemente sobre tecnología Apache.

### **1.2.3. Aspectos educativos**

Otro de los puntos esenciales de este proyecto es mejorar y adoptar ciertos conocimientos. Estos conocimientos estarán no sólo relacionados con la tecnología, como veíamos en la sección de aspectos tecnológicos, sino también con el punto de vista de documentación, planificación, redacción y exposición.

Un proyecto queda bastante lejos de cualquier práctica que hayamos podido realizar durante la carrera, es algo a lo que te enfrentas desde cero, algo sobre lo que has de documentarte, leer cientos de fuentes hasta llegar a entender lo que tenemos entre manos. Será en ese momento cuando estemos preparados para desarrollar el proyecto.

Una vez la planificación e incluso el desarrollo han sido completados nos queda mostrar nuestro trabajo a los demás, no sólo el redactar esta memoria, sino hacerlo de una manera lo más amena posible para el lector y, sobre todo, hacerlo de manera clara.

### 1.3. Aplicaciones similares

Una vez explicada, en líneas generales, el funcionamiento de la aplicación que aquí se plantea vamos a comentar algunas de las aplicaciones que pueden guardar, por su estructura o funcionamiento, alguna similitud. Existen numerosas aplicaciones que permiten monitorizar actividades de running o similares, es decir, tiempo empleado, distancia recorrida y en algunos casos la ruta seguida.

De la misma manera podemos encontrar aplicaciones de brújula aumentada o incluso manuales de supervivencia en formato aplicación Android.

El problema surge cuando se intenta añadir realidad aumentada y funciones de seguridad, nos encontramos que no existen aplicaciones que aúnen en una aplicación todos estos elementos. Este es, como hemos visto, el objetivo que se plantea en esta aplicación SOFTSPM.

### 1.4. Experiencias previas

El objetivo de esta sección es poner de manifiesto algunas de las experiencias previas en relación con las tecnologías que son empleadas durante el proyecto.

En cuanto a la parte servidora, como veremos más adelante, está desarrollada sobre Django. Django es una tecnología con la que ya hemos trabajado en alguna asignatura durante la carrera. Podemos decir que tenemos una experiencia previa en este campo.

En cuanto a Android sucede algo similar. En MADSA (Mobile Application Student Association) desarrollamos conjuntamente algunas aplicaciones con este sistema. alguna de estas aplicaciones sirvió de ayuda para la realización del módulo de realidad aumentada que se incluirá en SOFTSPM.

### 1.5. Estimación de la planificación realizada

En esta sección pondremos sobre la mesa el desarrollo del proyecto, este desarrollo lo exponemos en base a una línea temporal.

Una vez se desarrolló la idea de realizar una aplicación como la que brevemente hemos descrito se empezó a pensar en los siguientes pasos. Estos pasos comenzaron en julio de 2012, durante el verano de 2012 se desarrolló lo correspondiente a la parte de la aplicación sobre Android. Posteriormente y al comienzo de las clases el proyecto sufrió un parón debido a cuestiones de trabajo y estudios.

Durante las Navidades de 2012 se comienza a desarrollar conjuntamente el servidor Web y la interfaz Web de SOFTSPM. Este trabajo continuo hasta mediados de marzo. Desde este momento la estructura y una versión preliminar de la aplicación ya están desarrolladas.

Una vez contamos con un versión preliminar se comenzó el proceso de reparar algunos errores y una vez hecho esto, desplegamos en la red nuestra aplicación, estamos hablando de finales de marzo.

Desde finales de marzo se empezó con los procesos de pruebas con usuarios externos a la aplicación. Se elaboró un "kit de pruebas". Este kit consta de un manual de usuario (disponible para el lector en el apéndice A), una encuesta (apéndice B) y el instalador de la aplicación. El kit fue entregado a los usuarios a mediados de mayo, los resultados empezaron a llegar una semana después.

En base a las experiencias de los usuarios se realizaron pequeños cambios en la aplicación para corregir errores en la experiencia de usuario.

La redacción de la memoria que el lector tiene entre las manos se comenzó a primeros del mes de mayo hasta mediados del mes de junio.

Es difícil establecer una franja de tiempo total empleado en la aplicación, el proceso ha sido largo. Desde la documentación hasta la fase de pruebas ha pasado prácticamente un año.

## 1.6. Estructura de este documento

Para facilitar la lectura y entendimiento de este documento es muy interesante a la vez que recomendable exponer algunos detalles sobre la estructura que se ha seguido. Vamos a recorrer ligeramente cada uno de los capítulos sobre los que se extiende dicha memoria.

- Capítulo 1: Este primer capítulo es una introducción sobre el proyecto, que incluye entre otros una motivación y objetivos del proyecto.
- Capítulo 2: Enumeramos y comentamos las tecnologías empleadas así como su estado de desarrollo actual.
- Capítulo 3: Diseño e implementación de cada una de las partes que contribuyen al proyecto.

- Capítulo 4: Pruebas, puesta en marcha y resultados.
- Capítulo 5: Conclusiones del proyecto.

De la misma manera se incluyen apéndices, en particular presupuesto del **Proyecto**, **Manual básico de usuario** y **Encuesta**. Al final de esta memoria de proyecto se adjuntan, por si el lector lo precisa, una bibliografía que permitirá al lector ampliar información o aclarar alguno de los temas que aquí se tratan.

En formato digital se incluirá en un CD/DVD los elementos que forman parte de la aplicación. Estos son los códigos adjuntos al proyecto:

- Código Android de la aplicación móvil
- Código del servidor en Django
- Código HTML y CSS de la Interfaz Web



## Estado del Arte

Todo tiene un principio y un contexto, durante este segundo capítulo vamos a introducir y comentar la situación de las tecnologías empleadas en la implementación de este proyecto. De la misma manera tomaremos una visión general sobre otros aspectos como las aplicaciones móviles, realidad aumentada o la geolocalización.

### 2.1. Arquitectura de la aplicación

La estructura que sigue la aplicación que tenemos entre manos es la conocida arquitectura Cliente - Servidor. Dicha arquitectura es un modelo de aplicación distribuida, caracterizada por una división clara de los elementos que componen el proyecto. En nuestro caso éstos quedan delimitados en torno a los servicios o recursos que ofrecen. Más en detalle quedaría de la siguiente forma:

- **Servidor:** Se trata de uno de los elementos principales en este tipo de arquitectura distribuida. El servidor se encarga de poner a disposición de los clientes una serie de servicios o recursos de la aplicación ante una demanda de los mismos.
- **Cliente:** Son los elementos que hacen uso de los recursos ofrecidos por el servidor.

Podemos resumirlo de forma sencilla de la siguiente manera, el servidor ofrece un servicio ante una petición del cliente.

¿Qué ventajas ofrecen este tipo de arquitecturas? Su principal ventaja es la división de la capacidad de cómputo entre las partes anteriormente comentadas, es decir, al ser dispositivos diferentes la funcionalidad, la organización y el servicio en sí están descentralizados.

Otra de sus ventajas viene en su diseño y posterior implementación. Esta división de tareas nos facilita y aclara los servicios al subdividir el sistema en servicios, es decir, presenta un muy buen comportamiento de escalabilidad.

El funcionamiento en conjunto de un servidor y sus clientes se realiza mediante la interacción de ambas partes. Tenemos una red en la que existe una comunicación directa entre las partes, los clientes se conectan a un servidor en el que son ofertados una serie de recursos o servicios. El servidor es el que soporta la gestión principal de la aplicación, y el que enlaza y tramita la información del sistema.

La elección de esta arquitectura para nuestra aplicación de seguimiento de actividades está básicamente propiciada por la posibilidad de delimitar las funciones de cada una de las partes y en consecuencia poder desarrollar cada parte por separado, es decir, Python para la parte servidora y Android para el cliente. Esta división en partes no permite centrarnos secuencialmente en cada una de las partes haciendo más sencillo su implementación.

## 2.2. Python

Hablar de Python es hablar de un lenguaje de programación bajo licencia de código abierto, software libre. Una de sus características es el tratarse de un lenguaje interpretado, cuya base intenta conseguir una sintaxis clara y limpia. Es un lenguaje multiparadigma con las siguientes particularidades:

- Orientado a objetos
- Imperativos
- Funcional
- Reflexivo

Otro de los factores que hacen especialmente interesante este lenguaje es que se trata de un lenguaje disponible en multitud de entornos. Este hecho nos facilitara en gran medida su adaptación a diferentes sistemas y dispositivos.

En cuanto a la historia y desarrollo del lenguaje nos remontamos a finales de la década de los 80 cuando el holandés Guido van Rossum creó y distribuyó este lenguaje. Actualmente la última versión presentada es la 3.3.1 Debido a la larga trayectoria de este lenguaje existe gran multitud de

material y APIs que facilitan enormemente trabajar con esta tecnología. Python cuenta con una página oficial [5] donde el lector podrá, si así lo desea ampliar la información concerniente a este lenguaje. Uno de los exponentes más relevantes para el desarrollo de la aplicación que nos atañe es el framework de desarrollo Web Django, el cual veremos en profundidad a continuación.

## 2.3. Django - Framework de desarrollo Web

Se trata de un framework de desarrollo Web, escrito en Python y distribuido bajo una licencia de código abierto. Django podría ser incluido en el paradigma de Modelo-Vista-Controlador.

Django tiene una filosofía básica, la reutilización, conectividad y extensibilidad de sus componentes, de esta forma se consigue un desarrollo relativamente rápido y sencillo de aplicaciones Web.

Las características que Django presenta con más énfasis son las siguientes:

- Mapa de objetos relacional
- Interfaz de administración automática
- Diseño optimizado y elegante de URLs
- Sistema de caché
- Internacionalización

En definitiva, Django pone a nuestra disposición todo lo necesario para desarrollar aplicaciones Web complejas de una manera modularizada y simple, esta incluye un servidor sobre el que lanzar la aplicación así como soporte SQL para bases de datos.

Django también incluye múltiples módulos en lo que respecta a la gestión de usuario, registro login y demás.

Es un sistema en constante desarrollo y como comentábamos íntimamente relacionado con Python. Actualmente la última versión es la 1.5.1. El lector puede ampliar información sobre Django y sus características en la página oficial del proyecto [3].

## 2.4. Android

Android, como plataforma de software es, hoy en día, uno de los sistemas que mayor crecimiento ha experimentado en los últimos años. En abril de 2013 se hizo público que Android representaba 9 de cada 10 ventas de nuevos smartphones. Esta difusión junto con el potencial de este sistema fue una de las razones principales por la que se eligió Android para la parte cliente (terminal móvil) de la aplicación SOFTSPM.

Vamos a enumerar algunas ideas generales referentes a esta popular plataforma, así como las principales ventajas e innovaciones que Android nos ofrece.

### 2.4.1. El sistema operativo Android

Android es una plataforma de desarrollo de software que incluye un sistema operativo, orientado principalmente a los teléfonos móviles inteligentes. Android cuenta con un Kernel derivado de Linux. Éste hecho y su distribución bajo una licencia libre o abierta hacen de Android uno de los mayores proyectos de software libre.

Google adquirió Android Inc. en el año 2005. Desde entonces el gigante tecnológico ha sido el mayor valedor de esta plataforma. Este apoyo no representa únicamente el importante aspecto económico, sino también el apoyo a su difusión.

Ahora veremos Android desde dos puntos de vista diferentes, y que en mi opinión son dos de los pilares del enorme auge de Android. Estos pilares son: sus comunidades de desarrolladores y las comunidades de usuarios.

El desarrollo de aplicaciones para Android cuenta con una serie de elementos que facilitan y fomentan el desarrollo. Entre estos elementos podemos destacar el SDK (Software Development Kit), el cual cuenta entre otras muchas funcionalidades con un emulador en el que testear nuestras aplicaciones. Android permite en su desarrollo el que su código fuera implementado en Java, esto facilitó mucho la adaptación y reconversión entre programadores de Java y Android.

En lo respectivo al punto de vista de usuario, Android ha ido ganando terreno a sus competidores. Esto ha sido debido a la gran oferta de aplicaciones móviles disponibles para Android y sobre todo al aumento que los SmartPhones con sistema operativo Android representa en la tasa de mercado, debido en gran parte a la gran variedad de terminales que se ofrecen.

Estos dos puntos han hecho que Android cuente con una gran perspectiva de futuro. Android se ha asentado como una opción de peso en el campo del software libre, comunidades de usuario

y desarrolladores. Cada vez más Android está a la cabeza en software relacionado con la telefonía móvil.

### 2.4.2. Desarrollo en Android

Android permite un desarrollo sencillo y potente, Google pone a disposición de los desarrolladores paquetes de desarrollos SDK con todas las herramientas necesarias. A su vez existen numerosas APIs que simplifican o añaden funcionalidades o conectividad entre otros sistemas o interfaces.

En mi opinión, Android simplifica el control de numerosos elementos que mediante otras plataformas puede llegar a ser tedioso. Un buen ejemplo de esta situación es el desarrollo de interfaces de usuario, algo muy cuidado con el SDK de Android. Pero sin duda es el control de todo el hardware del teléfono lo que mayor ventaja representa. Con muy pocas líneas de código somos capaces de realizar controles básicos de GPS, conexiones y sensores de movimiento entre muchas otras funcionalidades interesantes.

### 2.4.3. Competidores y comunidades

En este momento sería interesante nombrar algunos de los competidores principales que tiene la plataforma Android en el campo de la telefonía móvil. El objetivo es simplemente hacernos una idea de la situación en la que se encuentra el mercado de la telefonía en estos momentos.

Los principales competidores son:

- iOS Movil
- BlackBerry OS
- Windows Phone

Sin duda, si tuviéramos que hablar de un competidor principal en el campo de los smartphone sería iOS Movil de Apple, no obstante la comunidad de desarrolladores que sustentan Android es mayor y más activa. En la siguiente imagen podemos ver una comparativa sobre las ventas de terminales, ordenadas por sistema operativo en el tercer trimestre de 2012.

**Top Six Smartphone Mobile Operating Systems, Shipments, and Market Share, Q3 2012 (Preliminary)**  
(Units in Millions)

Operating System	3Q12 Shipment Volumes	3Q12 Market Share	3Q11 Shipment Volumes	3Q11 Market Share	Year-Over- Year Change
Android	136.0	75.0%	71.0	57.5%	91.5%
iOS	26.9	14.9%	17.1	13.8%	57.3%
BlackBerry	7.7	4.3%	11.8	9.5%	-34.7%
Symbian	4.1	2.3%	18.1	14.6%	-77.3%
Windows Phone 7/ Windows Mobile	3.6	2.0%	1.5	1.2%	140.0%
Linux	2.8	1.5%	4.1	3.3%	-31.7%
Others	0.0	0.0%	0.1	0.1%	-100.0%
Totals	181.1	100.0%	123.7	100.0%	46.4%

Figura 2.1: *Cuota de mercado móvil 2012*

Existen desarrolladores y comunidades tanto de profesionales como no profesionales que diariamente aumentan el número de aplicaciones para la plataforma Android, sin olvidar diferentes versiones y adaptaciones de Android para otros dispositivos.

### Más información

Android es sin duda una gran apuesta en el futuro del software libre para dispositivos móviles, existen infinidad de fuentes de información en las que el lector, sí así lo desea, puede seguir investigando sobre la materia. Un buen punto de inicio es la página oficial de Android [1].

## 2.5. HTML Y CSS

SOFTSPM cuenta con una interfaz Web, que aunque servida por un servidor implementado sobre Django, es para el público usuario mostrado por un navegador. Este hecho hace evidente la importancia que el código HTML tiene para la interfaz Web.

La arquitectura Cliente-Servidor que explicábamos en el primer punto de este capítulo ponía de manifiesto la importancia del servidor. En nuestro caso el servidor, aparte de otras muchas funcionalidades, debe ser capaz de servir HTML con el que el usuario final interactuará a través de su interfaz Web. Es por esto que todo cambio en nuestro servidor tendrá un efecto en el usuario final, no sólo en el funcionamiento de la aplicación sino en el código HTML servidor al navegador.

Los navegadores actuales no solo son sensibles a código HTML, sino que entorno a esta tecnología

se han ido desarrollando otras tecnologías que mejoran o complementan a HTML. Desde el auge de las Webs 2.0, en el que ya no sólo se ofrece un contenido estático sino que el usuario es el protagonista al interactuar con el site, se han desarrollado una importante generación de tecnologías entre las que podemos destacar AJAX.

Otra de las tecnologías que merece especial mención es sin duda las hojas de estilo en cascada, CSS, las cuales permiten modificar las propiedades de estilo de documentos HTML. Lo veremos más en profundidad en la siguiente sección.

No solo las tecnologías en torno a HTML han ido desarrollándose. El propio HTML, el cual, desde un primer HTML por el año 1991 hasta el actual y muy interesante HTML5. Estos avances representan en sí mismos esta evolución que en gran parte atiende a unos mayores requerimientos de las aplicaciones pero también de los usuarios.

### 2.5.1. Hojas de estilo CSS

Las hojas de estilo en cascada, CSS por sus siglas en inglés, hacen referencia a un lenguaje que nos permite describir ciertas opciones de aspecto y formato de nuestros documentos HTML. Su principal aplicación es la de agrupar y dar propiedades estéticas a nuestros documentos HTML, aunque también acepta otro tipo de lenguajes.

Normalmente las hojas de estilo suelen añadirse mediante la importación de un archivo externo, aunque puede ser incorporado directamente en nuestro documento HTML.

Las ventajas principales que nos permite CSS es sin duda la de modularidad y reutilización de código. De esta forma obtenemos una división entre la estética o forma de nuestro documento y la funcionalidad de éste. Con esto conseguimos una mayor claridad en la estructura de nuestros servicios y la reutilización de CSS entre documentos con una estructura estética y de apariencia idéntica o similar.

En cuanto al desarrollo y estado actual de la tecnología CSS, podemos simplemente indicar que la primera versión fue publicada en 1996, actualmente nos encontramos con CSS3, lanzada oficialmente en 2011, y que incluyen entre sus novedades nuevas funcionalidades.

### 2.5.2. World Wide Web Consortium W3C

El World Wide Web Consortium W3C [7] se trata de un consorcio internacional que en la práctica funciona redactando recomendaciones entorno a la World Wide Web. Si estamos interesados en

ampliar información sobre HTML o CSS en sus últimas versiones, este consorcio es un buen punto de partida en lo relativo a estas y otras tecnologías de la red.

## 2.6. sqlite

sqlite es un sistema de gestión de bases de datos relacional, se trata de un lenguaje estructurado de consulta para acceso a bases de datos relacionales. El gran potencial que este tipo de bases de datos ofrece es el manejo del álgebra y el cálculo relacional que realizan consultas para recuperar la información de utilidad para nuestras aplicaciones.

Entrando particularmente en sqlite3 podemos entender sqlite como un proyecto de dominio público, sus primeras versiones datan de agosto del año 2000. Su última versión estable según la página oficial del proyecto es la 3.7.17

Algunas de sus características que hacen de sqlite una buena opción para la gestión de bases de datos son:

- Características ACID (Atomic, Consistent, Isolated and Durable).
- Derivada de SQL92, se conserva su semántica y estructura.
- Baja necesidad de espacio, muy ligera.
- No necesita dependencias externas.
- Todo el código y documentación es de dominio público.
- Multiplataforma.

Si el lector desea ampliar información puede hacerlo a través de la página oficial de sqlite [6].

## 2.7. Realidad Aumentada

La realidad aumentada puede ser entendida como un nuevo paradigma, un paradigma en el que la realidad es observada desde la pantalla de un dispositivo. Sobre esta pantalla se superponen con distintos fines una serie de elementos virtuales con el fin de mejorar o aumentar nuestra percepción de la realidad.

La principal diferencia con la realidad virtual es que la realidad no se modifica sino que simplemente se proyectan sobre la realidad física una serie de diversos elementos virtuales.

Con el avance de la tecnología en los últimos años la realidad aumentada ha sufrido un enorme auge, desde que una gran mayoría de nosotros contamos con smartphones tenemos a nuestro alcance elementos que son capaces de obtener la realidad y mostrárnosla mediante una pantalla, sensores y otros actuadores.

En mi opinión es un campo con una gran proyección de futuro, en estos momentos se continua estudiando el uso de dispositivos especializados. Algunos de estos son las gafas de visión artificial propios muchas veces más de ciencia ficción que de una realidad presente, pero que gracias a la realidad aumentada se han convertido en objetivos perfectamente alcanzables.

### 2.7.1. Tecnología para Realidad Aumentada

La realidad aumentada puede ser basada sobre dos pilares clave, por un lado tenemos la obtención de información física, información real de lo que estamos observando. Por otro lado tenemos que ser capaces de mostrar toda esta información al usuario de una manera eficiente, es por esto que se hace muy recomendable el uso de displays para proporcionar esa información.

Existen diferentes modos de display, pero por simplicidad vamos a comentar de forma rápida el que se emplea en la aplicación SOFTSPM. En esta aplicación contamos con un display, la pantalla de un smartphone en el que se combinan diferentes elementos:

- Imagen real: La cámara de un smartphone es capaz de recoger la imagen proyectada por la cámara y mostrarla sobre la pantalla del terminal.
- Sensores: El terminal recoge información sobre la posición, orientación, aceleración, etc.

Los elementos anteriores son mezclados sobre el display final, de forma que no solo representamos a la realidad mediante la imagen captada, sino que sobre ella y de manera conjunta somos capaces de ampliar dicha información con elementos virtuales.

### 2.7.2. Aplicaciones de la Realidad Aumentada

Las aplicaciones y posibilidades en realidad aumentada son enormes, ya que es capaz de representar un punto de vista desconocido o sin explotar hasta este momento. Vamos a poner una

serie de campos en los que por su utilidad o avances está teniendo un especial recorrido en estos momentos.

- Medicina: Es uno de los campos donde este tipo de tecnología puede marcar un gran cambio, desde simplemente un punto de vista educativo hasta servir como medio para realizar intervenciones quirúrgicas en seres humanos.
- Turismo: La realidad aumentada puede servir como un guía turístico personal, que se adapte en cada momento a nuestros deseos y recorrido, en el que se ofrecerá un información personalizada del tema a tratar bajo unas determinadas circunstancias.
- Arquitectura y urbanismo: En este campo destacan sobre otros la posibilidad que la realidad aumentada ofrece de ver objetos sobre una determinada posición. Esto por ejemplo nos permitiría observar un edificio sobre su ubicación final incluso antes de su construcción, es decir, aún sobre plano.

### 2.7.3. Próximos avances en Realidad Aumentada

La Realidad Aumentada está limitada por el hardware de que disponemos. Sería interesante, como algunos proyectos pretenden, cambiar la simple pantalla de un smartphone por un proyector holográfico que no sólo aumente las capacidades de determinada aplicación, sino que, nos permita entender esa realidad de una manera más cómoda o natural.

Cada vez más estos dispositivos van a ir sustituyen a objetos convencionales, en campos tan diversos como la automoción, planificación de rutas de vuelos, etc.

### 2.7.4. Framework de Realidad Aumentada

Actualmente existen algunos frameworks de desarrollo cuya funcionalidad principal es facilitar la creación de módulos o aplicaciones de realidad aumentada. Estos frameworks usualmente trabajan con un lenguaje de programación determinado, en este caso dos de los framework sobre Android que más interés han despertado en la comunidad son los siguientes:

- ARToolKit [2]
- DroidAR [4]

## 2.8. Geolocalización y sensores de movimiento

Hoy en día, las tecnologías de localización y captación del movimiento están cambiando en gran medida la forma en las que utilizamos las aplicaciones móviles, aplicaciones deportivas, rutas y en definitiva cualquier aplicación que nos muestre información personalizada basada en nuestra posición o en nuestros movimientos.

Estas tecnologías, junto al avance de las tecnologías incluidas en los smartphones, y el avance en radiocomunicaciones de banda ancha, han hecho posible que cualquier persona tenga la capacidad de detectar sus movimientos y su posición gracias a un terminal móvil en nuestro bolsillo.

De esta forma SOFTSPM nos representa estos dos aspectos, por un lado sigue nuestra ruta gracias a la geolocalización y por el otro tenemos los sensores de caída que son simplemente una detección de un movimiento más o menos brusco.

## 2.9. Difusión y posicionamiento en buscadores

Uno de los puntos donde la mayoría de aplicaciones online fracasan es en su visibilidad. Una aplicación que no podemos encontrar en buscadores de forma sencilla e intuitiva prácticamente no existe y está abocada a la desaparición.

Esta visibilidad se consigue con SEO (Search Engine Optimization). Se trata de una disciplina del marketing online que permite dar difusión en buscadores (principalmente Google) a nuestros productos frente a los demás en un campo determinado.

El SEO es un campo en constante cambio y con un auge enorme en los últimos años. Por esta razón que en este proyecto se incluirán una serie de pinceladas para mejorar la disposición de los buscadores a dar difusión a nuestra aplicación.

Existen dos vertientes en SEO que afectan de forma importante al posicionamiento de nuestra aplicación:

- Factores OnPage: Se tratan de factores aplicables técnicamente, por ejemplo un código HTML adaptado a una serie de patrones de búsqueda.
- Factores OffPage: Factores que no dependen directamente de nosotros, como por ejemplo los enlaces entrantes a nuestra página o la difusión de nuestro contenido.

En este proyecto nos centraremos principalmente en el SEO OnPage, se trata del que depende

directamente de nosotros e implica un conocimiento algo más técnico.

Estas técnicas o modificaciones se basan principalmente en aquello que examinan los buscadores, es decir, el código HTML servido por nuestro servidor. Existen una serie de recomendaciones o buenas prácticas para optimizar nuestro contenido hacía una serie de palabras clave sobre las que queremos difundir nuestra aplicación online.

En este proyecto simplemente vamos a describir e implementar algunas de estas recomendaciones, a un nivel muy básico ya que no es el objetivo principal de este proyecto.

# Capítulo 3

## Diseño de la aplicación

En este capítulo tercero vamos a presentar y explicar en profundidad las características y elementos más importantes en lo relativo al diseño y la implementación de la aplicación realizada, SOFTSPM.

Se desarrollará cada parte pormenorizadamente con el fin de dar al lector la posibilidad de entender las razones y el desarrollo elegido para cada uno de los elementos que cobran relevancia para el funcionamiento final de esta aplicación.

Empezaremos este capítulo con una introducción a cada uno de los elementos que componen la estructura de esta aplicación, que recordemos se basa en el modelo Cliente-Servidor. A continuación iremos más en profundidad sobre cada uno de los elementos que considero relevantes para entender la estructura de esta aplicación.

A lo largo de este capítulo se acompañarán a los distintos apartados con esquemas y capturas de pantalla que ayuden a entender e ilustrar cada una de las funcionalidades y elementos de esta aplicación.

Como comentábamos anteriormente se enunciarán algunas de las técnicas empleadas para mejorar la difusión en buscadores de nuestra aplicación web, con el fin de aumentar el alcance a los usuarios potenciales de SOFTSPM.

### 3.1. Introducción

La aplicación sobre la que se basa esta memoria, SOFTSPM, cuenta con una serie de elementos esenciales para el correcto funcionamiento de la misma. Podemos ilustrar un sistema equivalente a

nuestra aplicación de la forma que sigue, y que representa idealmente a cada uno de los elementos que componen nuestro sistema Cliente-Servidor.

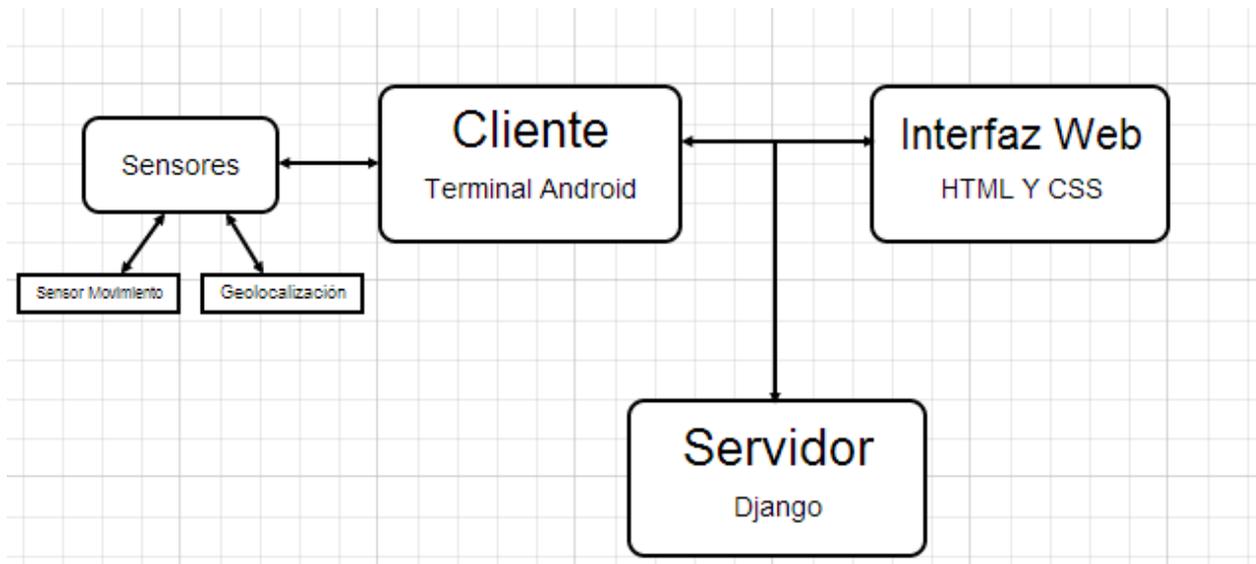


Figura 3.1: *Esquema Básico de la aplicación*

A continuación veremos de una manera aproximada cada una de las partes en la que se estructura la aplicación respecto al diagrama anterior.

- **Servidor:** Representa la carga lógica y de proceso principal de la aplicación. Por una parte intercambia información con el terminal Android y mantiene los servicios al usuario a través de la interfaz Web. Está desarrollado sobre el framework Django.
- **Interfaz Web:** Nuestro servidor pondrá a disposición del usuario final una interfaz desarrollada en HTML y CSS que permitirá al usuario una interacción para realizar cualquiera de las operaciones previstas en la aplicación.
- **Cliente-Terminal Móvil:** Esta aplicación estará en contacto con el Servidor siempre que la conexión móvil/WiFi lo permita, el terminal cuenta con una serie de servicios sensores que obtendrán información del exterior, en particular sensores de movimiento, magnéticos y geolocalización.

Como vemos existen tres componentes o elementos en los que dividiremos la arquitectura de la aplicación, por un lado la interfaz Web y el servidor, ambos estrechamente relacionados y por otra

parte nuestra aplicación para dispositivos Android.

## **3.2. Arquitectura REST en la aplicación**

Como veíamos en la introducción de este capítulo tercero tenemos una serie de componentes interrelacionados, los cuales se relacionan y organizan en base a una arquitectura REST. Veamos un poco más en profundidad esta interesante tecnología.

Este tipo de arquitectura está pensada para sistemas distribuidos, nos ofrece una serie de pautas a seguir para el desarrollo de aplicaciones Web cumpliendo con los requisitos y estándares de la World Wide Web.

La arquitectura REST presenta una serie de elementos arquitectónicos entre los que destacamos cliente y servidor. El servidor ofrece una lista de recursos que son accedidos mediante peticiones desde el cliente. Una de las particularidades que ofrecen estos recursos es el concepto stateless, es decir, los recursos ofrecen la respuesta simplemente basándose en la petición, no en situaciones pasadas.

Otro de los detalles que caracterizan este tipo de arquitectura es la conexión entre recursos, unos recursos deben tener enlaces (hipervínculos) a otros recursos. De esta forma conseguimos una interdependencia entre los recursos disponibles. Profundizando en los recursos ofrecidos por el servidor podemos darnos cuenta de que todo lo que está disponible hacia el cliente es un recurso, es decir, documentos, servicios, imágenes, etc.

Estos recursos se identifican inequívocamente bajo un identificador único, en nuestro caso una URL específica a cada recurso. Los recursos no representan detalles acerca de su implementación.

Es también interesante comentar algunas de las características de este tipo de arquitectura relacional, sobre todo las ventajas por las que se eligió para SOFTSPM. Estas características son:

- Sencillez
- Escalabilidad
- Rendimiento
- Visibilidad
- Generalidad

En el caso que nos atañe, SOFTSPM, nuestra parte servidora ofrece una serie de recursos, los cuales podemos ver enumerados a continuación junto con la URL de acceso a cada uno de ellos.

```
# Uncomment the next line to enable the admin:
# url(r'^admin/', include(admin.site.urls)),
(r'^$', 'say_main',),
(r'^loginMobile', 'loginMobile'),
(r'^login', 'loginUser'),
(r'^create_User/(.*)/(.*)', 'create_User'),
(r'^mostrarAlertas', 'mostrarAlertas'),
(r'^mostrarMensajesUsuario/(.*)', 'mostrarMensajesUsuario'),
(r'^mostrarMensajes', 'mostrarMensajes'),
(r'^newUser', 'newUser'),
(r'^logout', 'log_out'),
(r'^component/(?P<path>.*)$', 'django.static.serve()'),
{'document_root': 'component'}),
(r'^actualizarCoordenadas/(.*)/(.*)', 'actualizarCoordenadas',),
(r'^alerta/(.*)/(.*)', 'crearAlerta',),
(r'^consultarCoordenadas/(.*)', 'consultarCoordenadas',),
(r'^crearmsgmovil', 'crear_msg_movil',),
(r'^crear_msg', 'crear_msg',),
(r'^fichaAmigo/(.*)', 'fichaAmigo',),
(r'^fichaUsuario', 'fichaUsuario',),
(r'^css/(?P<path>.*)$', 'servirCss',),
(r'^agregarAmigo', 'agregarAmigo'),
(r'^amigos', 'verAmigos'),
(r'^agregarRuta/(.*)/(.*)/(.*)/(.*)', 'agregarRuta',),
(r'^verRutas/', 'verRutas',),
(r'^(.*)/(.*)$', 'mostrarContenido'),
```

Figura 3.2: Recursos disponibles en la aplicación

### 3.2.1. Métodos HTTP y REST

Existen una serie de métodos HTTP con la que se representa una semántica en las peticiones Cliente-Servidor. Estos métodos son la base de la comunicación entre partes, son cuatro los tipos de peticiones que existen, veámoslas.

- GET: Petición de un recurso especificado. Puede ser parametrizado para delimitar la representación de la solicitud.
- POST: Este método crea nuevos recursos y añade datos a un recurso ya existente.
- PUT: Creamos un nuevo recurso o actualizamos uno ya existente.

- DELETE: Eliminamos la vinculación en un recurso especificado.

En el desarrollo de la aplicación hemos empleado principalmente, debido a nuestras necesidades, los métodos GET y POST.

### 3.3. Aplicación Android

Hasta este momento nos hemos limitado a comentar los elementos principales de la estructura de nuestro sistema, en esta sección no detallaremos demasiado en profundidad la parte cliente, la cual, como vimos está desarrollada en Android.

#### 3.3.1. Introducción

Como veíamos en el esquema estructural de la aplicación nuestro cliente (Android) es uno de los elementos fundamentales del sistema. Por eso en los puntos que siguen presentaremos las funcionalidades, actividades y esquemas internos de la aplicación.

A su vez repasaremos cómo se realiza el intercambio de información entre esta aplicación y el servidor. Pasamos pues a ver tanto la implementación como el diseño empleado en la aplicación Android.

#### 3.3.2. Diseño y funcionalidades de la aplicación

Durante la planificación inicial del proyecto se enumeraron una serie de características de obligado cumplimiento, en base a estas premisas necesitamos un diseño que admita todas estas funcionalidades y requisitos. Una vez reconocidas estas funciones, las cuales podemos llamar funciones principales, debemos implementarlas consiguiendo la mayor eficiencia de los recursos de nuestro dispositivo Android.

Durante la fase de diseño tampoco podemos olvidar a quien va dirigida la aplicación, es decir, al usuario final. Teniendo esta condición en mente debemos adaptar la aplicación, su funcionamiento y diseño a conseguir una experiencia de usuario lo más simple y eficiente posible. Otra de las premisas que debemos observar al diseñar y elegir cada uno de los elementos es establecer unos requisitos técnicos. Estos requerimientos son un mínimo de hardware (sensores, conectividad, etc.) y software (versión Android) sobre el que la aplicación funciona de manera óptima. Por ello debido

a las características de un terminal Android de gama media se establece diseñar la aplicación sobre la versión Android 2.1

Ahora enumeramos las principales funciones que la aplicación cliente ha de proporcionar, tanto al usuario final como al servidor.

- Seguimiento: Actividad que monitoriza todos los factores esperados de la aplicación. En esta se incluye tanto monitorización deportiva como de seguridad de la misma.
- Mapa Zonal: Mapa de nuestra ubicación actual. En este mapa no sólo veremos la geografía de la zona sino que también mostraremos una serie de puntos de interés, los cuales, podrán ser de utilidad para el usuario durante la actividad al aire libre.
- Brújula Aumentada: El usuario tendrá un brújula con realidad aumentada en la que se establece la dirección que el usuario ha de seguir para alcanzar el punto de interés establecido más cercano. Opcionalmente también puede ser configurada para funcionar como una brújula estándar.
- Opciones: En esta actividad se dará la opción al usuario de configurar una serie de parámetros para adaptar la aplicación SOFTSPM a los gustos del propio usuario. Algunos de estos parámetros configurables son las condiciones del GPS para el seguimiento de la actividad o el usuario por defecto para realizar la autenticación de usuario con el servidor de la aplicación.

Como podemos intuir la actividad o servicio central de la aplicación que radica en el seguimiento será la que se desarrolle más en profundidad para explicar detalladamente su funcionamiento. En el siguiente apartado comentaremos como se ha realizado la implementación de la aplicación.

### 3.3.3. Implementación

Para entender la implementación que se ha realizado en esta aplicación vamos a ver las clases, interfaces y elementos que componen la misma. Comencemos comentando el siguiente árbol estructural.

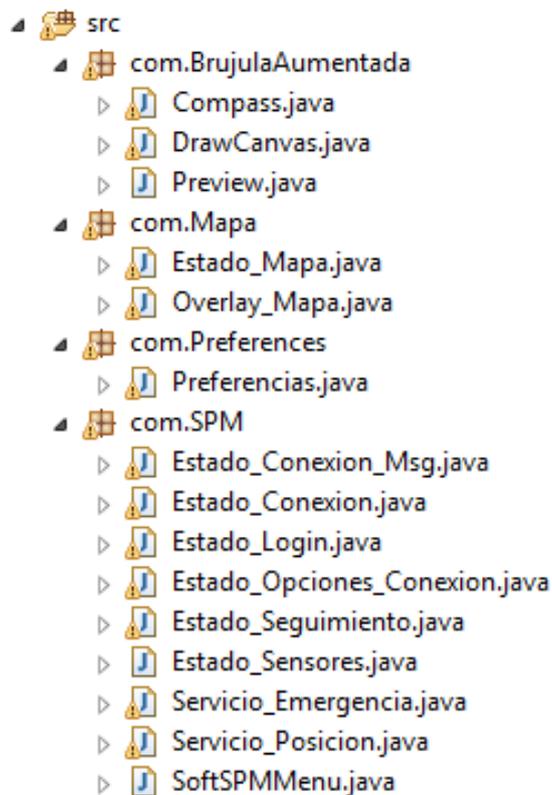


Figura 3.3: Estructura de clases por paquetes, Android

Las clases anteriores están agrupadas en paquetes con el fin de ordenar el código fuente de la aplicación y ser más claro su desarrollo, lectura y entendimiento por parte del lector o cualquier persona interesada en este proyecto.

- `com.BrujulaAumentada`: Incluye todas las clases que se encargan de trabajar con la brújula aumentada. Estas clases las podemos resumir en tres: una primera encargada de capturar la cámara de fotos del terminal y mostrarla por el display. Una segunda clase encargada de administrar y obtener información relativa a los sensores de movimiento del terminal y por última, la aplicación encargada de representar toda esta información sobre nuestro display.
- `com.Mapa`: Este paquete es el encargado de mostrar el mapa y sobreponer cada uno de los puntos de interés que se encuentren en la cobertura geográfica del mapa.
- `com.Preferences`: En este paquete planteamos al usuario la posibilidad de configurar varios parámetros relativos al uso de la aplicación, sobre todo de la parte de seguimiento GPS y autenticación.

- `com.SPM`: Sobre este paquete recae el peso de la aplicación, aquí encontramos el menú principal de la aplicación Android. De la misma manera incluimos la actividad de seguimiento y seguridad de la aplicación. Todas las comunicaciones entre la aplicación Android y el servidor así como los protocolos de seguridad están contenidos en este paquete.

Como podemos ver dentro del paquete `com.SPM` existen dos tipos de entidades. Nos referimos a las actividades y los servicios. Veamos cuales son las características de cada uno de estos con el fin de entender un poco mejor la estructura de este paquete.

- **Actividades:** Las actividades son clases, las cuales, componen la aplicación y trabajan en primer plano. Crean el núcleo central de funcionamiento.
- **Servicios:** Los servicios son componentes que trabajan en segundo plano, sin interacción directa con el usuario de la aplicación. Pueden ser lanzadas desde actividades. Se recomienda el uso de estos para derivar funciones de la actividad principal. Son muy usados debido a su eficiencia y gestión de recursos.

Analizando ambas figuras separadamente podemos concluir que la principal diferencia es la forma de ejecución. En nuestro caso los servicios son útiles por dos razones fundamentales: la primera es la capacidad de realizar acciones en segundo plano mientras tenemos nuestra actividad y la segunda es la facilidad para lanzar varias acciones al mismo tiempo, de esta forma, diversificamos la estructura y funcionamiento de nuestra aplicación.

Otra de las particularidades que presenta el desarrollo de Android, la cual se ha comentado anteriormente como una de sus principales ventajas, es el manejo de interfaces gráficas de usuario. Éstas son las que hemos utilizado durante la implementación de esta aplicación. Se ha empleado una terminología similar a las clases que representan para facilitar su interpretación al lector.

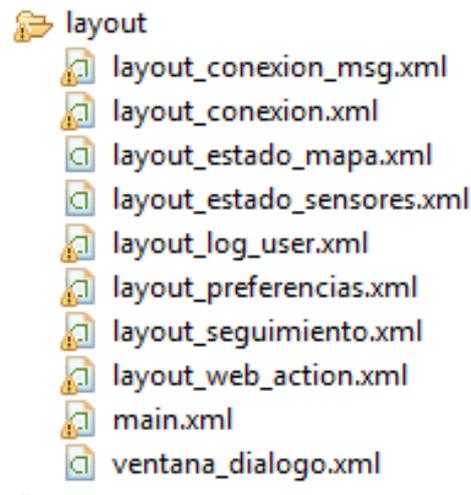


Figura 3.4: *Layout de la aplicación*

Cada uno de los layouts anteriores representa la interfaz que el usuario verá durante su experiencia de la aplicación. Estos layouts van relacionados con las actividades que veíamos anteriormente, de esta forma entre un layout y una actividad obtenemos el funcionamiento y capacidad de la aplicación.

En los puntos siguientes analizaremos las clases y servicios de los que forman la aplicación, su funcionamiento, su apariencia y particularidades sobre su programación que en mi opinión son interesantes. En general no se incluirá el código de las diferentes partes, el cual está a disposición del lector por si desea consultarlo.

### Menú de la aplicación

Esta actividad es la primera que se ejecuta al iniciar la aplicación en nuestro terminal. Se trata de un menú que ofrece una serie de opciones para iniciar cada uno de los servicios que veremos en los siguientes puntos. A continuación podemos ver el layout correspondiente a este menú de la aplicación.

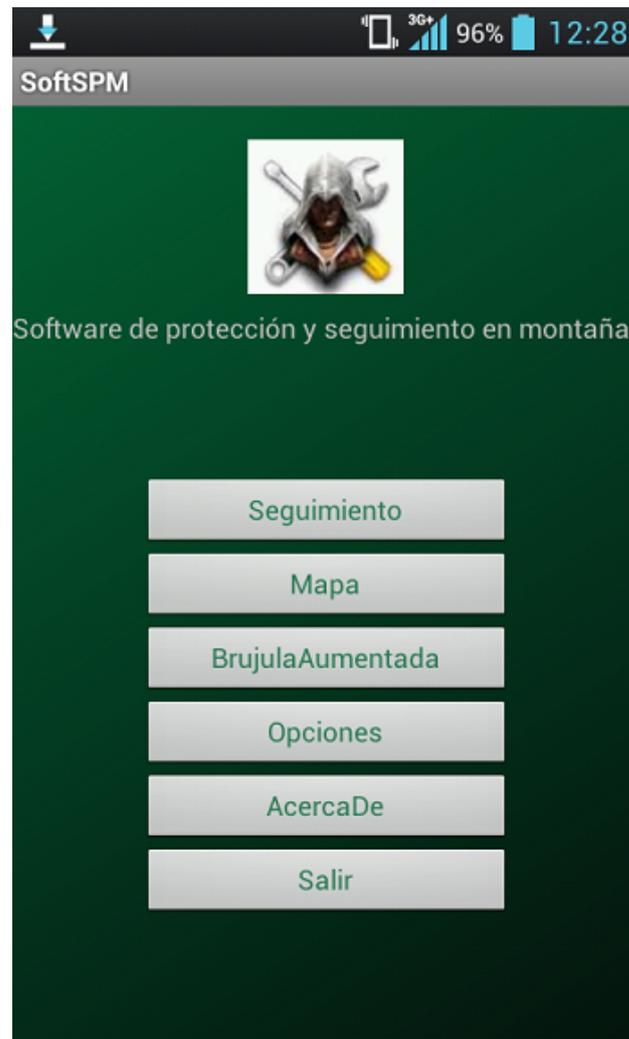


Figura 3.5: *Layout del menú de la aplicación*

### Mapas y sus actividades

En esta sección analizaremos el funcionamiento y la estructura del paquete com.Mapas, en el que, como veíamos anteriormente contamos con el análisis y posicionamiento en un mapa con unos determinados puntos de interés.

Este paquete lo componen dos actividades, EstadoMapa y EstadoOverlay. EstadoMapa es la actividad principal, mediante el GPS y geolocalización nos proporciona un punto de coordenadas, sobre este punto centraremos el mapa del área. La representación del mapa se realiza mediante una clase adecuada en Android, exactamente MapsActivity. Esta clase tiene una apariencia muy similar a Google Maps, tanto la interfaz como el funcionamiento han sido incorporadas al terminal

Android. Por otro lado podemos encontrar la clase EstadoOverlay. El objetivo de esta clase es representar sobre el mapa anterior una serie de información adicional. En el caso de SOFTSPM esta información viene dada por los puntos de interés, por ejemplo, los refugios o centros de primeros auxilios de la Cruz Roja que serán de gran utilidad en la realización de actividades al aire libre.

De la igual forma podemos realizar listas de puntos de interés (PDI) que se adapte a nuestros gustos o necesidades, para poner un ejemplo del gran campo de elección que tenemos con estos PDI elegimos los sistemas de GPS para coches, donde como PDI podemos señalar gasolineras, talleres o radares de tráfico.

En la siguiente imagen podemos ver el funcionamiento de la aplicación en cuanto a mapas se refiere, vemos claramente la indicación de dos de los PDI existentes.



Figura 3.6: *Layout Estado Mapa con PDI*

### Brújula Aumentada y sus actividades

Posiblemente la implementación de la brújula aumentada junto con la actividad de seguimiento es una de las que más complejidad, conlleva en cuanto al control de los sensores del terminal. La brújula aumentada emplea dos tipos de sensores básicos para determinar la orientación. El primero es el sensor de movimiento (acelerómetro), detectará los movimientos en el terminal, el segundo sensor será electromagnético, el cual nos indicara la posición del norte magnético.

El funcionamiento combina los dos sensores anteriores para, mediante realidad aumentada, orientarnos hacia un lugar específico. Tenemos dos opciones, por defecto la brújula nos indicara la dirección hacia el norte magnético, pero opcionalmente también puede ser configurado para indicar la dirección hacia uno de los PDIs introducidos en la aplicación.

Vamos a ver de forma rápida el funcionamiento de estos dos sensores, magnético y de movimiento.

- Sensor de movimiento o acelerómetro: Detecta las rotaciones en los ejes X, Y y Z ofreciéndonos valores con los que calcular la orientación de nuestro terminal en cada momento. Estas rotaciones de movimiento se consiguen detectando las aceleraciones de la gravedad en dichos movimientos.

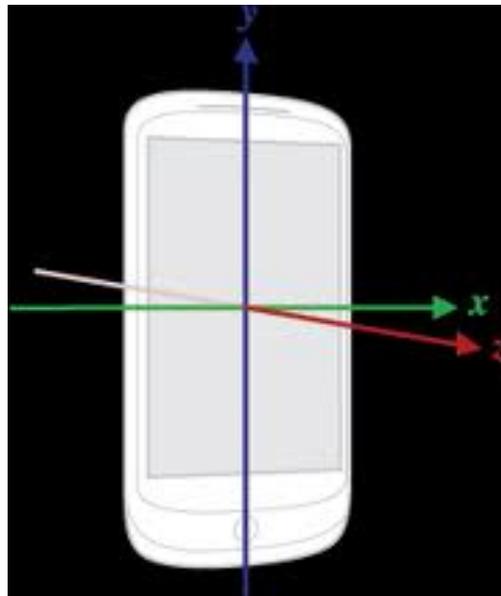


Figura 3.7: Esquema de los ejes de aceleración

- Sensor magnético: Consta de un detector de campos magnéticos, de esta forma identificamos

el norte magnético.

Una vez vistos algunos puntos interesantes sobre su funcionamiento e implementación interna vamos a concentrarnos en el usuario final, cómo utilizará este módulo de realidad aumentada y sus distintas opciones.

La brújula aumentada, como hemos comentado, nos indicará la dirección a seguir hasta la posición objetivo. Esto lo conseguimos calculando la posición actual, con la cual la aplicación obtiene el rumbo hacía la posición objetivo.

Para ilustrar estas explicaciones vamos a ver una de las posibles situaciones que puede encontrar el usuario mientras hace uso de esta función.

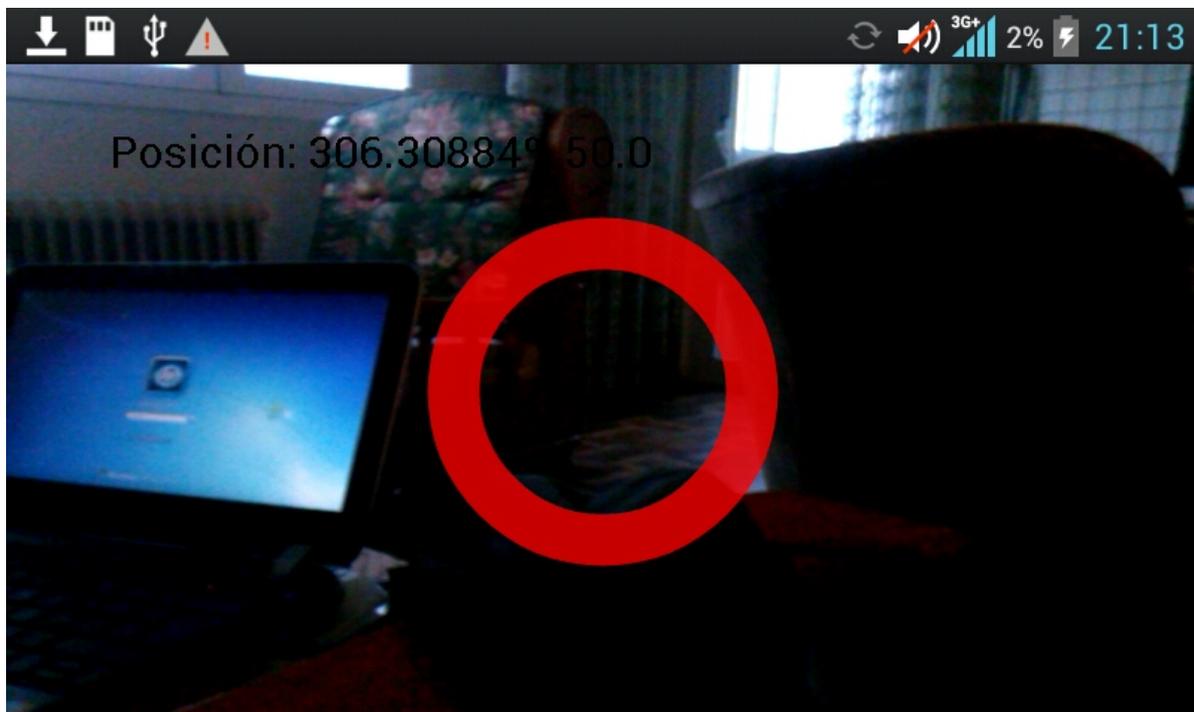


Figura 3.8: *Indicador de rumbo erroneo*

Como vemos obtenemos superpuesto a lo mostrado por la cámara una serie de elementos que nos proporcionan una información extra sobre nuestro rumbo. Por un lado tenemos impreso dos elementos, el rumbo actual y el rumbo objetivo, de esta forma el usuario podrá tomar referencias sencillas y visuales de la dirección que debe tomar para llegar a su objetivo.

Para ayudar al usuario a cuadrar la posición correcta sobre la pantalla se imprime una circunferencia que cambia de color en función de si el rumbo es correcto (color verde) o es incorrecto (color

rojo). Por sencillez la dirección correcta se considera con unos pequeños márgenes relativos, es decir, si el ángulo objetivo fuera de  $50^{\circ}$  consideraremos como ángulo correcto al abanico  $[45^{\circ}, 55^{\circ}]$ .

Veamos cómo quedaría una situación en la que el usuario está siguiendo el rumbo correcto.

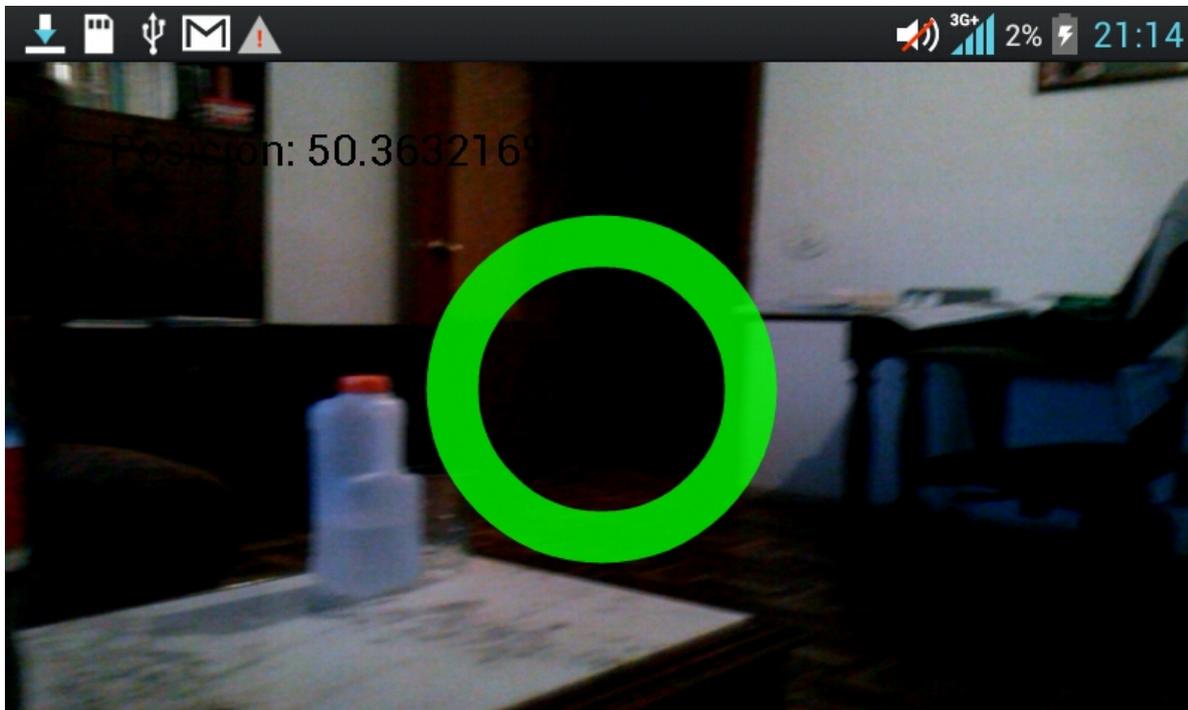


Figura 3.9: *Indicador de rumbo correcto*

### Preferencias de la aplicación

Una de las opciones que los usuarios más agradecen de una aplicación es la capacidad de personalizarla, en SOFTSPM se da la opción de configurar una serie de propiedades de la aplicación. Algunas de las funcionalidades que ofrece son la configuración de un usuario por defecto y los intervalos de actualización de los sensores de posicionamiento. No obstante en un futuro es posible incorporar otras opciones de configuración.

Analicemos cada una de estas tres opciones:

- Usuario por defecto: Esta opción permite que en la fase de autenticación con el servidor se emplee por defecto el nombre de usuario especificado.
- Intervalos de actualización de la geolocalización: El sensor de geolocalización se actualiza automáticamente en base a dos parámetros: distancia mínima desde la última actualización e

intervalo temporal mínimo. Estos parámetros pueden ser modificados desde esta interfaz de configuración.

Para implementar estas funcionalidades lo haremos en el paquete `com.Preferencias`, la principal clase que se ha empleado con el fin de recuperar y almacenar las preferencias del usuario es `SharedPreferences`.

Veamos la interfaz, el layout que el usuario final verá.

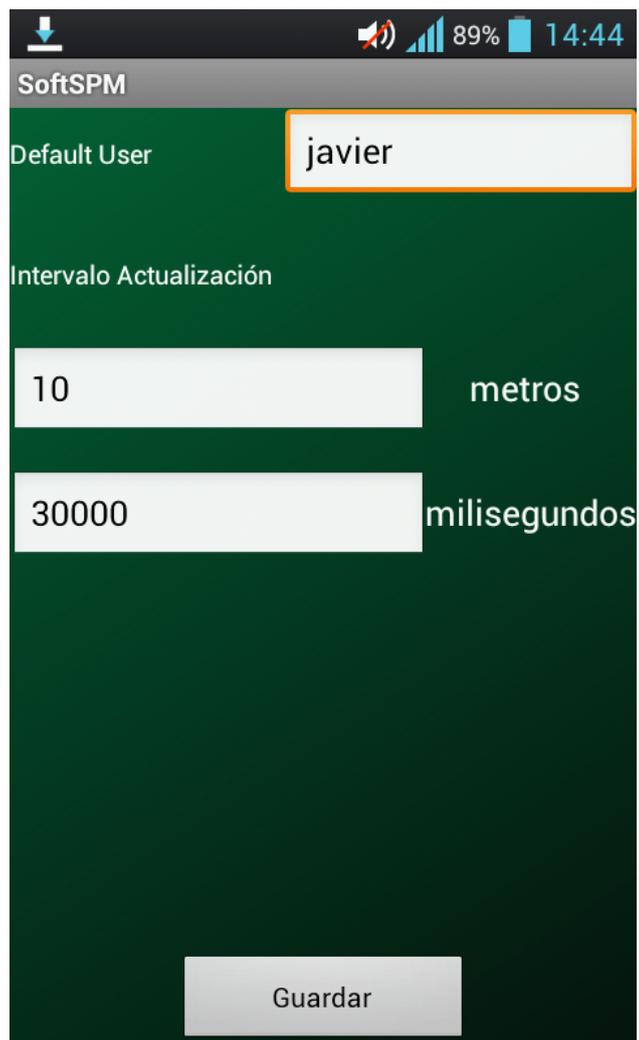


Figura 3.10: *Layout preferencias de la aplicación*

### Estado Login

Se trata de un estado intermedio de la aplicación. Está definido por una actividad cuyo funcionamiento a grandes rasgos es autenticar y verificar la identidad de un usuario antes de pasar a la

parte privado de la aplicación.

Con este fin se crea una conexión entre la aplicación Android y el servidor para realizar un intercambio de credenciales y decidir la veracidad de la información ofrecida por el usuario al servidor.

Este intercambio de información sensible por lo que ha de realizarse mediante protocolos seguros, en este caso se ha hecho uso de técnicas de cifrado para proteger la información sensible como lo es la contraseña del usuario.

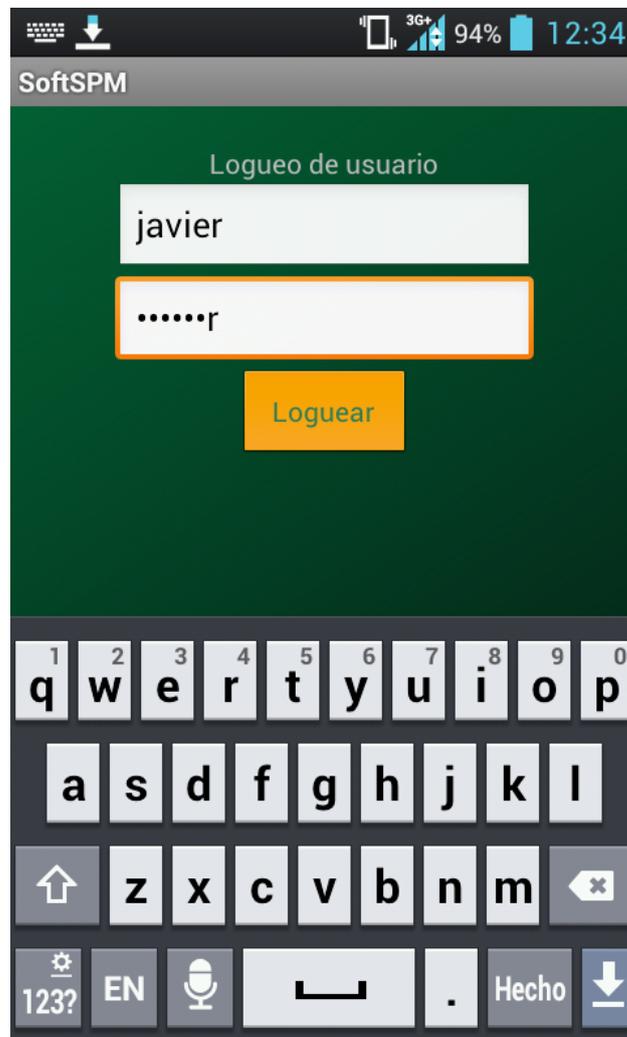


Figura 3.11: *Layout de logueo de usuario*

### Seguimiento, actividades y servicios

El seguimiento es la parte central de la aplicación. En ella se realiza la monitorización de la actividad al aire libre. El funcionamiento incluye la medición de parámetros sobre la actividad, así como una serie de elementos que permiten la seguridad de la actividad. Vamos a ver un pequeño esquema de las clases y servicios que se relacionan para esta funcionalidad, todos ellos se incluyen en el paquete com.SPM.

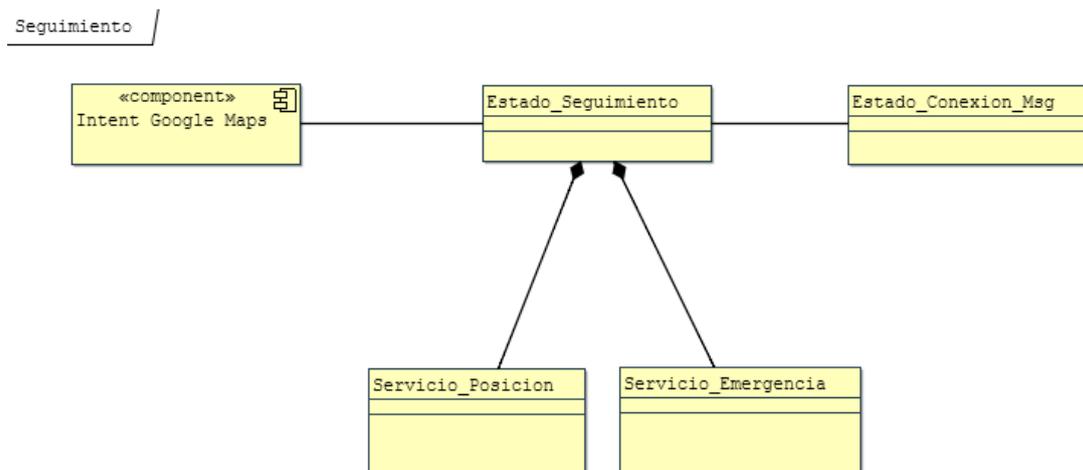


Figura 3.12: *Diagrama de clases del seguimiento de actividad*

Empecemos analizando los servicios disponibles desde esta actividad, cada uno de ellos es ejecutado en segundo plano.

- ServicioPosición: Proporciona una cobertura de geolocalización en tiempo real. Éste servicio proporcionara a la actividad principal EstadoSeguimiento la latitud, la longitud y la altitud en cada momento.
- ServicioEmergencia: Proporciona un servicio de reacción ante una emergencia. Esto incluye la emisión de una señal de aviso acústica, una llamada/mensaje a un servicio de ayuda y la publicación inmediata de una aviso de ayuda en la aplicación online para que pueda ser vista por el resto de la comunidad de usuarios.

De la misma forma continuamos viendo otra clase relacionada con EstadoSeguimiento, se trata exactamente de EstadoConexionMsg. Esta actividad permite enviar mensajes a la comunidad de

usuarios desde el propio terminal. Esta función ofrece y amplifica el componente social de esta aplicación.

Por último pero no menos importante nos encontramos con una actividad predefinida de Android, mediante los datos proporcionados por el ServicioPosición se creará una instancia de Google Maps que representara la ruta seguida. La ruta será representada desde el punto inicial hasta el punto actual o final de nuestra ruta siempre que el servicio de mapas tenga cartografiado los puntos establecidos.

A continuación podemos ver una captura de pantalla de la interfaz que el usuario final verá de la actividad EstadoSeguimiento. Una vez observemos dicha apariencia podremos hacernos una idea más real de sus funcionamiento y pretensiones.

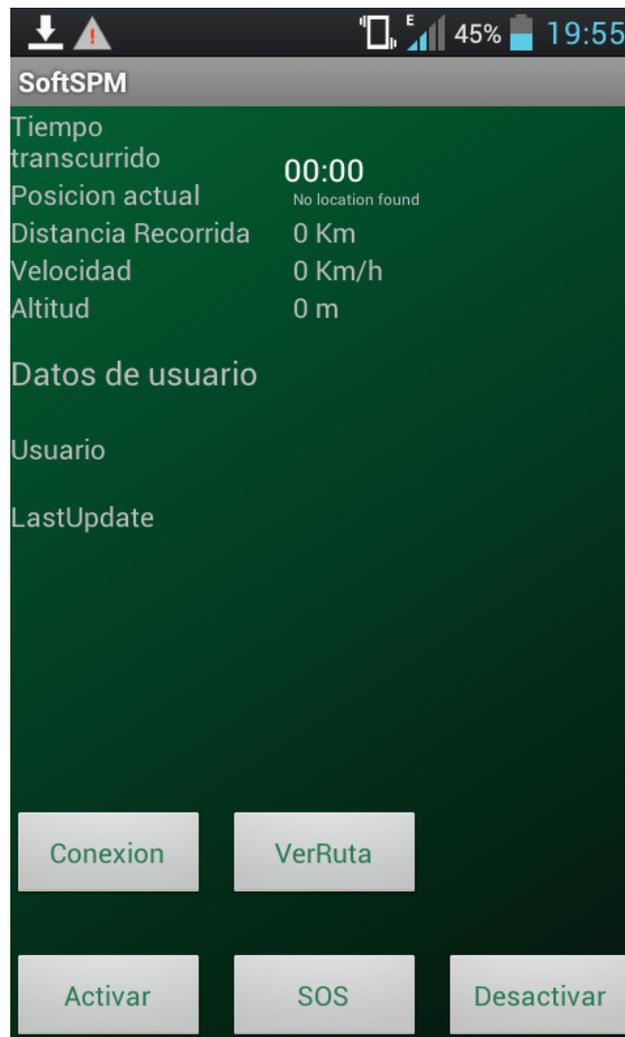


Figura 3.13: *Layout Estado Seguimiento*

A primera vista podemos ver representados algunos de los datos que la actividad suministra y que hemos comentado anteriormente, entre éstos se encuentra la latitud, la longitud y la altitud en cada momento.

En base a los datos proporcionados por el ServicioPosicion podemos calcular la distancia recorrida y la velocidad media de nuestra actividad hasta el momento. De la misma manera podemos ver información del nickname del usuario de la aplicación y la fecha de la última actualización con respecto al servidor.

A simple vista también nos encontramos con una serie de botones, la mayoría de ellos responden a elementos o servicios comentados anteriormente. Pasemos pues a comentarlos separadamente.

- **Conexión:** Abre el interfaz que permite enviar mensajes al servidor.
- **VerRuta:** Muestra la ruta en Google Maps entre la posición inicial y la actual/final.
- **Activar:** Iniciar el proceso de monitorización y seguridad.
- **Desactivar:** Finaliza el proceso de monitorización para iniciar una nueva actividad.
- **SOS:** Inicia el servicio ServicioEmergencia.

Esta actividad incluye una alerta actividad mediante movimiento, el funcionamiento de la alerta es similar a apretar el botón SOS (activar ServicioEmergencia). La diferencia fundamental radica en la forma de activar la alerta, se activa mediante detección de movimiento con el fin de simular un disparador de alerta ante una caída fortuita.

### 3.4. Aplicación Servidora

La aplicación servidora es la encargada de ofrecer una serie de servicios al resto de elementos que componen la aplicación. Estos elementos, como veíamos en el esquema de la aplicación, son la Interfaz Web y la aplicación Android.

La interacción entre estos componentes se realiza siguiendo el esquema arquitectónico representado por la arquitectura REST comentada en este mismo capítulo. En resumen, podríamos considerar que la aplicación servidora ofrece una serie de recursos que son accesibles mediante peticiones HTTP desde la Interfaz Web y desde la aplicación Android.

Por otro lado en la aplicación servidora guardamos todos los datos de usuario, rutas, mensajes, etc. Esto como es obvio hace patente la necesidad de una base de datos en nuestro servidor en el que almacenar todos los datos relevantes.

Para estructurar nuestra base de datos de forma eficiente debemos elaborar un modelo de datos que nos solucione las entidades que necesitamos representar y almacenar. Simplemente enunciar que nuestra base de datos se trata de sqlite3.

### 3.4.1. Modelo de datos

El modelado de datos de nuestra aplicación pretende dar servicio a los requisitos de entidad derivados de los objetivos del mismo. Uno de los principales problemas o hándicaps que tenemos es incorporar nuestros requisitos de diseño a algunos de los modelos predefinidos en Django, en particular en lo respectivo a la gestión de usuarios. Django incorpora por defecto el modelo de usuario, el cual incluye una serie de campos como el nombre, contraseña, etc. Esto no es suficiente para nuestro propósito. Es por esto que no sólo tenemos que crear nuevos modelos sino adaptar los ya existentes.

Para hacernos una idea de la estructura del modelado de datos resultante de la aplicación sobre Django podemos observar el siguiente diagrama o esquema relacional.

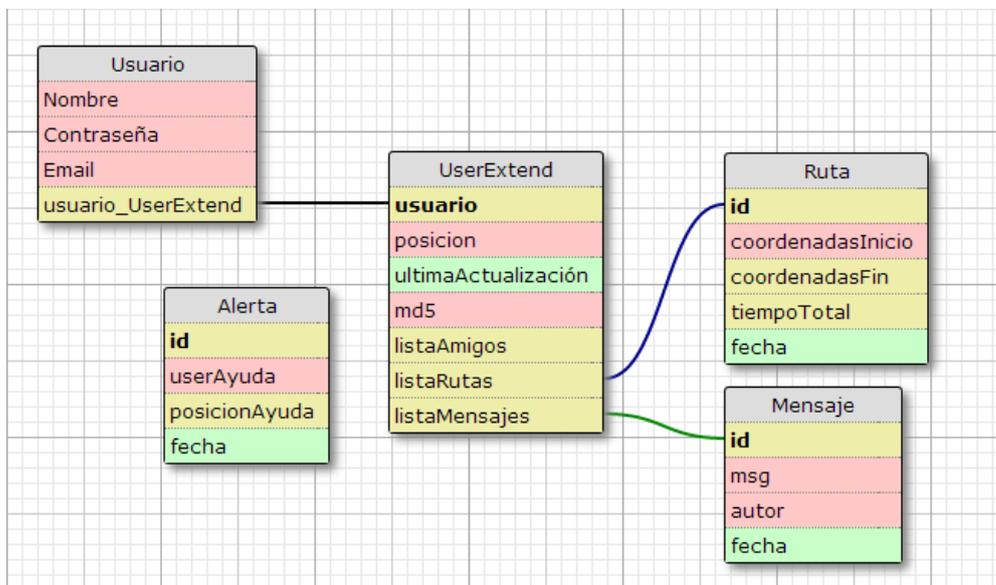


Figura 3.14: Esquema relacional del servidor

Vamos a comentar estos modelos más detalladamente:

- Usuario: Se trata, como hemos comentado, de un modelo implementado por defecto en Django. Lo usaremos extendiéndolo mediante UserExtended.
- UserExtended: UserExtended nace con la idea de añadir una serie de campos relevantes para la aplicación, como la posición o Foreign Keys para relacionar las rutas, amigos y mensajes de cada usuario.
- Mensaje: Entidad que representa a los mensajes que los usuarios pueden enviar. Se caracterizan por un id, mensaje, autor y la fecha en la que fue emitido.
- Alerta: Muy parecido a la entidad mensaje, se diferencia en que está no guarda relación con otras entidades debido a sus características de funcionamiento especiales. Se incluyen campos como nombre de usuario, posición, etc.
- Ruta: En este modelado recreamos una serie de datos que permite describir las rutas que el usuario puede realizar. Los campos principales de esta son las coordenadas de inicio y fin, el tiempo transcurrido y la fecha de realización de la actividad.

### 3.4.2. Servicio Web

El servidor de SOFTSPM cuenta principalmente con dos funcionalidades que podríamos definir como generales: servir los recursos HTML que conforman la Interfaz Web y responder a las peticiones HTTP de los recursos que este ofrece.

Este servicio Web, implementado mediante el framework de desarrollo Web Django, está constituido siguiendo un modelo estructural MVC (Modelo Vista Controlador). La característica principal del MVC es separar los datos y la lógica de la aplicación (Servidor) de la interfaz del usuario final (Interfaz Web en nuestro caso). De esta forma conseguimos una serie de ventajas derivadas de esta separación de funciones. Algunas de estas ventajas son:

- Un modelo no tiene que modificarse ante un cambio de la vista o interfaz.
- El diseño e implementación es más sencillo al dividir el trabajo.
- Mayor robustez al poder aislar componentes.
- Mayor escalabilidad del sistema.

Por el contrario este tipo de sistemas también presenta algunos inconvenientes inherentes de la separación de lógica e interfaz.

- Aumenta la complejidad del sistema al contar con numerosos elementos.
- Contamos con un gran número de elementos que debemos mantener y desarrollar separadamente.

### 3.4.3. Características y puesta en marcha del servidor

Una vez desarrollada y finalizada la implementación en Django de todos nuestros objetivos llega el momento de desplegar el sistema para ofrecer las funcionalidades requeridas. En un primer momento la aplicación es lanzada y probada en un servidor privado, sólo accesible en red local. Esto es adecuado para pruebas menores pero se hace patente la necesidad de desplegar el sistema en un nivel superior. Se decide desplegar el sistema en un servidor externo accesible globalmente desde un simple navegador Web.

Uno de los problemas que podemos encontrar es la elección de un servidor que reúna las características técnicas que la aplicación necesita. Estas necesidades son la posibilidad de creación y administración de una base de datos, soporte de Python y Django o la capacidad de almacenamiento o ancho de banda.

Tras evaluar numerosos servicios de hosting elegimos para el despliegue de nuestro sistema un servicio ofrecido por la empresa francesa `alwaysdata` [8]. `Alwaysdata` ofrece un servicio gratuito de almacenamiento que cumple los requisitos básicos necesarios, los únicos problemas planteables son el servicio al cliente debido a tratarse de una compañía francesa y a la limitación de carga del servidor. Esta limitación de carga puede llegar a ser un problema si se alcanza una tasa de carga de usuarios elevados en la aplicación.

La configuración del servidor se realiza mediante SSH y FTP para subir al servidor los ficheros de nuestra aplicación Django, tras algunos problemas en la configuración del servidor conseguimos ponerlo en funcionamiento bajo la URL `jrr123.alwaysdata.net`

## 3.5. Interfaz Web

La interfaz Web es la parte más representativa de la aplicación desde el punto de vista del usuario final. Se accederá mediante protocolo HTTP bajo una URL, en nuestro caso `jrr123.alwaysdata.net`.

Dada la importancia de esta interfaz Web se ha de poner énfasis principalmente en mejorar el diseño y la experiencia de usuario.

La idea de la interfaz Web es aunar en una misma interfaz dos tipos de servicios. El primero es recopilar y almacenar los datos de las rutas y actividades realizadas, el otro es difundir y reunir una comunidad de aficionados a las actividades al aire libre.

Cada una de las opciones de la interfaz Web representa un servicio ofrecido por el servidor en forma de recurso, siguiendo la estructura que comentamos en la arquitectura REST. En función del recurso al que se accede empleamos principalmente dos de los cuatro métodos HTTP que comentábamos, me refiero a GET y POST.

Vamos pues a recorrer cada una de las opciones accesibles desde esta interfaz, para ello vamos a hacer uso de capturas de pantalla de la interfaz.



Figura 3.15: *Interfaz Web de la aplicación*

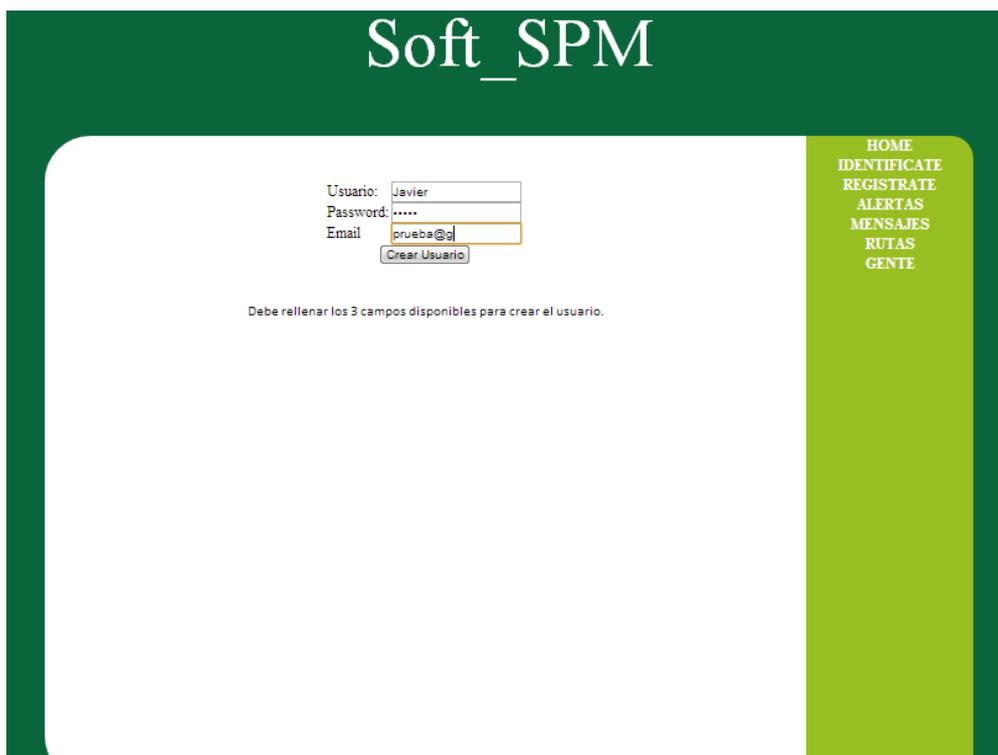
En esta primera captura podemos observar la "Home" de la interfaz Web, en ella se enumeran algunas de las características disponibles y se intuyen las opciones del menú de las que dispondrá la versión online de la aplicación.

### 3.5.1. Opciones de la interfaz Web

Veamos más en detalle las opciones más relevantes que ofrece esta interfaz, empecemos por las opciones disponibles para un usuario no identificado seguidamente de las opciones para un usuario identificado.

#### Registro e Identificación de usuario

Esta interfaz ofrece la opción a un usuario de registrarse en la aplicación, para ello se requerirán nickname, contraseña y email. Django incorpora una serie de procedimientos que simplifican en gran medida la creación de este tipo de servicios de la interfaz.



The screenshot shows a web interface for 'Soft\_SPM' with a dark green header. The main content area is white and contains a registration form with three input fields: 'Usuario:' with the value 'Javier', 'Password:' with '.....', and 'Email' with 'prueba@q'. Below the fields is a 'Crear Usuario' button. A message below the form reads: 'Debe rellenar los 3 campos disponibles para crear el usuario.' On the right side, there is a vertical green sidebar with a list of navigation links: HOME, IDENTIFICATE, REGISTRATE, ALERTAS, MENSAJES, RUTAS, and GENTE.

Figura 3.16: *Interfaz Registro de la aplicación*

Para el caso del login de usuario se ha implementado una simple rutina que compruebe el usuario y la contraseña con la base de datos de usuarios. Éste es el login de usuario que será necesario en la aplicación Android para acceder a la actividad de seguimiento.

## Perfil de usuario

Cada usuario registrado cuenta con un perfil en la aplicación. Este perfil, muy similar al de otras aplicaciones, cuenta con una serie de datos del propio usuario. Estos datos son nickname, email de contacto, la última posición del usuario conocida y la fecha de la última conexión así como un mapa de la zona posicionado sobre la última posición conocida.



Figura 3.17: Perfil de usuario de la aplicación

## Visualización de Alertas

Cada vez que un usuario está realizando una actividad y se encuentra en el modo seguimiento de la aplicación puede lanzar una alerta bajo determinadas circunstancias. Estas alertas, aparte de otras funcionalidades que enumeramos anteriormente, incluye la funcionalidad de enviar al servidor un mensaje de auxilio. Esta alerta estará disponible para todos los usuarios, su composición es la posición (latitud y longitud) donde se necesita ayuda y el usuario que lo necesita. Para ilustrar la alerta se incluye un mapa con la posición desde donde se lanzó la alerta.

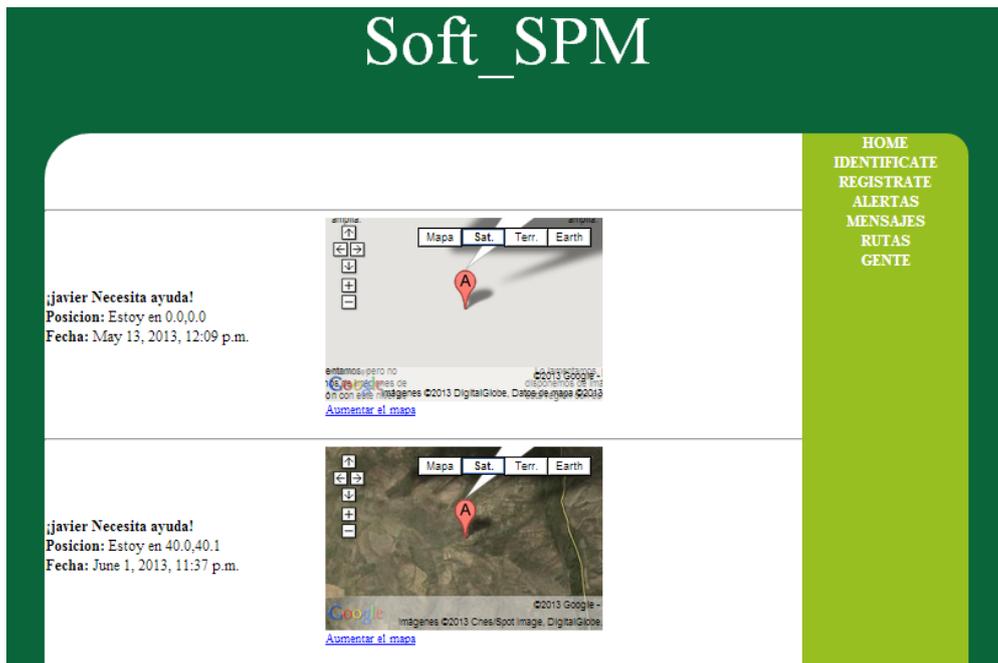


Figura 3.18: *Interfaz de alertas de la aplicación*

## Visualización de mensajes

La aplicación, tanto desde el terminal móvil como desde la interfaz, es capaz de enviar mensajes entre usuarios, siguiendo una distribución muy similar al timeline de Twitter. De esta forma un usuario puede ver los usuarios de determinados usuarios en función de sus preferencias.

Estas preferencias de visualización son configurables, por un lado podemos ver los mensajes sin filtrar, es decir, todos o sólo aquéllos de los usuarios clasificados como amigos.

De cada mensajes se muestra una serie de datos para mejorar su interpretación, estos datos son el miembro que lo envió, la fecha que lo hizo y el propio contenido del mensaje. Los mensajes están ordenados por fecha de envío.

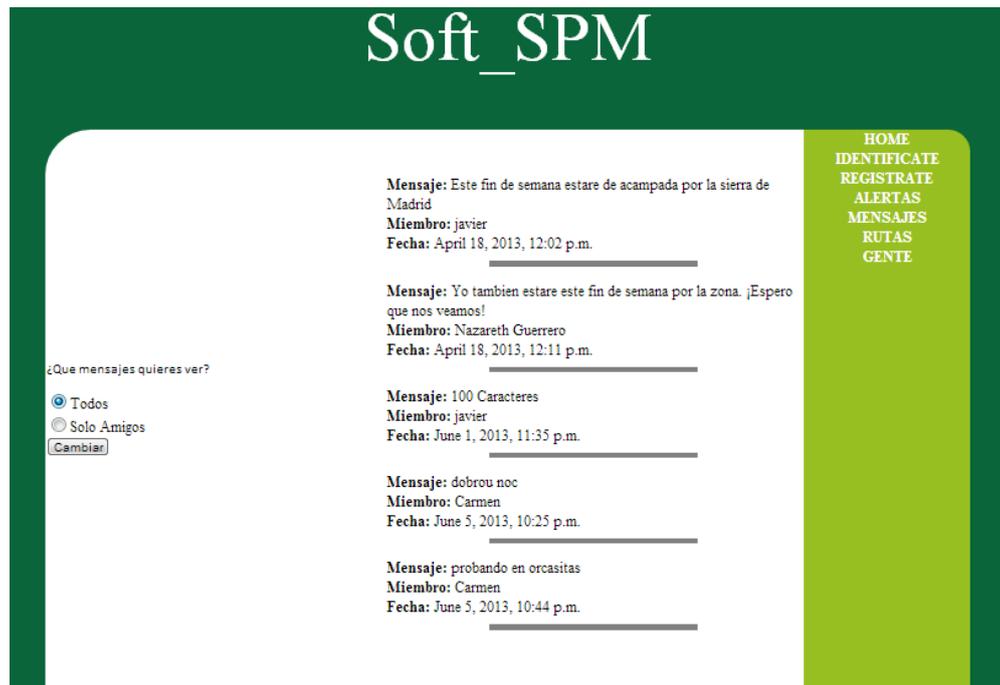


Figura 3.19: *Interfaz Mensajes de la aplicación*

## Rutas

En esta sección tenemos disponibles todas las rutas que el usuario ha realizado, se presenta la ruta de forma gráfica. Se incluyen varios parámetros de la ruta como el tiempo invertido, la distancia total y otros parámetros típicos de la ruta, de igual forma para ilustrar la ruta se representa un mapa de la zona en la que la ruta se ha realizado.

Con esto conseguimos preservar las rutas que el usuario ha ido haciendo durante su trayectoria en la aplicación junto con datos importantes sobre las mismas. Las rutas están ordenadas en orden cronológico creciente.



Figura 3.20: *Interfaz Rutas de la aplicación*

## Gente

Un usuario puede añadir a sus amigos a la aplicación, para añadir a un usuario simplemente se debe introducir su nombre de usuario en el interfaz. Con esta característica obtenemos facilidades de previsualización de algunos de los servicios comentados anteriormente como por ejemplo la discriminación de mensajes para un determinado usuario.

El motivo principal de la implementación de este módulo es incorporar una función social. El intercambio de información entre usuarios fomenta la interacción entre los distintos usuarios.



Figura 3.21: *Interfaz Amigos de la aplicación*

## Envío de mensajes

Esta función ofrece la opción de redactar y enviar mensajes, los cuales estarán disponibles para su posterior visualización en la sección correspondiente. Podrán ser filtrados por autor posteriormente en la interfaz adecuada.

Con este recurso ya son dos las opciones de envío de mensajes que ofrece al usuario el conjunto de la aplicación. Si recordamos en la aplicación Android ya incorporábamos una función similar de envío de mensajes análoga a esta incorporada sobre la interfaz Web.

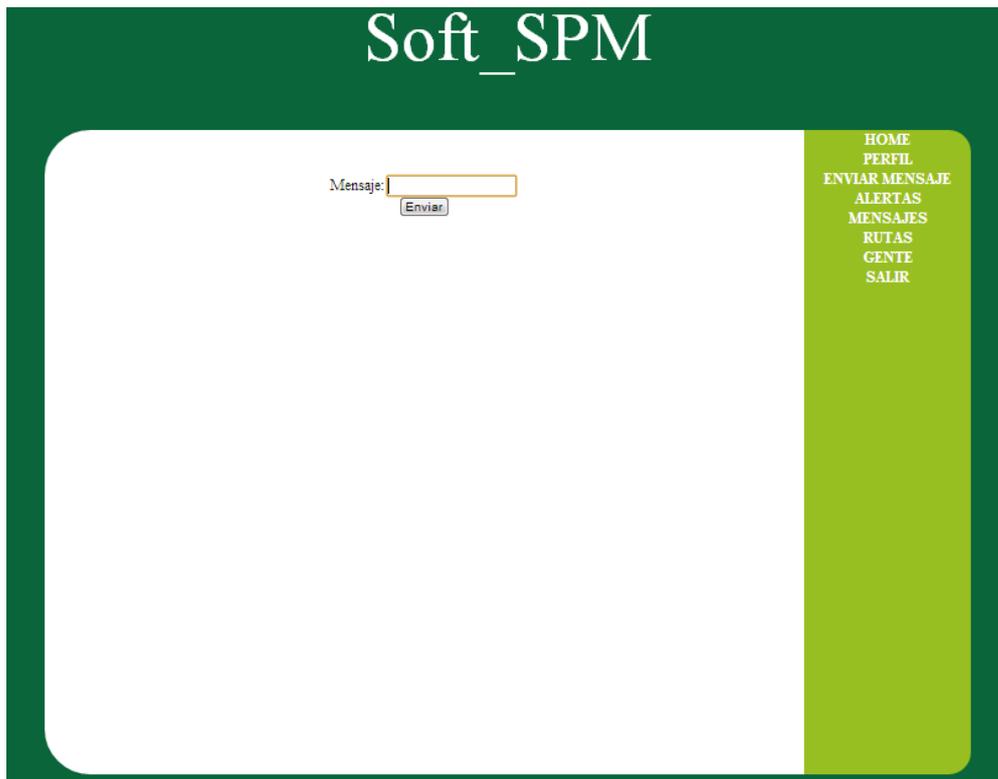


Figura 3.22: *Interfaz Envío Mensajes de la aplicación*

### 3.6. Implementación de técnicas de difusión en buscadores

Como ya enunciamos existen dos tipos de vertientes para mejorar el posicionamiento en buscadores, nosotros nos centraremos en la vertiente técnica debido a la naturaleza del proyecto. Estas modificaciones van en la línea de adaptar nuestro código HTML de la interfaz Web con la intención de mejorar la opinión de los buscadores en lo respectivo a nuestra página Web.

Antes de empezar con la implementación propiamente dicha de estas técnicas de posicionamiento, es muy interesante analizar el mercado de este tipo de aplicaciones. Este análisis nos permitirá elegir unos términos de negocio de acuerdo a nuestras necesidades, por ejemplo, nuestra posición en Google respecto al término **actividades de montaña**. Google pone a nuestra disposición una serie de herramientas que nos permiten extraer los hábitos de búsqueda de los usuarios. Examinemos la herramienta de Google, Google Adword Tools, se trata de una herramienta empleada en marketing online y que nos indica el volumen de búsqueda para una combinación de palabras clave. Para esta aplicación hemos tomado tres palabras sencillas para mostrar el proceso

de selección.

Palabra clave	Competencia	Búsquedas globales mensuales ?	Búsquedas locales mensuales ?
actividades de montaña ▾	Baja	2.900	880
actividades al aire libre ▾	Baja	12.100	2.900
actividades deportivas ▾	Baja	49.500	27.100

Figura 3.23: *Estudio de palabras clave de negocio*

Como vemos en la imagen anterior tenemos el volumen de búsqueda para tres términos de negocio relacionados con nuestro proyecto. Sería lógico pensar que es mejor orientar nuestro proyecto Web en la dirección que nos proporcione un mayor nivel de búsquedas. En términos generales esto es así, aunque debemos de prestar especial atención a una serie de elementos como por ejemplo la competencia. Una vez, con la herramienta anterior, hemos elegido nuestro termino de negocio sólo queda orientar nuestra interfaz Web hacia este término para hacer a los buscadores interesarse por nuestro sitio Web. Empecemos por una serie de elementos que debemos modificar en el código HTML de la plantilla, estos elementos son:

- Title: Se trata del código HTML entre las etiquetas **title**, en este etiqueta es recomendable repetir al menos una vez la palabra clave elegida. La longitud máxima no debe superar los 60 caracteres.
- Description: Una breve descripción explicativa de nuestro sitio Web incluyendo la palabra clave al menos una vez. Longitud máxima 145 caracteres.
- Cabeceras: Se corresponde con las etiquetas h1, h2, h3. Es recomendable establecer la palabra clave y derivaciones de la misma.

```
<title>Soft_SPM - App Actividades Deportivas</title>
<meta name="description" content="SOFTSPM es una aplicación de monitorización y seguimiento para actividades al aire libre" />
<meta name="keywords" content="Aplicacion, seguimiento, monitorizacion, montaña, seguridad, aire libre" />
```

Figura 3.24: *Optimización de metadatos HTML*

- Metatags: Directamente se incluyen 5 -10 palabras clave interesante para el proyecto.

Con estas pequeñas modificaciones seremos mucho más amigables con los buscadores, con lo que las probabilidades de aparecer en las primeras páginas de resultados es más factible.

### 3.7. Problemas afrontados

Como en todo proyecto, y aún cuando desde un principio se estableció una serie de pautas para seguir una metodología de trabajo, nos encontramos con problemas. Problemas fáciles y otros más complicados. Vamos a ver, estructurándolo por partes, los principales problemas que hemos encontrado en cada parte.

#### Aplicación Android

En la aplicación Android los problemas principales han venido de la mano de la puesta a punto de los distintos sensores que la aplicación emplea. En la brújula aumentada se emplean los sensores magnéticos y acelerómetros, los cuales han sido un quebradero de cabeza para conseguir que funcionen como se espera de ellos.

De la misma manera en lo relativo a los sensores de geolocalización mediante GPS empleados en el seguimiento de la aplicación. Se pedía que estos sensores proporcionaran la posición y altitud cada cierto intervalo. El problema radica en el gasto de batería que posibilita esta función. Para intentar mejorar esto se separó esta funcionalidad en un servicio que se ejecuta de forma secundaria respecto a la aplicación. De esta forma el servicio ejecuta el sensor GPS y simplemente envía la información necesaria a la actividad de seguimiento principal.

#### Servidor Django

El servidor Django ha sido diseñado con el fin de cumplir sus objetivos de la forma más sencilla posible. Esto a priori no ha presentado demasiados problemas hasta la hora de desplegar el sistema en un servidor externo. Al ir a desplegar el sistema, hemos encontrado numerosos problemas de compatibilidad entre versiones, tanto de Python como de la base datos que se utiliza por obligación en el servidor. Esta adaptación entre versiones ha obligado a cambiar algunas de las funciones empleadas al no estar disponibles debido a la versión.

Por otra parte se han tenido que readaptar muchas de las rutas empleadas internamente en la aplicación al desplegar en servidor externo, sobre todo al servir archivos estáticos o imágenes.

## Interfaz Web

La Interfaz Web como hemos visto está implementada en HTML y CSS. En mi opinión personal en este punto el problema principal es conseguir elaborar una interfaz simple pero efectiva de cara al usuario final. Una interfaz atractiva que permita cumplir los objetivos establecidos. Para conseguir estos objetivos empleamos hojas de estilo en cascada que nos permiten agrupar estilos de diseño a módulos HTML que queremos que sigan un mismo estilo.

## 3.8. Conclusiones y experiencias

Esta sección tiene como finalidad mostrar una serie de conclusiones y experiencias derivadas del trabajo en la aplicación. Este trabajo será evaluado desde el punto de vista del diseño y la implementación de las distintas partes de SOFTSPM. Como bien hemos visto la aplicación fue segmentada en tres partes claramente diferenciadas, cada una de ellas con una serie de características específicas.

Uno de los puntos que considero más destacable es el diseño y agrupación de funcionamiento de tecnologías que a priori son muy distintas. Esto permite evaluar los pros y contras de cada una de ellas así como el acoplamiento entre las mismas.

Durante este capítulo analizamos separadamente las tres partes principales, que si las recordamos eran la aplicación Android, el servidor en Django y la interfaz Web en HTML. Cada uno de ellos fue explicado desde la perspectiva de diseño, así como de funcionalidad de usuario. Como vimos cada una de las partes tiene su funcionalidad pero todas ellas giran en torno al servidor, el cual, pone a disposición del resto de elementos sus recursos.

Tratando de forma global el funcionamiento de la aplicación podemos verlo de una forma muy general desde dos perspectivas diferentes.

- Monitorización y protección de actividades con intercambio de datos con el servidor.
- Acceso y compartición de los datos disponibles en el servidor.

En resumen, el servidor ofrece unos recursos mediante distintos tipos de peticiones HTTP, variando en función de las necesidades de la aplicación y el usuario. Los recursos y funcionalidades exactas de la aplicación se han intentado plasmas en esta memoria a lo largo de este capítulo de la forma más sencilla y visual posible.

Una vez visto el funcionamiento y diseño de la aplicación podemos decir que resulta cuanto menos interesante ver en funcionamiento y encajando como un puzzle las piezas que forman SOFTSPM. En cuanto al diseño propiamente dicho de la parte visual de la aplicación sí que podemos comentar que existe ámbito de mejora, sobre todo en el aspecto HTML de la interfaz Web de forma que sea mucho más atractivo para la vista del usuario final. No obstante creo que el funcionamiento cumple los objetivos propuestos sobre la aplicación. Para terminar esta parte sobre la estructura de la aplicación me gustaría hacer hincapié en el empeño que se ha tenido por hacer seguir al lector de forma fácil todo el proceso

# Capítulo 4

## Resultados

Durante el capítulo 3 se explicaron el diseño y la implementación de los elementos que forman la aplicación SOFTSPM. En este capítulo cuarto veremos los resultados obtenidos en diversas pruebas realizadas sobre la aplicación con el fin de corregirla y mejorarla. Estas pruebas, para obtener una mínima independencia de criterio, han de ser realizadas por personas ajenas al proyecto. Estas personas (usuarios) se han de centrar en el funcionamiento de SOFTSPM desde el punto de vista de un usuario final abstrayéndose completamente de su diseño e implementación. Es decir, como cualquier usuario potencial consumidor de este tipo de aplicaciones.

Para facilitar esta labor de prueba al usuario se ha realizado un pequeño manual de usuario (apéndice A) de la aplicación SOFTSPM. En este manual se explican las funcionalidades principales de la aplicación. En mi opinión este manual presenta las suficientes garantías para resolver las principales dudas de uso y orientar al usuario en el funcionamiento de la aplicación.

A cada usuario que testee la aplicación se le ofrecerá por un lado el manual de usuario de la aplicación, el .apk de la aplicación Android y en última instancia una encuesta para poder analizar sus conclusiones tras la prueba. En algunos casos se ha prestado soporte a la hora de instalar la aplicación Android en usuarios con una baja experiencia en este tipo de dispositivos Android.

Una vez entregado el material comentado a los usuarios se les invita a probar la aplicación durante la realización de actividades en el exterior preferiblemente. Al final del periodo de prueba se les entregaran una serie de cuestiones a modo de encuesta para obtener los mejores datos posibles sobre la experiencia de usuario que estos han experimentado.

## 4.1. Primeras pruebas

Una vez hemos finalizado la implementación, y antes de hacer pruebas con usuarios externos al proyecto, es necesario crear un plan de actuación. Este plan tiene como finalidad probar la aplicación con unas mínimas garantías y eficiencia. Este plan de actuación intenta recorrer todas las acciones que un usuario estándar podría realizar en la aplicación, de esta forma establecemos un primer nivel de depuración de errores.

El objetivo de este tipo de procesos es encontrar algunos errores de programación básicos. De esta forma podemos decir que corregimos los errores de bulto de la aplicación. La forma de actuar es seguir el flujo de ejecución de la aplicación para obtener una serie de resultados. Por ejemplo, un flujo de ejecución para registrar un usuario sería el acceso a la página principal del proyecto, una vez en la página principal seleccionar la opción de registro. Una vez aquí tenemos dos posibles opciones, éstas son: introducir unos datos correctos o introducir unos datos incorrectos (campo sin rellenar, usuario ya existente, etc.). De esta forma, estamos siguiendo el flujo de ejecución que un usuario podría seguir y desvelar cualquier error relacionado con este flujo.

Durante estas primeras pruebas encontramos errores sobre todo en la interfaz de seguimiento de la aplicación Android. Estas pruebas dieron como resultados algunas modificaciones en la forma de medir la distancia y sobre cómo hacerlo para solventar algunos errores de medición. De la misma forma durante el seguimiento fueron necesarias muchas pruebas para conseguir ajustar la sensibilidad de los sensores de movimiento que lanza el servicio emergencia.

En definitiva, tras estas pruebas, tenemos una versión preliminar de la aplicación, una aplicación lo suficientemente decente y funcional para ser mostrada al conjunto de usuarios que testearan la aplicación.

## 4.2. Pruebas de usuarios

La finalidad de esta sección es la de obtener opiniones y consejos sobre usuarios ajenos al proyecto que nos atañe y de diferente opinión. Como veíamos en el preámbulo de este capítulo se le otorgó a cada usuario una serie de elementos (manual, instalador y encuesta) para realizar las pruebas con garantía.

Los únicos requisitos eran que el usuario tuviera un terminal con Android 2.1 o superior y conexión a Internet en el terminal, algo habitual hoy en día. Como nota, poner de manifiesto que

a más de un usuario fue necesario instalarle el .apk de la aplicación personalmente.

Al final de las pruebas, a cada usuario se le envió una encuesta con una serie de preguntas (disponible en el apéndice B). Sobre estas respuestas extraemos conclusiones sobre posibles mejoras o ampliaciones para la aplicación SOFTSPM.

Veamos algún resumen sobre cada uno de los cuatro usuarios que realizaron las pruebas. Enunciemos también las preguntas que se le hicieron al usuario en la encuesta proporcionado.

- ¿Cuál ha sido el contexto de uso a la aplicación en su caso particular?
- ¿Qué dispositivo móvil (modelo) y qué tipo de conexión emplea?
- ¿Ha leído el manual de usuario antes de empezar a usar la aplicación?
- ¿Qué opinión le merece la aplicación?
- Si tuviera que añadir algo de la aplicación, ¿qué sería?
- ¿Ha tenido algún problema durante el uso de la aplicación?
- ¿Qué puntuación le daría objetivamente a la aplicación?

### Usuario 1

El primer usuario de la aplicación ha realizado la prueba con un dispositivo Android LG Optimus L7, con Android 4.0.3. Las opiniones del usuario son en general buenas, se indica que se ha realizado la prueba mientras se caminaba en una de las rutas de casa al trabajo durante aproximadamente 20 minutos en un par de ocasiones.

Como puntos positivos el usuario comentó la especial utilidad de la aplicación Android con la integración del mapa con los PDIs y la brújula aumentada.

Como problema principal el usuario comentó del asunto de gasto de batería que el modo seguimiento supone. Esto hizo volver a revisar el modo de implementación de esta parte, desencadenando una optimización del servicio GPS. Sobre todo en lo referente a los intervalos de actualización del propio GPS para intentar gastar la mínima batería posible. Hemos de reconocer que el servicio GPS de por sí toma una gran cantidad de batería, siendo uno de los principales handicap del proyecto.

Este handicap reduce la autonomía del teléfono en la fase seguimiento a unas 7 u 8 horas, algo muy similar a lo que proporcionan otras aplicaciones similares o servicio GPS para automóviles. Esto puede ser en determinados casos insuficiente. En general la opinión del usuario ha sido satisfactoria.

### Usuario 2

El usuario 2 por su parte empleo un dispositivo Android Samsung Galaxy SIII, exactamente la versión 4.1.2.

El usuario empleó la aplicación durante una salida de fin de semana de alrededor de 2 horas. Este lugar destacó porque la conectividad de internet móvil no ha sido constante durante todo el recorrido.

El usuario ha comentado como problema la calibración del sistema de detección de caídas, y es que tras diversas comprobaciones depende en gran medida del terminal móvil. Esto provoca mayor o menos sensibilidad dependiendo del terminal empleado. La solución a esto pasa por recalibrar a un nivel mínimo el sensor, de esta forma sacrificamos posibles caídas menores (que no alcancen el nivel mínimo establecido) pero eliminamos en gran medida la posibilidad de falsos positivos.

Por otra parte el usuario indica la imposibilidad de ejecutar el servicio de seguimiento sin conexión a Internet. Esto hizo pensar en buscar alternativas. Por un lado, es lógico ya que el servicio emplea la conectividad a Internet para guardar los datos de la ruta y el envío de mensajes/alertas. Por otro lado, sí que podría ser útil dar la posibilidad de, en ausencia de línea, poder medir la distancia y el tiempo. Su adaptación no tiene mayor problema, simplemente se elimina la autenticación de usuario si no se dispone de conexión a Internet.

Este usuario recomendó como opción adicional un módulo para interactuar con la cámara deportiva GoPro. La opinión general es satisfactoria.

### Usuario 3

El usuario 3 cuenta con un terminal idéntico al usuario 2, Samsung Galaxy SIII con Android 4.1.2. Este usuario no ha especificado ninguna crítica en particular a la aplicación durante su uso. Por otro lado el usuario indica que el uso y empleo de la aplicación ha sido satisfactorio, simplemente añade la posibilidad de realizar también una versión cliente para iOS. Este punto es algo que, de momento, queda fuera del proyecto, pero que puede ser interesante si se quiere llevar la aplicación a un nivel superior.

### Usuario 4

El usuario 4 realizó la prueba con una terminal Nexus 4, con Android 4.2, el más moderno hasta el momento de los terminales empleados en las pruebas.

El usuario indica por un lado que la URL de la interfaz Web podía ser más intuitiva que la que existe actualmente. El usuario tiene toda la razón; esto es algo que se tiene en cuenta para posibles futuras ampliaciones.

El usuario también indica que en algunos momentos la aplicación se ha mostrado lenta en algunos servicios, sin indicar exactamente cuáles fueron dichos servicios. En cuanto a la interfaz Web, el usuario ha encontrado un enlace que no funcionaba como debería y que ha sido solucionado de inmediato.

Como opinión final el usuario indica que es positiva y que, con un poco más de desarrollo gráfico, tiene mucho potencial.

### 4.3. Reflexiones y soluciones

Una vez realizado las pruebas expuestas anteriormente, podemos decir que la aplicación ha mejorado, no sólo en la corrección de usuario, sino también mejorando la experiencia de usuario con alguna de las indicaciones que los usuarios han transmitido al proyecto.

Con estas pruebas podemos dar por finalizadas las pruebas del proyecto. Estamos seguros de que aún podrían existir algunos errores que se nos han pasado por alto a nosotros o a los usuarios. En este momento podemos expresar que la aplicación SOFTSPM cuenta con una versión más robusta que las versiones anteriores.

En cuanto a la opinión general de los usuarios podemos decir que ha sido positiva. Esto es en parte reconfortante. La aplicación cubre las necesidades para las que fue ideada y ofrece al usuario herramientas útiles. Hemos de tener en cuenta algunas de las opiniones negativas sobre diseño o funcionamiento de la aplicación.

Los errores encontrados por el usuario han sido muy útiles para poder solventarlos sin mayor retraso. En cuanto a las recomendaciones de posibles funcionalidades que los usuarios proponen implementar, se ha tomado buena nota. Algunas no son difíciles de llevar a cabo, por lo que pueden ser llevadas a cabo en futuras posibles expansiones.



## Conclusiones

Este capítulo cerrará esta memoria de proyecto. Durante toda esta memoria se ha intentado hacer ver al lector cada una de las decisiones tomadas, tanto en diseño como en implementación. Un resumen del proceso que se ha seguido desde las primeras ideas y motivaciones iniciales hasta el despliegue y pruebas de usuarios final.

Durante este capítulo veremos los objetivos que se han llevado a cabo en comparación con los objetivos planteados al principio de esta memoria. Para acabar este capítulo realizaremos una autocrítica y análisis personal de lo que este proyecto ha significado. También considero importante y relevante enumerar algunos posibles campos de ampliación de este proyecto, el cual, como veremos, tiene bastantes interesantes ampliaciones en campos muy diversos.

### 5.1. Análisis de objetivos

Durante el capítulo introductorio pusimos de manifiesto una serie de objetivos que nos planteamos conseguir al realizar este proyecto. Veremos, si al final de este proyecto hemos cumplido los objetivos. Recordemos rápidamente cuáles eran los objetivos generales establecidos.

- Desarrollar una estructura funcional capaz de prestar un servicio.
- Afianzar conocimientos sobre programación Android y Python.
- Presentar una interfaz Web apta y funcional.
- Perfeccionar el uso de  $\text{\LaTeX}$ .

Comenzando por la aplicación propiamente dicha podemos ver que se ha desplegado una aplicación completamente funcional con un objetivo claro: el de monitorización y ayuda a usuarios en actividades de exterior. La aplicación finalmente es capaz de monitorizar actividades al aire libre, con lo que podemos trazar la ruta que se ha seguido y algunos parámetros adicionales de la ruta (tiempo, altitud, etc.). En este aspecto podemos afirmar que se ha cumplido el objetivo previsto inicialmente.

En cuanto al campo de ayuda y soporte en actividades desarrollamos servicios diversos que pueden ser de utilidad ante situaciones de riesgo o necesidad. Uno de estos servicios es la brújula aumentada, la cual presta un gran servicio en cuanto a la orientación en el medio exterior o seguimiento de una ruta determinada. De forma parecida la cartografía del área en la que nos situamos, que no sólo nos sirve para orientarnos sino que también con el sistema de PDIs (Puntos de interés) permite localizar determinados servicios de interés para el usuario.

Si nos centramos particularmente en servicios de protección activa, veremos cómo los sensores de movimiento disparan los servicios de emergencia de la aplicación ante caídas o movimientos muy bruscos. Estos servicios pueden ser de gran utilidad en situaciones complicadas.

Para acabar esta pequeña evaluación de estos servicios podemos concluir que los objetivos previstos inicialmente se han cumplido de forma satisfactoria. Algunos objetivos en mayor grado, otros en mejor grado, pero de forma satisfactoria. Durante esta memoria se han ido explicando si se han conseguido estos objetivos y esperamos que el propio lector haya conseguido extraer toda la información e impresiones necesarias para concluir en este sentido.

Ahora pasemos al objetivo de crear un sistema que permita consultar y compartir las rutas que hemos obtenido en la realización de la actividad.

Nuestro servicio Web cuenta con diversas funcionalidades. Una de ellas es recabar la información sobre las rutas realizadas con el dispositivo Android, es decir, tenemos una persistencia de esta información. Por otro lado la aplicación da la oportunidad al usuario de enviar y recibir mensajes a nuestros amigos. De esta forma fomentamos un intercambio de información con el consiguiente componente social en el servicio Web.

Evaluando pues en orden global la aplicación SOFTSPM podemos concluir que se han cubierto todas las funcionalidades inicialmente previstas, es cierto que algunas de ellas quizás no hayan sido lo más estilosamente diseñadas.

Otro de los objetivos derivados de este proyecto, los cuales destacan en el campo educativo, es la capacidad de síntesis y redacción de todo el proceso de diseño e implementación. De este objetivo,

esta memoria es un buen ejemplo. La memoria que el lector puede leer esta realizada sobre L<sup>A</sup>T<sub>E</sub>X, con el cual hasta el momento no había tenido mucho contacto. De esto podemos derivar que ha sido un objetivo extra en la realización de este proyecto.

Es importante poner de manifiesto que detrás de la aplicación, existen varias tecnologías de diferentes campos. Algunos de estas tecnologías han sido, como bien el lector sabe, Android, Python, HTML e incluso Java. Muchos de estos lenguajes han sido vistos durante mi aprendizaje en la universidad, otros como Android ha sido por iniciativa propia. Algo que podemos asegurar es que durante la realización de este proyecto se ha profundizado en muchos aspectos que quedan fuera de los típicos métodos educativos, por poner un ejemplo, personalmente me ha resultado de gran interés y utilidad el empleo de los sensores.

Los sensores en Android, o cualquier otro módulo ponen de manifiesto una serie de requerimientos. Estos requerimientos han de ser afrontados para conseguir con garantías el éxito en el trabajo. Para saber cómo afrontar estos factores es esencial una buena documentación de la tecnología con la que trabajamos. Una vez obtenida esta información, simplemente un momento de reflexión, pruebas y más pruebas hasta que somos capaces de trabajar fluidamente con dichas tecnologías.

En algunos momentos cumplir con los objetivos establecidos no ha sido sencillo. En ese momento muchos de nosotros echamos manos de la red para intentar buscar una solución. Es en este preciso momento cuando más se agradece trabajar con tecnologías de dominio público y con una gran extensión. Esto tiene inmediatamente una gran ventaja: una comunidad de cientos de usuarios de distintos niveles en los que poder apoyarse ante imprevistos.

Ahora, desde la perspectiva post-implementación, podemos decir que uno de los objetivos derivados ha sido obtener, entender y seleccionar la documentación necesaria.

Recapitulando las líneas anteriores, se presentaron unos objetivos y unas funcionalidades iniciales que el proyecto debía de ser capaz de cubrir. La forma de cubrir estos objetivos puede ser mejor o peor, pero las funcionalidades ofrecidas por la aplicación cubren los objetivos iniciales. Durante el capítulo 3 se desarrollaron los procesos empleados para cumplir estos objetivos.

## 5.2. Autocrítica

Tras hacer una valoración de lo que se ha conseguido en el proyecto y de lo que se pretendía conseguir antes de empezar este proyecto veremos una valoración personal del conjunto de este proyecto. Una valoración que incluirá tanto, las funcionalidades conseguidas como algo que considero

muy importante, la forma de conseguirla.

El desarrollo de la aplicación ha sido, salvo algunas partes muy puntuales completamente desde cero. Esto ha hecho posible un control y una toma de decisiones muy adaptables para enfocarlo al objetivo general del proyecto.

En esta línea, podemos decir que el diseño de la aplicación ha ido en la consecución de obtener un funcionamiento efectivo pero sobre todo sencillo e intuitivo para el usuario final de la aplicación. Esto es, en mi opinión, algo fundamental ya que el usuario de la aplicación es francamente probable que tenga un conocimiento técnico limitado o nulo. Este factor lo vemos por ejemplo en el diseño de la interfaz: una interfaz poco intuitiva puede crear en el usuario cierto desinterés en la aplicación. Por lo tanto desde una crítica interna se podría decir que la interfaz gráfica podría ser más simplificada y vistosa, con el fin de despertar un mayor interés en el usuario.

En cuanto al funcionamiento interno de la aplicación, se ha hecho un gran empeño en ser lo más robusto frente a errores, tanto internos como debidos a una interacción no prevista del usuario. Una vez dicho esto, también es lógico asumir que se ha podido pasar por alto algunas situaciones que pueden desencadenar errores de funcionamiento, por ejemplo, alguna petición errónea, algún recurso no disponible o disponible cuando no debería serlo, etc.

También es cierto que se ha desarrollado un proceso de pruebas con el fin de detectar y corregir todos estos lógicos y factibles fallos.

Otros de los factores, esta vez sobre la aplicación Android, que han despertado más mi interés es la administración energética del dispositivo Android durante el seguimiento de una actividad. Cuando la aplicación se encuentra en este estado provoca un gran gasto de batería del terminal. Este gasto, por una parte, es lógico debido a los sensores que se están empleando pero se ha intentado reducir empleando una arquitectura diferente mediante el empleo de servicios. De la misma manera se incluyeron opciones que permiten bajar el intervalo de actualización del GPS con el fin de ahorrar batería.

En cuanto a la aplicación general, creo que cumple de sobra los requisitos empleados. En mi opinión es una aplicación razonablemente robusta, una aplicación funcional y útil para un gran público.

Actualmente se encuentra desplegado en un servidor externo. No obstante este servidor es un servidor gratuito limitado. Esto provoca una cierta sensación de lentitud en el usuario cuando hay un número medio-bajo de usuarios en la aplicación. Por iguales razones la URL de la aplicación no es nada amigable ni para los buscadores ni para el usuario final debido, de nuevo, a la gratuidad

del servicio.

Como conclusión final creo que se trata de una aplicación lo suficientemente amplia y robusta para cubrir las necesidades del usuario para lo que fue diseñada e implementada.

### 5.3. Futuras extensiones de la aplicación

Uno de los factores que más interés ha despertado personalmente es la gran capacidad de extensión que esta aplicación puede tener. Existen numerosos y muy diversos campos donde estas actuaciones pueden llevarse a cabo. Durante la implementación de este proyecto han ido surgiendo muchos de estas líneas de trabajo, algunas de ellas se han aplicado. Muchas otras se han dejado de lado por cuestiones de plazo y extensión.

Es por estas razones que este apartado sirve para dar salida a todas esas ideas, que son o serán de utilidad y no están implementadas actualmente. Con esto se abre la puerta a una futura ampliación del proyecto a cualquier que lo considere oportuno.

Vamos pues a enumerar una serie de ideas, desarrollándolas ligeramente con la intención de que el lector tome una idea aproximada de cuál es la forma y el trasfondo.

Una de estas ideas, quizás la más fácilmente realizable, es la de llevar a cabo un despliegue de la aplicación óptimo. Este despliegue tendría un nombre de dominio adecuado a la actividad de la aplicación. En lo relativo al hosting es necesario contratar un servicio de alojamiento que soporte un número mínimo de usuarios online. Con esto conseguimos desplegar un servicio serio que permita desarrollar todo el potencial de la aplicación.

Otro de los puntos que personalmente más se me ha pasado por la cabeza es, sin duda, el empleo de dispositivos externos. Dispositivos que permitan a la aplicación Android obtener una mayor cantidad de datos, por ejemplo, las pulsaciones del usuario de la aplicación. Si contamos con este tipo de dispositivos somos capaces de mostrar una mayor cantidad de datos y en nuestro caso particular podríamos incluir esto en el servicio emergencia. En este caso, cuando se detectara una anomalía en las pulsaciones se activaría el servicio de emergencia.

Por otra parte creo que en cuanto a la interfaz gráfica se refiere, y como se ha puesto de manifiesto anteriormente, puede ser importante mejorar el diseño de la interfaz. Con esto se fomentaría la intención del usuario en seguir empleando la aplicación y en distribuirla.

En cuanto a la implementación creo que también sería interesante seguir optimizando el código realizado, añadiendo módulos y ampliando funcionalidades. En particular el tema de la batería,

seguir investigando en este campo con el fin de obtener la eficiencia de funcionamiento mejor posible.

En definitiva, existen multitud de campos de actuación con más o menos recorrido pero queda a disposición de lector retomar este proyecto y hacer grande. En mi opinión esta memoria contiene todas las claves para dar a entender al lector aquello que necesita para ampliar el proyecto.

## 5.4. Recopilación y conclusiones finales

Una vez el lector ha llegado a este punto, en el cual estamos cerca de terminar esta aventura, es el momento de echar la vista atrás y recopilar. Durante este proyecto hemos seguido una serie de etapas. Estas etapas pasan desde el diseño y la implementación hasta las pruebas de funcionamiento de nuestro SOFTSPM.

Tras presentar las tecnologías base empleadas en la creación de nuestra aplicación pasamos a explicar el procedimiento de diseño. Algunas directrices sobre el diseño y como implementar cada uno de los elementos clave de la aplicación. Recordemos estos elementos aplicación Web (Android), servidor (Django) e interfaz Web (HTML y CSS).

La implementación de estos elementos ha sido muy interesantes, en particular establecer conexiones funcionales entre los elementos anteriormente descritos. Una vez se ha realizado la implementación de estos elementos es sumamente satisfactorio ver como cada uno de los elementos se conecta funcionalmente, es decir, la aplicación funciona.

Todo este proceso, con la ayuda del lector, ha supuesto el proceso de diseño, desarrollo y despliegue de un servicio web de monitorización de actividades al aire libre para dispositivos Android. Este proceso ha durado prácticamente un año, con momento de estancamiento, momentos de iniciativa y momento simplemente de "picar código".

Ahora que estamos prácticamente al final de este proceso, es el momento de explicar la parte menos técnica y más sentimental de este proyecto. Este proyecto, aunque posiblemente tenga un par de exámenes delante ha supuesto el colofón de un proceso, un proyecto propio que he visto crecer desde 0. Un proyecto que ha tomado forma y se ha convertido en algo que, de forma coloquial, sirve para algo.

En cuanto a este texto, esta memoria, simplemente dar las gracias al lector por aguantar y leer este proyecto. Se ha intentado obviar implementaciones demasiado específicas de código, las cuales están disponibles en el CD/DVD adjunto. En esta memoria intento dar al lector un idea de los

procesos y decisiones tomadas y las razones de ellas, tanto para un público especializado como para alguien con unos conocimientos básicos. De todas formas estoy seguro que para muchos esto es demasiado simple y para otros un jeroglífico indescifrable.

Por otro lado, pedir disculpas si el vocabulario o la forma de expresarme no es del agrado del lector, pero esta hecho con la mejor intención, la intención de compartir el trabajo realizado durante este último año.

Seguro que me dejo cosas en el tintero pero esto llega a su fin, quedo a disposición del lector para cualquier duda que no pueda ser resuelta por esta memoria o para cualquier comentario.

En definitiva, gracias por este tiempo, espero que el lector haya conseguido algo como resultado de la lectura de este proyecto.



# APÉNDICES



## Manual de Usuario

Este apéndice expone las funcionalidades de la aplicación así como una serie de indicaciones sobre el uso de esta aplicación Web desde el punto de vista del usuario que la utilizara.

### A.1. Introducción y funcionalidad

SOFTSPM se trata de una aplicación que permite monitorizar nuestras actividades en el aire libre, de forma que contamos con una serie de servicios para monitorizar y proteger la realización de estas actividades. Para ello emplearemos un dispositivo móvil inteligente.

Por otra parte la información recabada durante la realización de la actividad estará disponible para su consulta e interacción mediante un servidor online accesible desde cualquier emplazamiento con conexión a Internet.

Sus funcionalidades principales son:

- Brújula Aumentada que nos indicara los PDIs (Puntos de interés) más próximos.
- Mapa con los PDIs.
- Monitorización de la ruta (distancia, altitud, tiempo, coordenadas, etc.).
- Petición de auxilio en caso de emergencia.
- Compartir y consultar la información de las rutas mediante el empleo de elementos sociales en Internet.



Figura A.1: Esquema de la aplicación

## A.2. Funcionamiento

Debido a la distribución y arquitectura de la aplicación veremos el funcionamiento de la misma desde el dispositivo Android y desde su Interfaz Web.

### A.2.1. Funcionamiento en dispositivos Android

La aplicación sobre Android cuenta con un intuitivo menú inicial, en el que fácilmente están disponibles las opciones que la aplicación permite. Dichas funciones son:

- Seguimiento: Es el núcleo de la aplicación, nos permite medir la actividad que se realiza y enviar dicha información al servidor, es aquí donde igualmente se desarrollan los procedimientos de seguridad durante la actividad al aire libre. Lo veremos en profundidad más adelante.
- Mapa: Proporciona un mapa de la zona en la que nos encontramos, en el cual se marcan destacadamente los puntos de interés definidos, estos puede ser puesto de socorro, refugios, etc.

- Brújula Aumentada: Nos proporciona mediante realidad Aumentada la dirección al punto de interés más cercano, para ello toma información de la posición actual, la posición de destino y elabora una trayectoria.
- Opciones: Opciones de aplicación, como por ejemplo las condiciones de actualización del servicio GPS.
- AcercaDe: Información sobre la aplicación.
- Salir: Salir de la aplicación.

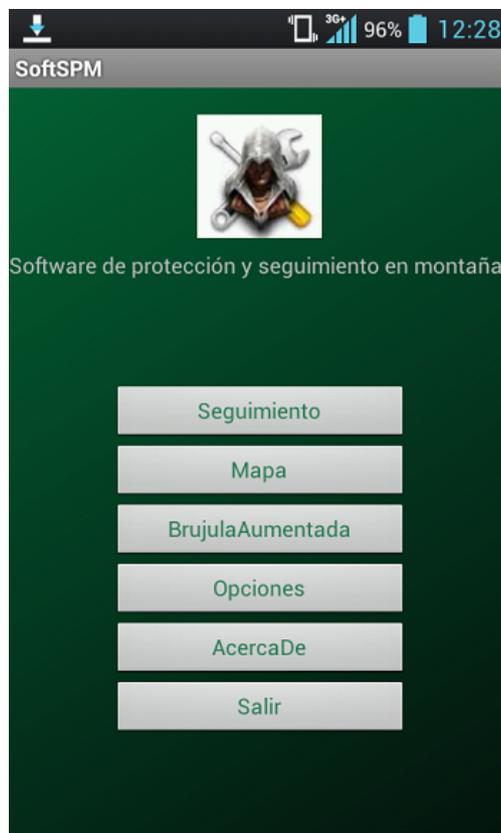


Figura A.2: *Menu de la aplicación*

Ahora nos centraremos en la parte Seguimiento, al entrar en esta parte, la aplicación nos lanzara una pantalla de autenticación de usuario mediante contraseña. Una vez autenticados entraremos en la interfaz con la cual podemos realizar la actividad que deseamos.

La interfaz de seguimiento cuenta con cinco posibles botones con los que interactuar en la aplicación. Dos de los botones se corresponden con la activación (Activar) y desactivación (Desactivar)

del proceso de seguimiento. Mientras estemos bajo el seguimiento, tendremos datos actualizados de nuestra posición, la distancia recorrida, la altitud, la velocidad media y el tiempo transcurrido.

A su vez tenemos el botón VerRuta, el cual mediante el servicio de Google Maps nos proporcionara la ruta desde el punto a inicial a la última posición conocida siempre que la cartografía lo permita.

El botón conexión nos permite enviar mensajes sociales al servidor con el fin de compartirlos con el resto de usuarios de la aplicación.

El botón SOS es el encargado de lanzar el servicio de emergencia, por una parte se envía un mensaje de auxilio al servidor con la posición de lugar de la emergencia y por otra se realiza una llamada a un número de ayuda, de la misma manera se inicia un tono acústico de auxilio desde el mismo terminal.

Otra de las características de la aplicación consiste en un sensor de movimiento, el cual está configurado para ante movimientos bruscos reaccionar de la misma manera que presionando el botón SOS.

### **A.2.2. Funcionamiento Interfaz Web**

La interfaz Web estara siempre accesible sobre una URL determinada. Esta interfaz permite a cada usuario tener su propio espacio personal. Este espacio incluye un perfil del usuario con datos básicos del mismo.

Cada usuario desde su cuenta es capaz de realizar diferentes acciones, alguna de ellas de caracter social. En una primera aproximación podriamos englobarlas en:

- Registro y Login de usuario.
  
- Registro de las rutas realizadas.
  
- Mensajería social entre usuarios.



Figura A.3: *Interfaz Web de la aplicación*

### A.3. Requisitos

Los requisitos para el empleo de la aplicación tanto de software como hardware son los siguientes:

- SmartPhone con sistema Android (optimizado para Android 2.1), GPS y sensores de movimiento.
- Conectividad a internet en el SmartPhone.
- Ordenador de sobremesa o portátil con conexión a Internet.



# APÉNDICE **B**

## Encuesta

Una encuesta ha de ser especialmente adaptada al producto que se quiera evaluar, en nuestro caso la encuesta está destinada a obtener la opinión de un usuario sobre la experiencia derivada del uso de la aplicación SOFTSPM.

A continuación podemos ver el modelo de encuesta que hemos utilizado.







# Bibliografía

- [1] Android official webpage. [www.android.com](http://www.android.com).
- [2] Artoolkit. <http://sourceforge.net/projects/artoolkit/>.
- [3] Django proyect documentation. [www.djangoproject.com](http://www.djangoproject.com).
- [4] Droidar. <https://github.com/bitstars/droidar>.
- [5] Python official webpage. [www.python.org](http://www.python.org).
- [6] sqlite. <http://www.sqlite.org/>.
- [7] World wide web consortium w3c. [www.w3.org](http://www.w3.org).
- [8] alwaysdata. Aalwaysdata, servicio de hosting. <https://www.alwaysdata.com/>, 2006-2013.

