



Universidad  
Carlos III de Madrid

Departamento de Informática

Trabajo Fin de Grado

SISTEMA MÓVIL DE INFORMACIÓN Y GUIADO

**Autor:** Adrián García Herías

**Tutor:** David Griol Barres

Colmenarejo, septiembre de 2013

Título: SISTEMA MÓVIL DE INFORMACIÓN Y GUIADO

Autor: Adrián García Herías

Director: David Griol Barres

## EL TRIBUNAL

Presidente: \_\_\_\_\_

Vocal: \_\_\_\_\_

Secretario: \_\_\_\_\_

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día \_\_ de \_\_\_\_\_  
de 20\_\_ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de  
Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

# Índice general

Índice general .....	3
Índice de figuras .....	5
Índice de tablas.....	6
Agradecimientos.....	7
Resumen.....	8
Abstract .....	9
<b>1 Introducción .....</b>	<b>10</b>
1.1 Antecedentes .....	10
1.2 Objetivos .....	11
1.3 Fases de desarrollo.....	11
1.4 Planificación temporal.....	14
1.5 Medios y documentación utilizados .....	16
1.5.1 <i>Dispositivos Hardware</i> .....	16
1.5.2 <i>Aplicaciones Software</i> .....	16
1.6 Estructura de la memoria.....	16
<b>2 Estado del Arte .....</b>	<b>18</b>
2.1 Principales sistemas operativos móviles.....	18
2.2 Android.....	21
2.2.1 <i>Historia</i> .....	22
2.2.2 <i>Arquitectura</i> .....	22
2.2.3 <i>Aplicaciones</i> .....	25
2.2.4 <i>Versiones</i> .....	25
2.3 Sistemas de localización.....	29
2.4 Sistemas de dialogo.....	30
<b>3 Aplicación .....</b>	<b>33</b>
3.1 Descripción general del sistema.....	33
3.1.1 <i>Objetivos</i> .....	33
3.1.2 <i>Requisitos</i> .....	33
3.1.3 <i>Funcionalidad</i> .....	36
3.1.4 <i>Herramientas del sistema</i> .....	36
3.1.5 <i>Recursos del sistema</i> .....	38
3.1.6 <i>Implementación de las operaciones generales</i> .....	39
3.2 Descripción detallada de los módulos del sistema .....	48

3.2.1	<i>Inicio</i> .....	48
3.2.2	<i>Perfil</i> .....	49
3.2.3	<i>Crear perfil</i> .....	51
3.2.4	<i>Modificar o eliminar perfil</i> .....	61
3.2.5	<i>Reglas</i> .....	65
3.2.6	<i>Crear regla</i> .....	66
3.2.7	<i>Modificar o eliminar regla</i> .....	68
3.2.8	<i>Localizaciones</i> .....	69
3.2.9	<i>Crear localización</i> .....	70
3.2.10	<i>Modificar o eliminar localización</i> .....	72
<b>4</b>	<b>Conclusiones y Trabajos Futuros</b> .....	<b>74</b>
4.1	Conclusiones.....	74
4.2	Trabajos Futuros.....	75
4.2.1	<i>NFC</i> .....	75
4.2.2	<i>Sistemas empotrados</i> .....	75
	<b>Presupuesto</b> .....	<b>76</b>
	<b>Glosario</b> .....	<b>79</b>
	<b>Bibliografía</b> .....	<b>80</b>

# Índice de figuras

Ilustración 1 - Diagrama WBS de las tareas definidas en el Trabajo Fin de Grado .....	13
Ilustración 2 - Diagrama de Gantt de la planificación temporal del Trabajo Fin de Grado.....	15
Ilustración 3 -Cuota de mercado de los diferentes sistemas operativos móviles [IAB].....	21
Ilustración 4 - Arquitectura de Android .....	23
Ilustración 5 - Arquitectura modular de un sistema de diálogo .....	31
Ilustración 6 - Sistema de ficheros de la aplicación .....	37
Ilustración 7 - Interfaz correspondiente al módulo perfil.....	49
Ilustración 8 - Interfaz correspondiente al módulo crear perfil.....	51
Ilustración 9 - Interfaz correspondiente al módulo crear perfil.....	52
Ilustración 10 - Pop up correspondiente al nombre del perfil.....	53
Ilustración 11 - Pop up correspondiente al volumen del perfil.....	54
Ilustración 12 - Pop up correspondiente a la vibración del perfil.....	55
Ilustración 13 - Pop up correspondiente al brillo de la pantalla del dispositivo.....	55
Ilustración 14 - Pop up correspondiente al tiempo de espera de la pantalla del dispositivo.....	56
Ilustración 15 - Pop up correspondiente al rotado de la pantalla del dispositivo.....	57
Ilustración 16 - Pop up correspondiente al Wifi del perfil.....	58
Ilustración 17 - Pop up correspondiente al Datos del perfil.....	58
Ilustración 18 - Pop up correspondiente al Bluetooth del perfil.....	59
Ilustración 19 - Interfaz correspondiente al módulo modificar/eliminar perfil.....	62
Ilustración 20 - Interfaz correspondiente al módulo regla.....	65
Ilustración 21 - Interfaz correspondiente al módulo localización.....	69
Ilustración 22 - Interfaz correspondiente al módulo Crear localización.....	71

# Índice de tablas

Tabla 1 - Formato de requisitos .....	34
Tabla 2 - Requisito RF01 .....	34
Tabla 3 - Requisito RF02 .....	34
Tabla 4 - Requisito RF03 .....	34
Tabla 5 - Requisito RF04 .....	34
Tabla 6 - Requisito RNF01 .....	35
Tabla 7 - Requisito RNF02 .....	35
Tabla 8 - Requisito RNF03 .....	35
Tabla 9 - Requisito RR01.....	35
Tabla 10 - Requisito RR02.....	35
Tabla 11 - Detalle de costes de recursos humanos.....	77
Tabla 12 - Detalle de coste total .....	78

# Agradecimientos

En primer lugar, quisiera agradecer a mi tutor, David Griol, por darme la posibilidad de realizar un proyecto como este. Incluyendo algunos de los campos en los que está especializado, como es el caso de los sistemas de dialogo. Además, de haberme guiado y proporcionado libertad a la hora de desarrollar mí idea sobre el Trabajo de Fin de grado.

Quisiera dar una mención especial a mis padres y hermano, quienes me han apoyado en los buenos y malos momentos, dándome un ánimo y apoyo incondicional desde que inicie tanto mis estudios de secundaria como los universitarios.

A su vez, tengo presente al resto de familiares, abuelos, tíos, primos. Puesto que siempre me he sentido respaldado por ellos y convencidos de que lo lograría. Dándome siempre buenos consejos y enseñándome buenos valores a emplear en el día a día.

Quiero destacar a los familiares que no podrán leer este agradecimiento, pero que estarán presentes en mi recuerdo por siempre.

También quiero acordarme en este momento de todo mi grupo de amigos, que siempre me han ayudado en los malos momentos y me han animado a finalizar mis estudios universitarios. Por ello, no pueden faltar en este apartado Alejandro, Alexis, Álvaro, Javier, José Luis, etc.

Por último y no menos importantes, quisiera agradecer a mis compañeros y amigos de la facultad, con los que he compartido muchísimas horas de estudio, pero también de ocio. Entre todos ellos, querría destacar a Aitor, Jesús y Mónica, con los que a día de hoy mantengo una grandísima amistad.

En conclusión, muchísimas gracias a todos los que habéis confiado en mí, puesto que sin vuestro apoyo no habría alcanzado este objetivo.

# Resumen

Actualmente con la gran cantidad de dispositivos móviles inteligentes (Smartphone), los usuarios demandan nuevos servicios que automaticen o faciliten sus tareas cotidianas. Por ello, el objetivo de este Trabajo Fin de Grado es el desarrollo de una aplicación que automatice una tarea cotidiana del usuario aplicando para ello la inteligencia ambiental, mediante un dispositivo móvil Android.

A través de la aplicación, los usuarios podrán automatizar sus preferencias mediante el posicionamiento del mismo y/o periodo de tiempo. Además, la aplicación permite la interacción tradicional mediante el teclado virtual del dispositivo y también, el reconocimiento de voz.

A su vez, entre las funcionalidades secundarias de la aplicación se encuentra la creación de perfiles, reglas y localizaciones con las preferencias del usuario y su administración.

Para llevar a cabo dicha aplicación, es necesario apoyarse en otras tecnologías adicionales, empezando por el uso de diferentes lenguajes de programación (Java, SQL y XML), implementando las propias APIs de Android (Google Voice Recognizer, Google Places y Speech To Text), proporcionando a la aplicación mayor flexibilidad y eficiencia. Por último, se emplea como motor de base de datos, el integrado en el sistema operativo Android (SQLite), proporcionando de este modo la función de almacenamiento a la aplicación.

Como complemento a esta aplicación, se lleva a cabo un estudio detallado de los principales sistemas operativos móviles, profundizando especialmente en el sistema operativo elegido para el desarrollo de la aplicación (Android). Además, se realiza un estudio de los sistemas de dialogo, detallando sus arquitectura general y describiendo el tipo de sistema implementado en la aplicación. Estos estudios complementarios, forman parte del análisis previo al desarrollo de la aplicación, permitiendo seleccionar tanto el sistema operativo más atractivo para el desarrollo de la misma, así como eligiendo el tipo de sistema de dialogo más adecuado para la aplicación.

**Palabras clave:** Dispositivos móviles, Sistemas operativos móviles, Android, sistemas de diálogo, sistemas de localización, aplicación, bases de datos.

# Abstract

Nowadays and with the huge amount of intelligent mobile devices (smartphones), users ask for new services which can make easier their everyday tasks. The objective of this research paper is the development of an application that makes automatic some tasks of the users, applying () in an Android mobile device.

With the application, users will be able to automate their preferences by the positioning of the device and/or by the same period of time. Furthermore, the application allows the interaction between the virtual keyboard of the device and also between the voice recognition function.

The creation of new profiles, rules and locations with the preferences of the user and his administration are considered as secondary functionalities.

In order to create this application, it is necessary to lie on other additional technologies, such as the use of different programming languages (Java, SQL and XML), implementing Android's own APIs (Google Voice Recognizer, Google Places and Speech To Text), giving flexibility and efficiency to the application. Finally, as database engine, it is used the same one which is integrated in Android operating system (SQLite).

As a complement for this application, it has been done a detailed study about the principal mobile operating systems, with particular regard to the operating system chosen for the development of the application (Android). In addition, it is carried out a study of dialogue systems, detailing its overall architecture and describing the type of system implemented in the application. These further investigations are part of the pre-analysis application development, allowing the selection of the operating system that best fits with the development of the application, as well as choosing the type of system of dialogue more suitable for the application.

**Keywords:** mobile devices, mobile operating systems, Android, dialogue systems, location systems, application, database.

# Capítulo 1

---

## 1 Introducción

En este primer apartado se exponen los antecedentes del Trabajo de Fin de Grado, se establecen los objetivos del mismo y se detallan los aspectos generales relativos a las distintas fases del proyecto (desarrollo, planificación temporal y medios empleados para su ejecución). Por último, se describe la estructura empleada en este documento.

### 1.1 Antecedentes

En la actualidad, los dispositivos móviles constituyen una herramienta para la vida cotidiana de las personas. Son proveedores de información, comunicación y otras funcionalidades de manera inmediata y portable. Mientras que los usuarios son demandantes de servicios prácticos y sencillos. Todo ello ha favorecido al gran crecimiento de este tipo de dispositivos frente a otros muchos.

Según los resultados del informe publicado por el Observatorio nacional de las telecomunicaciones y de la SI sobre “Las TIC en los hogares españoles – XLII Oleada (Octubre – Diciembre 2013) [ONSI]”, el 95,6% de los hogares españoles dispone de al menos un teléfono móvil.

Según otro estudio publicado por IAB Spain “Radiografía del mercado móvil en España – Mobile World Centre (Septiembre 2013) [IAB]”, el 80% de los dispositivos móviles de los hogares españoles son “Smartphone”.

A su vez este mismo estudio, arroja datos acerca del comportamiento de los usuarios con sus dispositivos. El 84% del uso está orientado a conexión a internet, de los cuales el 71% es a través de aplicaciones que satisfacen la demanda del usuario.

Estos datos ponen de manifiesto la necesidad de usuario de acceder a la información orientada a conexión. Originalmente los buscadores y navegadores nos ofrecían este tipo de servicios, con la aparición de nuevos sensores en el hardware de los smartphones, han surgido nuevos sistemas de acceso a la información como los sistemas de localización. Esta información en base al posicionamiento, ofrece un cambio en cuanto a la comunicación, ocio y aparición de nuevos servicios (domótica, inteligencia ambiental, etc.).

A través de la aplicación a desarrollar en este Trabajo Fin de Grado, se llevara a cabo la automatización de tareas previamente descritas por el usuario mediante el sistema de geolocalización (inteligencia ambiental). Además, se proporcionará accesibilidad por voz para realizar consultas.

De otro modo, el usuario podrá realizar consultas no solo mediante voz, sino también a través del teclado del dispositivo, pudiendo combinar ambas opciones, facilitando una mayor accesibilidad de la aplicación.

## 1.2 Objetivos

Como se avanzó en el apartado anterior, el objetivo principal de este Trabajo Fin de Grado es desarrollar una aplicación en el sistema operativo Android con la que los usuarios puedan automatizar tareas respecto a su geolocalización.

Aprovechando las posibilidades que ofrece el sistema operativo Android, se ha implementado una aplicación útil en el uso cotidiano y que será utilizada por los usuarios para gestionar el uso de sus perfiles diariamente.

A su vez trataremos varios objetivos secundarios:

- Llevar a cabo un análisis general de los principales sistemas operativos (Android, Apple IOS y Windows Phone). Se ha tenido en cuenta su evolución en el mercado, arquitectura, lenguaje de programación, sistema y coste de publicación de aplicaciones.
- Realizar un estudio detallado del sistema operativo empleado (Android). Analizando las distintas versiones del mismo (mejoras incorporadas y cuotas de mercado).
- Llevar a cabo un estudio detallado de los sistemas de localización (GPS, Wifi y Geolocalización), centrándonos en sus características, ventajas y desventajas.

## 1.3 Fases de desarrollo

La ejecución de este Trabajo Fin de Grado ha sido realizada en tres fases de desarrollo, siguiendo los conocimientos adquiridos en la asignatura de Dirección de proyectos de Software.

### **Fase 1: Planificación.**

- Análisis general de los principales sistemas operativos móviles (Android, Apple IOS y Windows Phone): Determinación de sistema operativo en el cual se desarrollara la aplicación.
- 
- Estudio detallado del sistema operativo Android: Determinación de la versión mínima en la cual se desarrollará la aplicación.
- 
- Estudio detallado de los sistemas de localización: Aproximación a los principales sistemas de localización (GPS, Wifi y Geolocalización).
- 
- Aprendizaje de las tecnologías necesarias: Android, API'S Google, Java, SQLite y XML.
- 
- Análisis de requisitos de la aplicación: Determinación de los requisitos que cumplirá la aplicación.

### **Fase 2: Desarrollo.**

- Análisis y diseño conceptual: división de las funcionalidades en los diferentes módulos y submódulos del sistema.
- Análisis y diseño gráfico: división de las interfaces para los diferentes módulos del sistema.
- Implementación del sistema: Desarrollo de todos los módulos y submódulos, así como las correspondientes interfaces facilitando la interacción entre ambos.
- Pruebas unitarias: Análisis, implementación y ejecución de las pruebas funcionales de cada módulo y submódulo del sistema.
- Pruebas de integración y de sistema: proceso de pruebas del sistema completo hasta alcanzar una versión estable del sistema.
- Evaluación de la aplicación: Aprendizaje de sistemas de evaluación para interfaces de usuario, realización de la encuesta de evaluación, recogida y análisis de resultados.

### **Fase 3: Documentación.**

- Memoria del Trabajo Fin de Grado: Redacción del presente documento de memoria del Trabajo Fin de Grado.
- Preparación de la presentación.

Para una sencilla comprensión de las fases de desarrollo y sus tareas, se construye un diagrama WBS (Work Breakdown Structure). Se trata de una estructura en árbol en la que se ordenan las tareas y subtareas con arreglo a un determinado criterio. La principal utilidad de este tipo de diagrama es proporcionar un método sistemático para confeccionar la lista de tareas.

El diagrama WBS utilizado se muestra en la Ilustración 1. Como puede observarse, para la ejecución del Trabajo Fin de Grado se han definido tres fases, que constituyen el primer nivel de la jerarquía a partir del cual se han desarrollado las 13 tareas que componen este proyecto (tareas con índice del 1 al 13). Estas 13 tareas conforman el último nivel jerárquico, agrupándose en columnas dependiendo a la fase que pertenezcan.

Solo el nivel más bajo (nivel 2) representa tareas reales, mientras los niveles superiores (nivel 1) son consideradas fases que resumen el proyecto.

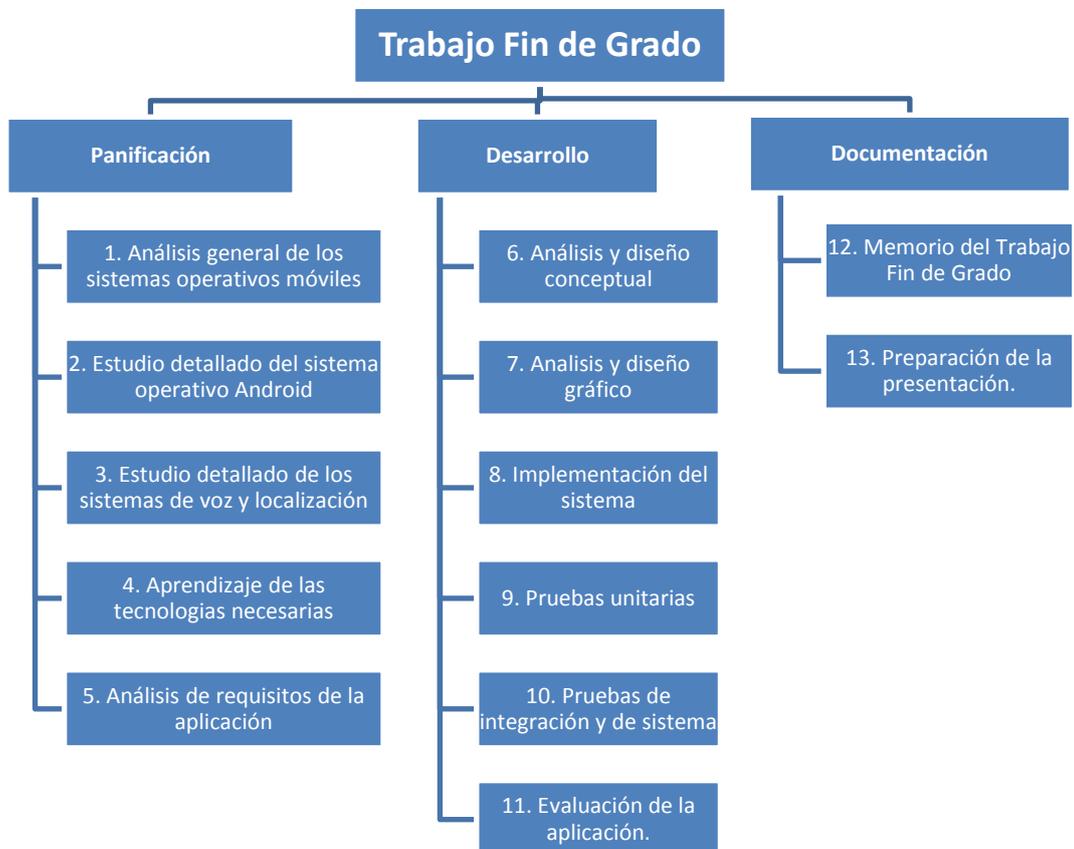


Ilustración 1 - Diagrama WBS de las tareas definidas en el Trabajo Fin de Grado

## 1.4 Planificación temporal

Una vez establecidas las fases y tareas en las que se ha dividido el proyecto (Apartado 1.3 - Fases de desarrollo), se ha realizado la planificación temporal de las fases mediante un diagrama de Gantt. Posibilitando de esta manera el seguimiento detallado de la evolución de cada una de las tareas.

Para la realización del diagrama de Gantt, además de incluir las 13 tareas divididas en sus respectivas fases (Ilustración 1 Diagrama WBS), se han añadido un hito de inicio de Trabajo Fin de Grado y un hito de fin de Trabajo Fin de Grado. Estos hitos son tareas de duración nula, son usados para marcar puntos de control de avance en el Trabajo Fin de Grado.

Seguidamente se introducen las duraciones de cada tarea, se ha considerado una jornada de trabajo consta de un máximo de 8 horas por cada día de la semana. Se establecen relaciones de precedencia entre cada tarea, fijando la restricción de no empezar una tarea si la precedente a la misma no ha sido finalizada.

Una vez aplicado este procedimiento, obtenemos el diagrama de Gantt resultante que puede observarse en la Ilustración 2. Con este diagramado se muestra una visión global de la planificación temporal del Trabajo Fin de Grado y de sus distintas fases y tareas.

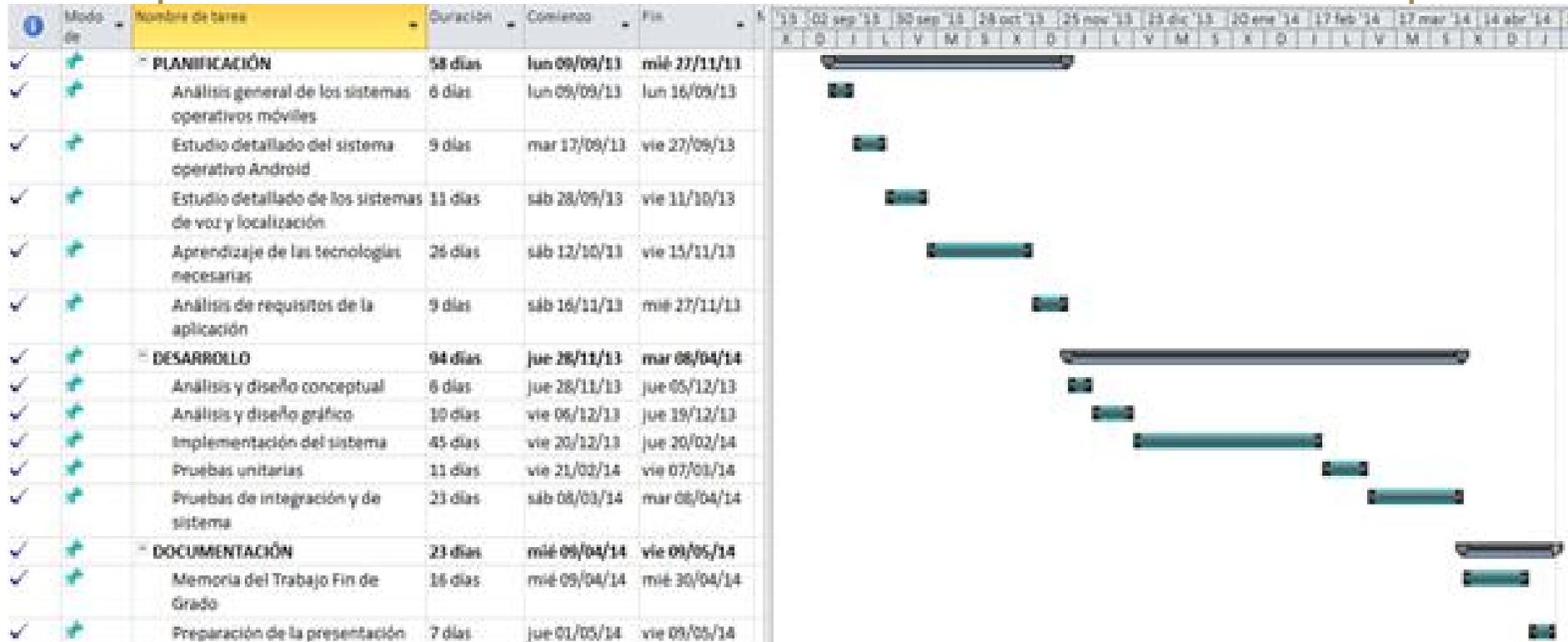


Ilustración 2 - Diagrama de Gantt de la planificación temporal del Trabajo Fin de Grado

## 1.5 Medios y documentación utilizados

A continuación se enumeran los medios empleados en la realización del Trabajo Fin de Grado:

### 1.5.1 Dispositivos Hardware

- Ordenador portátil.
- Ordenador sobremesa.
- Monitor.
- Periféricos habituales (teclado, ratón).
- Smartphone Android (2.3 o superior).
- Cable USB.

### 1.5.2 Aplicaciones Software

- Entorno de desarrollo (Eclipse + plugins Android).
- Editor de texto (Notepad ++).
- Paquete Microsoft Office 2010.
- Adobe Reader.
- Skype.

En el apartado Presupuesto, se adjunta información referente al coste de los anteriores medios citados.

En cuanto a la documentación utilizada, se han empleado numerosos manuales y libros acerca del lenguaje de programación, el sistema operativo Android y sus respectivas API'S, además de diversos artículos específicos del sistema operativo Android y los sistemas de localización.

## 1.6 Estructura de la memoria

Con la intención de facilitar el seguimiento de la memoria, se incluye a continuación una breve descripción del contenido de cada uno de los capítulos que integran este documento:

**Capítulo 1: Introducción.** Este primer capítulo establece el propósito y los objetivos de todo el contenido del Trabajo Fin de Grado. Además, se incluyen las fases de desarrollo, la planificación temporal, los medios empleados y la estructura de la memoria.

**Capítulo 2: Estado del arte.** En este capítulo se realiza un estudio general de los principales sistemas operativos móviles (Android, Apple IOS y Windows Phone). Mediante a los datos obtenidos, se determina Android como sistema en el cual se desarrollará la aplicación. Se realiza un estudio detallado del sistema operativo seleccionado (Android), determinando

la versión a utilizar del mismo para el desarrollo de la aplicación. Finalmente, se realiza un estudio detallado de los sistemas de localización.

**Capítulo 3: Aplicación.** Dicho capítulo está dividido en tres apartados:

- **Descripción general del sistema:** En este apartado se proporciona una visión global de la aplicación desarrollada, analizando las tecnologías empleadas y detallando las operaciones más comunes tratadas a lo largo de su implementación.
- **Descripción detallada de los módulos del sistema:** En esta sección se describe detalladamente cada uno de los módulos de los que consta la aplicación. Cada uno de los módulos estará asociado a una interfaz o actividad (Crear perfil, Modificar/Borrar perfil, Añadir regla). Finalmente, se realizarán pruebas unitarias para corroborar el correcto funcionamiento de los módulos.

**Capítulo 4: Conclusiones y trabajos futuros.** En este capítulo se exponen las principales ideas, pensamientos, cuestiones y conclusiones fruto de la realización del Trabajo Fin de Grado, así como las posibles líneas de investigación que a partir de este proyecto se pudieran generar.

**Presupuesto.** Este apartado contiene una relación de los costes de diseño y desarrollo del Trabajo Fin de Grado, detallando el coste humano y material necesario para la realización del mismo.

**Glosario.** En este apartado se recogen los principales términos y conceptos técnicos utilizados a lo largo del documento, con el objetivo de facilitar su comprensión al lector.

**Bibliografía.** En este apartado se recopilan las citas bibliográficas que se han consultado para la realización del presente documento y aplicación.

# Capítulo 2

---

## 2 Estado del Arte

### 2.1 Principales sistemas operativos móviles

En la actualidad, los sistemas operativos móviles ofrecen una serie de ventajas frente a los sistemas operativos tradicionales. A su vez, dentro de este grupo existe una gran diversidad de opciones para cada dispositivo. Los principales sistemas operativos móviles del mercado son Android de Google, IOS de Apple y Windows Phone de Microsoft.

El componente más destacable en estos sistemas operativos es su propio núcleo (Kernel). Estos kernel comparten origen con los sistemas operativos de los ordenadores, siendo modificados para su uso en los dispositivos móviles. Android usa un kernel basado en Linux, con la particularidad de utilizar un motor Java (Dalvik). Por otro lado, los dispositivos Apple (IOS) utilizan un kernel basado en OS X, siendo este a su vez una variante de Unix.

Seguidamente, se detallarán las características más destacables de los principales sistemas operativos móviles:

**Android OS (Google Inc.):** Este sistema operativo fue desarrollado por la empresa “Android Inc” para el uso en dispositivos móviles de pantalla táctil. Su presentación oficial data del año 2007, pero no está hasta octubre de 2008 cuando se vende el primer móvil (HTC Dream) con este sistema operativo.

Este sistema operativo móvil, es el preferido por la gran mayoría de los usuarios, puesto que se trata de un sistema operativo de código libre y abierto (basado en Linux).

Este hecho favorece la diversidad de fabricantes que usan este sistema operativo, satisfaciendo las necesidades funcionales y económicas del usuario. A su vez, esto representa un problema a la hora de distribuir las actualizaciones del sistema, ya que cada fabricante es el encargado de distribuirlas.

Una característica de estas actualizaciones es su compatibilidad con sus versiones predecesoras, puesto que cada nueva versión lanzada es compatible al 100%, por tratarse de la inclusión de nuevas funcionalidades.

Caso curioso es el nombre utilizado para estas revisiones del sistema operativo Android por tratarse de nombres de dulces y a partir de la tercera actualización seguir un orden alfabético (Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean y KitKat).

En cuando a los desarrolladores, pueden crear aplicaciones sin ningún coste, puesto que el entorno de desarrollo es gratuito (SDK y simulador). No obstante para publicar la aplicación en la tienda oficial es necesario pagar 25\$ anuales. A diferencia de otros sistemas operativos es suficiente con auto firmar digitalmente la aplicación a publicar.

**BlackBerry OS:** Este sistema operativo fue desarrollado por la empresa “Research In Motion” para su uso exclusivo en los dispositivos BlackBerry. Su origen se remonta a la aparición de las primeras PDA’s en 1999.

Este fabricante, está especializado en grandes corporaciones y ofrece a sus usuarios una gran capacidad de sincronización con Microsoft Exchange, Lotus Domino, Novell GroupWise y email.

No fue hasta 2008 con la popularización del acceso a internet desde el móvil, cuando BlackBerry se coloca entre los principales sistemas móviles. Llegando a los usuarios de a pie, con terminales de bajo coste con la versión BlackBerry OS 5, este boom es mayor en Latinoamérica.

En cuando a los desarrolladores, pueden crear aplicaciones sin ningún coste, puesto que el entorno de desarrollo es gratuito (SDK y simulador). No obstante para publicar la aplicación en la tienda oficial, es necesario firmar digitalmente la aplicación y por ello pagar 200\$ por cada 10 aplicaciones.

**iPhone OS / iOS (Apple):** Este sistema operativo fue desarrollado por la empresa “Cupernico” para el dispositivo iPhone (iPhone OS). Actualmente el sistema operativo extendido en los dispositivos móviles de marca Apple es el iOS.

Se trata de una versión reducida del sistema operativo OS X para ordenador, desarrollado íntegramente por Apple Inc., y en el cual no está permitida su instalación en dispositivos de terceros.

Este sistema operativo ha sufrido varias evoluciones en cuento a su primera versión iPhone OS 1.0 y la última iOS 7 (próximamente presentarán oficialmente el IOS 8).

Estas versiones son distribuidas a través del servicio iTunes para la actualización de todos los dispositivos con sistema operativo IOS. Este servicio surge como necesidad de sincronizar todos los dispositivos móviles (iPod, iPhone e iPad), ofreciendo al usuario final parches de seguridad, nuevas mejoras y otros muchos servicios.

Su arquitectura básica consta de 4 capas [APPLE]:

- Capa principal del OS: Esta capa contiene las características de bajo nivel, siendo utilizadas por aplicaciones u otros marcos como puede ser la comunicación de un accesorio hardware externo.
- Capa de servicios principales: Esta capa contiene los servicios fundamentales del sistema para sus aplicaciones. A su vez, esta capa da soporte a características como la ubicación, iCloud, medios de comunicación social, y creación de redes.

- Capa de Medios principales: Esta capa contiene las interfaces gráficas, audio y video que se emplean para llevar a cabo las experiencias multimedia en sus aplicaciones.
- Cocoa Touch: Esta capa contiene los marcos fundamentales para la construcción de aplicaciones de IOS. Estos marcos definen el aspecto grafico de la aplicación. Además de proporcionar la infraestructura de aplicación básica y el apoyo para tecnologías como multitarea, entrada táctil, notificaciones y servicios de alto nivel.

Actualmente los desarrolladores de aplicaciones IOS, deben pagar 99\$ para que Apple les facilite el certificado Root CA (Root Certificate Authority), con el que será firmada la aplicación para la publicación de la misma a través del App Store. Antes de que se produzca la publicación de las aplicaciones o revisiones de las mismas en el App Store (repositorio de aplicaciones para IOS, gratuitas y de pago) ha de pasar por dos filtros de seguridad. El primero automatizado por un sistema y el segundo por una persona del departamento de Apple. Este hecho no favorece la agilidad del desarrollador, pero a priori favorece la calidad de la aplicación y seguridad de las mismas.

Otro de los problemas a los que se enfrenta el desarrollador de aplicaciones IOS, es el hermetismo del propio sistema operativo, no favoreciendo la customización tanto del software, como del hardware. Además de la corta gama de dispositivos móviles, puesto que solo Apple emplea este sistema operativo.

**Symbian OS (Nokia):** Este sistema operativo fue producto de la alianza de varios fabricantes de telefonía móvil. El objetivo de Symbian fue crear un sistema operativo para terminales móviles que pudiera competir con Palm y Windows Mobile.

Symbian ofrece una gran integración con comunicaciones y la administración de información personal. Combina el “middleware” de las comunicaciones inalámbricas con la integración al buzón de entrada del correo electrónico.

**Windows Mobile (Windows Phone):** Este sistema operativo móvil fue desarrollado por Microsoft para el uso en Smartphone y dispositivos móviles con o sin pantalla táctil. En 2010 tuvo lugar el lanzamiento de su nueva plataforma para dispositivos móviles Windows Phone 7, mientras que su versión actual (Windows Phone 8) fue presentada el 29 de septiembre de 2012.

Este sistema operativo móvil desarrollado por Microsoft, está basado en el kernel Windows CE 5.2, cuenta con un conjunto de aplicaciones básicas utilizando las APIs de Microsoft Windows. En cuanto a su diseño gráfico, es muy similar a las versiones de escritorio de Windows.

Actualmente los desarrolladores, pueden crear aplicaciones a través de App Hub, previo a un registro en el mismo, este servicio conlleva un coste de 99\$. Esto les permite crear aplicaciones tanto para sistemas móviles como para la videoconsola XBOX. A su vez, se le permite al desarrollador publicar un número ilimitado de aplicaciones de pago en el Marketplace. Mientras que tiene un máximo de 5 aplicaciones gratuitas, una vez superada esta cifra, para publicar una nueva aplicación gratuita el desarrollador tendrá que pagar 19,99\$.

Antes de publicar la nueva aplicación en Marketplace, es necesario pasar un proceso de aprobación, donde se evalúan factores de contenido y performance.

Finalizando este apartado, se puede observar que todos los sistemas operativos móviles, son semejantes. Aunque las características específicas de cada uno, los hacen más o menos atractivos a la hora de empezar a desarrollar una aplicación.

Como se puede apreciar en Figura 3 [IAB], el sistema operativo más utilizado en España es Android, seguido de lejos por iOS. Además es el único que ha crecido entre el 2012 y 2013, en cuanto a cuota de mercado se refiere, posicionándose en un 78%.



 Sistema operativo	2013		2012
Android	78%	▲	63%
iOS	11%		14%
BlackBerry	5%		8%
Windows	3%		3%
Symbian	2%		7%

Ilustración 3 -Cuota de mercado de los diferentes sistemas operativos móviles [IAB].

## 2.2 Android

Es un sistema operativo basado en Linux, que junto con aplicaciones middleware está destinado a ser utilizado en dispositivos móviles. Debido a su éxito en el mercado, este sistema operativo ha sido integrado en tablets, notebooks y diversos dispositivos más, a parte de los ya mencionados Smartphone.

Sistema operativo desarrollado por la empresa “Android Inc”, esta empresa fue comprada por Google en 2005, convirtiéndose así en el principal producto de “Open Handset Alliance” (conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio). Siendo en la actualidad el sistema operativo móvil más utilizado en el mundo.

En la actualidad, la gran comunidad de desarrolladores que posee Android creando aplicaciones favorece la extensión de modo asombroso la funcionalidad y el potencial de los dispositivos portadores de este sistema operativo. En estos momentos, se alcanzan los

1.3 millones de aplicaciones y teniendo en cuenta que dos tercios de las mismas se tratan de aplicaciones gratuitas, es lógico que la gran mayoría de los usuarios prefieran este sistema operativo frente a otros.

El posible talón de Aquiles de este sistema operativo, podría ser la seguridad. Puesto no siguen un control exhaustivo de la seguridad de las aplicaciones publicadas en Google Play. Aunque la mayoría del “malware”, proviene de sitios de terceros y no del repositorio oficial.

### 2.2.1 Historia

El propio origen del nombre Android hace alusión a la película “Blade Runner” (adaptación de la novela “¿Sueñan los androides con ovejas electrónicas?” de Philip K. Dick). Tanto es así, que el propio logotipo de Android (el robot “Andy”), es la forma de llamar en “Blade Runner” a los Androides.

Este sistema operativo fue desarrollado por la empresa “Android Inc” para el uso en dispositivos móviles de pantalla táctil. Siendo esta empresa comprada por Google en 2005. Levantando las sospechas y rumores sobre el desembarco de Google en el mercado de los Smartphone.

En ese momento, Google desarrolló una plataforma para dispositivos móviles basada en el kernel de Linux. Éste, fue promocionado a fabricantes de dispositivos y operadores con la promesa de proveer un sistema flexible y actualizable. En aquel momento, se verificó la cooperación entre fabricantes de hardware, software y las principales operadoras.

Finalmente, el 5 de Noviembre de 2007 se oficializa Android como sistema operativo móvil, junto con la creación del consorcio de fabricantes “Open Handset Alliance”, el cual consta inicialmente de 78 empresas de hardware, software y telecomunicaciones dedicadas al desarrollo de estándares abiertos para dispositivos móviles.

Google, lleva a cabo la liberación de gran parte del código de Android bajo la licencia Apache, la cual es libre y de código abierto, dando de este modo plena confianza en los desarrolladores.

### 2.2.2 Arquitectura

Android es una plataforma para dispositivos móviles que contiene una pila de software donde se distingue un sistema operativo basado en Linux, middleware y aplicaciones básicas para el usuario.

A continuación se dará una visión más detallada por las capas que componen la arquitectura de Android. Como se observa en el Ilustración 4, la arquitectura está basada

en capas. Cada de ellas utiliza los servicios ofrecidos por las anteriores, estando todas ellas basadas en software libre.



Ilustración 4 - Arquitectura de Android

### **Kernel**

El kernel del sistema operativo Android está basado en el Kernel de Linux (versión 2.6). Esta capa de la arquitectura actúa como abstracción entre el hardware y el resto de la pila de software. A su vez, para cada elemento de hardware del dispositivo se dispondrá de un controlador, que permitirá su uso desde el resto de pilas de software.

Esta capa proporciona servicios base del sistema como la seguridad, la gestión de memoria o procesos y la pila de protocolos. El desarrollador no tendrá acceso directo a los servicios de esta capa, sino que empleará las librerías disponibles en capas superiores.

La plataforma hardware principal que soporta Android es la arquitectura ARM, pero también da soporte a x86 y Google TV, utilizando una versión especial de Android x86.

### **Librerías nativas**

En esta capa que se sitúa sobre el Kernel e integra el runtime del sistema, se compone por las bibliotecas nativas de Android, están en lenguaje C y C++ y compiladas en el código nativo del procesador. Estas librerías o bibliotecas proporcionan funcionalidad a las aplicaciones para tareas repetitivas, evitando así desarrollarlas cada vez que sean necesarias y garantizando la optimización de las mismas. Esas funcionalidades se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android.

### **Runtime de Android**

Está basado en el concepto de máquina virtual, utilizando Java para ello. Android corre su propio proceso, con su propia instancia de Dalvik (máquina virtual usada). Dalvik, ha sido desarrollada de manera que un dispositivo ha de ser capaz de ejecutar varias máquinas virtuales a la vez y de forma eficiente. De este modo, Dalvik ejecuta archivos en formato

Dalvik Executable (.dex), el cual está optimizado para minimizar el consumo de memoria de los dispositivos.

Alguna de las características más destacables en cuanto a optimización de recursos por parte de la máquina virtual Dalvik, son:

- Máquina virtual basada en registros.
- Cada aplicación se ejecuta en su propio proceso Linux, con su propia instancia de la máquina virtual.
- Se delega en la capa Kernel las funciones de procesos o la gestión de memoria entre otras.

A su vez, se incluyen un set de bibliotecas que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java.

### **Marco de trabajo de aplicaciones**

Esta capa está diseñada para simplificar la reutilización de componentes, pudiendo las aplicaciones publicar sus capacidades, y así otras aplicaciones poder hacer uso de esas capacidades. A su vez, los desarrolladores tienen acceso completo a la API del framework usada por las aplicaciones base.

Entre los servicios más importantes de esta capa se encuentran los siguientes:

**View:** Este servicio es el encargado de representar los componentes interactivos de la interfaz de usuario en la pantalla del dispositivo.

**Resource Manager:** Este servicio gestiona los elementos que conforman la aplicación y que están fuera del código de la misma.

**Activity Manager:** Este servicio administra la pila de actividades de la aplicación, desde el ciclo de vida, hasta la interacción entre ellas mismas.

**Notification Manager:** Este servicio es el encargado de notificar o alertar al usuario de un nuevo evento, mediante la barra de estado o incluso activando alguno de los sensores del dispositivo (vibración, LED's, etc.).

**Content Providers:** Este servicio proporciona la encapsulación de los datos que se comparten entre aplicaciones, controlando el acceso a la información de las mismas.

### **Aplicaciones**

Esta capa está conformada por el conjunto de aplicaciones base instaladas en Android. Estas aplicaciones han de ser ejecutadas en la máquina virtual Dalvik, garantizando de este modo la seguridad del sistema.

Entre las aplicaciones base más destacables, se incluyen las siguientes:

- Contactos

- Teléfono
- Navegador web
- Otros

### 2.2.3 Aplicaciones

Las aplicaciones se desarrollan como norma general en el lenguaje “Java”, con el Software “Development Kit de Android” (SDK). También se encuentran disponibles aplicaciones desarrolladas mediante lenguajes de programación como C o C++. Es el ejemplo de Google App Inventor, y otras muchas.

Para el desarrollo de aplicaciones Android, basta con tener un conocimiento aceptable de “Java”, además de tener clara la arquitectura de Android, teniendo especial atención con el uso de actividades y vistas. A su vez, es necesario tener un entorno de desarrollo adecuado. Entre los muchos que existen, el más extendido y popularizado es “Eclipse”, junto con el ya mencionado “SDK” de Android.

Finalmente, la totalidad de las aplicaciones en Android están comprimidas en formato “APK”, pudiéndose instalar en los dispositivos compatibles mediante un explorador de archivos. Aunque el sistema operativo Android se encuentra en la gran mayoría de Smartphone, existen aplicaciones exclusivas para otros tipos de dispositivos móviles, tales como tablets, notebooks, Google Tv e incluso complementos inteligentes (relojes, auriculares, gafas, etc.).

### 2.2.4 Versiones

Desde la aparición del sistema operativo Android en 2007 hasta la actualizada, se han producido diversos cambios en el mismo, dándose así una cierta diversidad de versiones. Es importante analizar esta diversidad antes de comenzar un nuevo proyecto en Android, desde las funcionalidades que aportan cada versión hasta la cuota de mercado.

A continuación se detallará cada versión lanzada hasta la fecha, describiendo las funcionalidades introducidas.

La nomenclatura de las versiones viene dada por:

- Versión: número decimal identificativo de la misma, partiendo inicialmente desde el 1.0.
- Nivel de API: número entero identificativo, partiendo inicialmente desde el 1 y aumentando en una unidad con cada nueva versión.
- Nombre: nombre identificativo de cada versión correspondiéndose con un dulce, a partir de la tercera versión dicho nombre sigue un orden alfabético.

#### **Android 1.0 Nivel de API 1 (septiembre 2008)**

Se trata de la primera versión de Android. En ella se incluía un navegador web, soporte para SMS y MMS, y una aplicación básica para el uso de la cámara del dispositivo.

### **Android 1.1 Nivel de API 2 (febrero 2009)**

Se trata de la primera versión comercial, por ello es la opción a escoger si se quiere desarrollar una aplicación completamente compatible para todos los dispositivos Android. La cual solo sirvió para optimizar y solucionar problemas presentes en la primera versión, ya que apenas se añadieron nuevas funcionalidades.

### **Android 1.5 Nivel de API 3 (abril 2009, Cupcake)**

Se trató de la primera gran revisión en cuanto a adición de funcionalidades. En ella se incorporaron la funcionalidad de teclado con predicción de texto, se incluyó auto sincronización y el soporte para bluetooth. Además, se personaliza la interfaz de usuario mediante widgets y las transiciones animadas entre pantallas.

### **Android 1.6 Nivel de API 4 (diciembre 2009, Donut)**

Esta revisión incluye la funcionalidad de realizar búsquedas avanzadas en el dispositivo, desde la pantalla principal de forma más rápida y eficiente, integrando tanto la búsqueda en la red como en el propio dispositivo. Además, se posibilita que las aplicaciones puedan utilizar distintos tamaños y resoluciones. A su vez, se proporciona soporte para CDMA/EVDO, VPNs y tecnología de síntesis de texto a voz.

### **Android 2.0.x Nivel de API 5 (octubre 2009, Éclair)**

Esta revisión incluye como mejoras en la gestión de contactos, permitiendo la sincronización de múltiples cuentas al dispositivo. Además, se incorporan nuevas funcionalidades a la cámara del dispositivo, como flash, zoom y múltiples opciones de edición. También incorpora un diccionario inteligente en el teclado, facilitando la experiencia al usuario. A su vez, se proporciona soporte HTML5, Bluetooth 2.1, así como una optimización de la velocidad del hardware.

### **Android 2.1 Nivel de API 7 (enero 2010, Éclair)**

Se consideró como una actualización menor, por ello conservo el nombre de su predecesora. En esta revisión se incluyó el reconocimiento de voz, como sustituto del teclado. A su vez, se incluyeron sistemas para modificar la resolución de pantalla, administrar la configuración de la caché de aplicaciones y el almacenamiento web. También se incluyeron nuevos métodos para manipular bases de datos en la librería WebKit.

### **Android 2.2.x Nivel de API 8 (mayo 2010, Froyo)**

En esta revisión se consigue un incremento de las prestaciones Software, tanto en el sistema como en sus aplicaciones, gracias a la optimización de los recursos Hardware.

Además, se permite la instalación de aplicaciones en la tarjeta microSD, así como las actualizaciones automáticas de las mismas. Por último, cabe destacar el soporte de reconocimiento de voz para interactuar con las aplicaciones y la aparición de nuevos motores de reconocimiento por terceras partes.

### **Android 2.3.x Nivel de API 9 (diciembre 2010, Gingerbread)**

Con la aparición de las tablets, esta revisión soporta mayores resoluciones de pantalla mediante las resoluciones WXGA y superiores. Se incorpora una actualización de la interfaz de usuario, destacando la funcionalidad multitáctil.

También se da soporte a varias cámaras, pensado en la cámara frontal del dispositivo. Originado por esta inclusión Hardware, se incluye la función de reconocimiento facial.

A su vez, destaca la inclusión de soporte a telefonía VoIP/SIP, reproducción de video WebM/VP8 y codificación de audio AAC, tecnología NFC y soporte para nuevos sensores incluido en el Hardware de los dispositivos.

### **Android 3.x Nivel de API 11 (febrero 2011, Honeycomb)**

Esta revisión fue lanzada para mejorar la interfaz de usuario en los nuevos dispositivos móviles (Tablets), incluyendo una optimización para este tipo de dispositivos. Esta nueva interfaz es utilizada por las versiones predecesoras a ella, simplificando la función multitarea.

A su vez, se añaden mejoras multimedia incluyendo soporte a HTTP Live Streaming, a la protección de derechos musicales y para la transferencia de archivos multimedia a través de USB, mediante los protocolos MTP y PTP.

En mayo de ese mismo año, aparece una nueva revisión (Android 3.1), solucionando problemas de la anterior versión e incluyendo soporte para manejar dispositivos conectados por USB.

Un mes más tardes, surge una nueva revisión (Android 3.2), terminando de estabilizar y optimizar las versiones predecesoras. Además de incluir posibilidad de sincronización multimedia desde un dispositivos de almacenamiento externo.

### **Android 4.0.x Nivel de API 14 (octubre 2011, Ice Cream Sandwich)**

Con esta versión se pretende integrar la gran mayoría de los dispositivos, puesto que los Smartphone usaban en su mayoría la versión 2.x, mientras las tablets usaban la 3.x.

Además, se establece Google Chrome como navegador por defecto, se mejora los APIs de reconocimiento facial, voz, redes sociales, calendario, revisor ortográfico y bases de datos entre otros.

### **Android 4.1 Nivel de API 16 (julio 2012, Jelly Bean)**

Se trata de una revisión de las más importantes, ya que se modifica la versión del núcleo del sistema operativo, ahora basada en Linux 3.0.31. Con ello se persigue mejorar la estabilidad, funcionalidad y rendimiento de la interfaz de usuario.

Se potencia la búsqueda por voz y se mejoran la seguridad permitiendo cifrar aplicaciones, también se permiten las actualizaciones parciales de las aplicaciones.

A su vez, una de las mejoras más importantes es el uso de la entrada por voz sin necesidad de conexión a la Internet. Esto se hace posible ya que el intérprete se encuentra en el propio dispositivo. También se mejoró la entrada por teclado, otorgándole la capacidad de predecir la próxima palabra que el usuario dará entrada.

#### **Android 4.2.x Nivel de API 17 (noviembre 2012, Jelly Bean)**

Esta revisión aporta como característica más importante el soporte multiusuario para tablets. Cada usuario puede tener su cuenta independiente, con sus aplicaciones y su correspondiente aplicación, permitiendo el intercambio de cuentas sin necesidad de cerrar sesión.

Se incorpora un nuevo teclado predictivo, que permite al usuario escribir deslizando su dedo por las letras del teclado. Además, se mejora el servicio de Google Now y se ofrece un rediseño de la interfaz de usuario para el uso de la cámara.

#### **Android 4.3.x Nivel de API 18 (mayo 2013, Jelly Bean)**

En esta versión se incorpora un administrador de permisos, pudiendo el usuario desactivar algunos de los permisos de las aplicaciones. En cuanto a la interfaz gráfica, se da soporte de manera oficial a OpenGL ES 3.0.

A su vez, se permite el control parental, permitiendo decidir al usuario que aplicaciones o contenidos puede acceder un determinado usuario.

#### **Android 4.4.x Nivel de API 19 (octubre 2013, KitKat)**

Se trata de la versión actual. Con esta versión se trata de erradicar la segmentación de versión, para ello se ha optimizado el rendimiento del sistema, minimizando el uso de recursos del mismo (memoria, cpu, batería, etc.), para poder ser utilizado tanto por terminales antiguos o de baja gama como por terminales de alta gama.

Además, se proporciona soporte con Bluetooth 4.0 y su propio perfil MAP (Message Access Profile), por lo que los vehículos con tecnología Bluetooth puede intercambiar mensajes con sus dispositivos.

También se integran nuevas funcionalidades de seguridad, como el Administrador de dispositivos, permitiendo al usuario localizar o borrar su dispositivo en caso de robo o pérdida. Además, se incrementa la seguridad de las zonas de pruebas de aplicaciones con el módulo de seguridad Security-Enhanced Linux.

A su vez, se proporciona una nueva arquitectura abierta para pagos NFC que funciona con cualquier operador móvil y permite que las aplicaciones administren tu información de pago en la nube o en cualquier dispositivo.

## 2.3 Sistemas de localización

El Sistema de Posicionamiento Global (GPS) es un sistema de localización, diseñado por el Departamento de Defensa de los Estados Unidos con fines militares para proporcionar estimaciones precisas de posición, velocidad y tiempo; operativo desde 1995 utiliza conjuntamente una red de ordenadores y una constelación de 24 satélites para determinar por triangulación, la altitud, longitud y latitud de cualquier objeto en la superficie terrestre [GPS].

En el ámbito civil y alegando razones de seguridad sólo se permite el uso de un subconjunto degradado de señales GPS. Sin embargo la comunidad civil ha encontrado alternativas para obtener una excelente precisión en la localización mediante las denominadas técnicas diferenciales. Gracias a ellas las aplicaciones civiles han experimentado un gran crecimiento y actualmente existen más de 70 fabricantes de receptores GPS.

El sistema se descompone en tres segmentos básicos:

- Segmento espacio: conformado por los satélites GPS.
- Segmento control: conformado por las estaciones encargadas de mantener los satélites en órbita y las antenas terrestres que se comunican con dichos satélites.
- Segmento usuario: conformado por las antenas y receptores pasivos situados en la tierra. Dichos receptores son los encargados de calcular la distancia, posición y tiempo.

El objetivo principal de un sistema GPS es calcular la posición de un punto cualquiera en un espacio de coordenadas (x,y,z) partiendo del cálculo de las distancias del punto a un mínimo de tres satélites cuya localización es conocida.

A continuación se describen las fuentes de error que en la actualidad afectan de forma significativa a las medidas realizadas con el GPS:

- Perturbación ionosférica. La ionosfera está formada por una capa de partículas cargadas eléctricamente que modifican la velocidad de las señales de radio que la atraviesan.
- Fenómenos meteorológicos. En la troposfera, cuna de los fenómenos meteorológicos, el vapor de agua afecta a las señales electromagnéticas disminuyendo su velocidad. Los errores generados son similares en magnitud a los causados por la ionosfera, pero su corrección es prácticamente imposible.
- Imprecisión en los relojes. Los relojes atómicos de los satélites presentan ligeras desviaciones a pesar de su cuidadoso ajuste y control, lo mismo sucede con los relojes de los receptores.
- Error multi-sonda. Las señales transmitidas desde los satélites pueden sufrir reflexiones antes de alcanzar el receptor. Los receptores modernos emplean técnicas avanzadas de

proceso de señal y antenas de diseño especial para minimizar este error, que resulta muy difícil de modelar al ser dependiente del entorno donde se ubique la antena GPS.

- Interferencia "Disponibilidad Selectiva S/A". Constituye la mayor fuente de error y es introducida deliberadamente por el estamento militar.

## 2.4 Sistemas de dialogo

En este apartado trataremos los sistemas de dialogo basados en procesamiento del habla, detallando sus características principales, describiendo su arquitectura y profundizando el tipo de sistema empleado en la aplicación.

Estos sistemas reciben como entrada palabras u oraciones del lenguaje natural, expresadas de forma oral por el usuario, generando como salida palabras u oraciones del lenguaje natural, expresadas del mismo modo por parte del sistema.

La finalidad de los mismos es emular el comportamiento inteligente del ser humano, a través del dialogo con otra persona con el objetivo de realizar una tarea en cuestión [GRIOL].

El ideal de un sistema de dialogo es la interacción simultánea con un ser humana mediante la voz, para ello se deberían garantizar los siguientes servicios [LLIESTERRI]:

- Reconocimiento del habla espontanea.
- Comprensión de enunciados sin restricciones de contenido.
- Respuesta con sentido contextual, gramaticalmente bien formada y pragmáticamente adecuada.
- Respuesta con voz completamente natural.

En la actualidad, estos sistemas están limitados tanto en hardware como en software al encontrarse sujetos a las restricciones del reconocimiento automática del habla, limitados a dominios específicos en comprensión y respuesta, necesidad de estrategias de verificación. Además de estar condicionados por el habla sintetizada y existir el problema del dialogo de forma espontánea.

A su vez, estos sistemas posibilitan la accesibilidad, que en ciertos casos no es posible con los sistemas tradicionales como el teclado, ratón, pantalla. Esta característica es utilizada en particular en personas con discapacidades visuales o motoras, en aplicaciones accesibles desde vehículos, donde el conductor no puede emplear la vista para tareas ajenas a la conducción.

El esquema empleado para el desarrollo de sistemas de dialogo tiene a conjuntar varios módulos genéricos, que deben sincronizarse para garantizar los requerimientos del usuario [LLIESTERRI].

Como se puede observar en la **Ilustración 5**, muestra la arquitectura básica de un sistema de dialogo basado en el procesamiento del habla. Esta arquitectura está representada de

cinco módulos que tratan tanto la voz entrante como la resultante, de igual manera estos módulos toman recursos de otros tres módulos para garantizar los servicios anteriormente descritos.

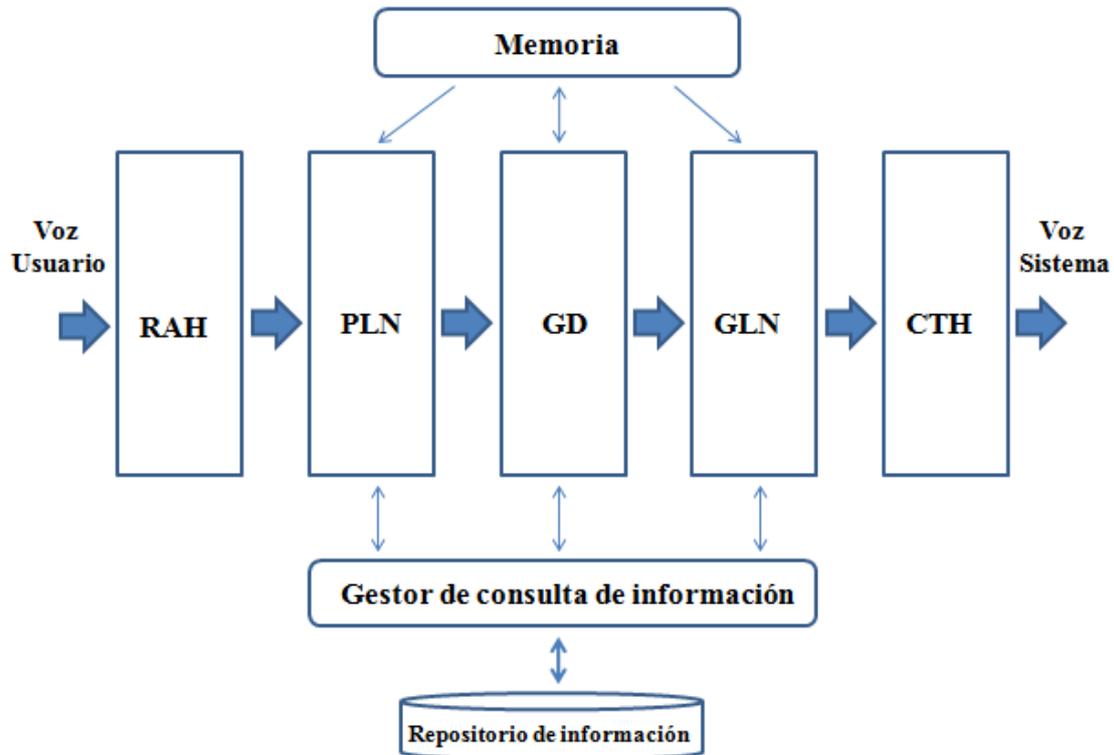


Ilustración 5 - Arquitectura modular de un sistema de diálogo

A continuación, se detallarán las principales características y funcionalidades de los módulos mostrados en la Ilustración 5:

**Módulo de Reconocimiento Automático del Habla (RAH):** Procesa la voz entrante y la transforma en la secuencia de palabras reconocidas mediante reglas probabilísticas.

**Módulo de Procesamiento del Lenguaje Natural (PLN):** Procesa el significado de las palabras reconocidas por el módulo de reconocimiento automático del habla, obteniendo la representación semántica de la oración y expresándolo en términos de un lenguaje especificado para la tarea.

**Gestor del diálogo (GD):** Establece el flujo del sistema tras cada interacción del usuario. Este módulo es en el que reside la inteligencia de estos sistemas, puesto que su finalidad es garantizar que la interacción con el usuario sea cómoda e inteligente. Para ello utilizada la interpretación semántica de la entrada del usuario, el contexto del dialogo, la información disponible y estado del sistema en tiempo real. Además, de la información registrada en la base de datos y la estrategia definida.

**Módulo de Generación del Lenguaje Natural (GLN):** Este módulo es el encargado de la generación de oraciones o palabras, gramaticalmente correctas y en un lenguaje lo más cercano al natural.

**Módulo de Conversión Texto Habla (CTH):** En este módulo se lleva a cabo la conversión de la oración o palabra resultante en una señal de audio.

**Módulo de memoria:** Este módulo es el encargado de almacenar las representaciones semánticas de las muestras de entrada, así como las oraciones generadas previamente por el sistema, proporcionando de este modo un registro histórico a los módulos de procesamiento del lenguaje natural, gestor de diálogo y generación del lenguaje natural. De este modo, el sistema puede resolver de manera más eficiente las referencias anafóricas existentes en las oraciones introducidas por el usuario, pudiendo conocer que oraciones ha expresado previamente y utilizando esa información para establecer un contexto de diálogo.

**Módulo gestor de consulta de información de la aplicación:** Este módulo es el encargado de crear las consultas necesarias al repositorio de información, procesándolas y proporcionando la información obtenida al módulo de gestor del diálogo.  
Repositorio de información: Se trata de un repositorio o base de datos donde se almacena la información necesaria del sistema y que será consultada por el gestor de consulta de información.

A continuación detallaremos las características del sistema de diálogo empleado en la aplicación del Trabajo de Fin de Grado:

- Sistema de diálogo guiado por el sistema: La interacción se realiza mediante turnos concretos y fijados entre pregunta y respuesta, restringiendo las iniciativas del usuario.
- Dependiente del hablante: Un sistema dependiente del hablante está desarrollado para funcionar para un hablante determinado (solo hispanoparlantes). Estos sistemas son más simples y económicos de desarrollar y la precisión es mayor que la de los sistemas independientes del hablante.
- Comunicación Multimodal: Se utilizan varios canales de comunicación. Por ejemplo, dispositivos de entrada como el habla, el teclado.
- Monolingüe: Permiten interacción sólo mediante un idioma.
- Reconocimiento continuo: Un sistema de reconocimiento continuo funciona sobre un lenguaje en el que las palabras están conectadas, es decir, no están separadas por pausas.

# Capítulo 3

---

## 3 Aplicación

### 3.1 Descripción general del sistema

En este apartado se describen las características generales del conjunto de módulos que componen la aplicación desarrollada: objetivos, requisitos, funcionalidad y arquitectura. Seguidamente se describen las herramientas utilizadas, así como los recursos necesarios para el desarrollo de la aplicación. Por último, se detallan las operaciones más generales, que se han tenido en cuenta a lo largo de la aplicación.

#### 3.1.1 Objetivos

Actualmente el desarrollo tecnológico proporciona a las personas la automatización o simplificación de las tareas cotidianas. Este hecho se ha acrecentado con los dispositivos móviles, ya que la gran mayoría de las personas lleva consigo sus “Smartphone”.

La potencia de estos dispositivos permite infinidad de recursos, puesto que en cada dispositivo se aúnan una gran cantidad de sensores y servicios software, que permiten a un usuario automatizar o simplificar algún proceso de su vida cotidiana. Además, estos dispositivos ofrecen esta gran cantidad de recursos sin tener que invertir una gran cantidad de dinero.

Por ello, podemos definir los objetivos de este Trabajo Fin de Grado en crear una aplicación basada en sistemas de localización y consulta por voz, permitiendo al usuario automatizar la gestión de perfiles.

#### 3.1.2 Requisitos

En ausencia de un entorno empresarial convencional, la especificación de requisitos será tomada por el actor más cercano a lo que pudiera interpretarse como nuestro cliente (tutor) y el público objetivo, al que va dirigida la aplicación (cualquier usuario con dispositivo Android).

##### 3.1.2.1 Formato de la especificación de requisitos

Las tablas de requisitos estarán compuestas por:

- Identificador: Clave única que identifica cada requisito. Estará compuesta por las siglas de Requisito Funcional (RF), Requisito No Funcional (RNF) o Requisito Restrictivo (RR) seguido de un número único asignado de dos cifras.
- Fuente: Origen del requisito.
- Nombre: Nombre del requisito, será significativo en relación a la descripción.

- Descripción: Explicación detallada del requisito.
- Prioridad: Nivel de preferencia que tiene el requisito para el equipo de desarrollo.
- Verificabilidad: Apartado formado por los identificadores de las pruebas que verifican el cumplimiento del requisito.
- Necesidad: Necesidad de la implementación del requisito en el sistema
- Claridad: Claridad del requisito

IDENTIFICADOR: RFXX o RFNXX o RRXX		
<b>NOMBRE:</b>		
<b>FUENTE:</b>		
<b>PRIORIDAD:</b>	<input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	<b>NECESIDAD:</b> <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL
<b>CLARIDAD:</b>	<input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	<b>VERIFICABILIDAD:</b> <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
<b>DESCRIPCIÓN:</b>		

Tabla 1 - Formato de requisitos

### 3.1.2.2 Requisitos funcionales

IDENTIFICADOR: RF01		
<b>NOMBRE:</b> Acceso a la aplicación		
<b>FUENTE:</b> Adrián García Herías		
<b>PRIORIDAD:</b>	<input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	<b>NECESIDAD:</b> <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL
<b>CLARIDAD:</b>	<input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	<b>VERIFICABILIDAD:</b> <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
<b>DESCRIPCIÓN:</b> El usuario no deberá registrarse para acceder a la aplicación.		

Tabla 2 - Requisito RF01

IDENTIFICADOR: RF02		
<b>NOMBRE:</b> Perfiles		
<b>FUENTE:</b> Adrián García Herías		
<b>PRIORIDAD:</b>	<input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	<b>NECESIDAD:</b> <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL
<b>CLARIDAD:</b>	<input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	<b>VERIFICABILIDAD:</b> <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
<b>DESCRIPCIÓN:</b> El usuario deberá crear, modificar o borrar sus perfiles.		

Tabla 3 - Requisito RF02

IDENTIFICADOR: RF03		
<b>NOMBRE:</b> Reglas		
<b>FUENTE:</b> Adrián García Herías		
<b>PRIORIDAD:</b>	<input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	<b>NECESIDAD:</b> <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL
<b>CLARIDAD:</b>	<input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	<b>VERIFICABILIDAD:</b> <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
<b>DESCRIPCIÓN:</b> El usuario deberá crear, modificar o borrar sus reglas de activación de perfiles.		

Tabla 4 - Requisito RF03

IDENTIFICADOR: RF04		
<b>NOMBRE:</b> Activación manual de perfiles		
<b>FUENTE:</b> Adrián García Herías		
<b>PRIORIDAD:</b>	<input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	<b>NECESIDAD:</b> <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL
<b>CLARIDAD:</b>	<input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	<b>VERIFICABILIDAD:</b> <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
<b>DESCRIPCIÓN:</b> El usuario deberá activar el perfil de manera táctil o voz.		

Tabla 5 - Requisito RF04

### 3.1.2.3 Requisitos no funcionales

IDENTIFICADOR: RNF01		
<b>NOMBRE:</b>	Universalidad de dispositivos	
<b>FUENTE:</b>	Adrián García Herías	
<b>PRIORIDAD:</b>	<input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	<b>NECESIDAD:</b> <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL
<b>CLARIDAD:</b>	<input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	<b>VERIFICABILIDAD:</b> <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
<b>DESCRIPCIÓN:</b>	Se debe permitir el uso de la aplicación en cualquier tipo de dispositivo móvil Android.	

Tabla 6 - Requisito RNF01

IDENTIFICADOR: RNF02		
<b>NOMBRE:</b>	Universalidad de conexiones	
<b>FUENTE:</b>	Adrián García Herías	
<b>PRIORIDAD:</b>	<input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	<b>NECESIDAD:</b> <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL
<b>CLARIDAD:</b>	<input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	<b>VERIFICABILIDAD:</b> <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
<b>DESCRIPCIÓN:</b>	Se debe permitir el uso de cualquier tipo de comunicación inalámbrica.	

Tabla 7 - Requisito RNF02

IDENTIFICADOR: RNF03		
<b>NOMBRE:</b>	Entrada de voz	
<b>FUENTE:</b>	Adrián García Herías	
<b>PRIORIDAD:</b>	<input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	<b>NECESIDAD:</b> <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL
<b>CLARIDAD:</b>	<input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	<b>VERIFICABILIDAD:</b> <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
<b>DESCRIPCIÓN:</b>	Se debe permitir el uso de voz como modo de entrada de datos.	

Tabla 8 - Requisito RNF03

### 3.1.2.4 Requisitos restrictivos

IDENTIFICADOR: RR01		
<b>NOMBRE:</b>	Versión Android	
<b>FUENTE:</b>	Adrián García Herías	
<b>PRIORIDAD:</b>	<input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	<b>NECESIDAD:</b> <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL
<b>CLARIDAD:</b>	<input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	<b>VERIFICABILIDAD:</b> <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
<b>DESCRIPCIÓN:</b>	La aplicación solo funcionara en dispositivos Android con versión superior a la 8.	

Tabla 9 - Requisito RR01

IDENTIFICADOR: RR02		
<b>NOMBRE:</b>	Conexión internet	
<b>FUENTE:</b>	Adrián García Herías	
<b>PRIORIDAD:</b>	<input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	<b>NECESIDAD:</b> <input checked="" type="checkbox"/> ESENCIAL <input type="checkbox"/> DESEABLE <input type="checkbox"/> OPCIONAL
<b>CLARIDAD:</b>	<input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA	<b>VERIFICABILIDAD:</b> <input checked="" type="checkbox"/> ALTA <input type="checkbox"/> MEDIA <input type="checkbox"/> BAJA
<b>DESCRIPCIÓN:</b>	La aplicación solo funcionará en dispositivos con acceso a internet.	

Tabla 10 - Requisito RR02

### 3.1.3 Funcionalidad

La aplicación desarrollada para este Trabajo Fin de Grado proporciona al usuario la siguiente lista de funcionalidades:

- Lista de perfiles
- Crear perfil
- Modificar o borrar perfil
- Lista de reglas
- Crear regla horaria
- Crear regla localización
- Modificar o eliminar reglas
- Gestión automática de perfiles
- Gestión manual de perfiles

Todas estas funcionalidades podrán ser accesibles mediante el teclado del dispositivo y alguna de ellas a través de voz. A su vez, estas funcionalidades o módulos serán debidamente detallados en el apartado **“3.2 Descripción detallada de los módulos del sistema”**.

Puesto que se ha establecido la versión 9 de nivel de API, la aplicación deberá tener acceso a Internet para poder procesar la entrada de audio y acceder a los servicios de geolocalización.

La interacción con la aplicación comienza cuando el usuario accede por primera vez a la misma, permitiendo a usuario inicialmente crear nuevos perfiles y reglas, pudiendo diferenciar entre horarias o por localización, esta última permitiendo el uso de la entrada por voz.

Una vez el usuario ha registrado sus preferencias en la aplicación, es la misma aplicación la que se encarga internamente de la gestión de dichas preferencias. Además, se permite al usuario la gestión manual de sus preferencias, tanto por voz como de manera táctil.

A su vez, el usuario podrá en todo momento modificar o borrar sus preferencias.

### 3.1.4 Herramientas del sistema

A continuación se especifican las herramientas utilizadas para el desarrollo de la aplicación. Dichas herramientas no ha supuesto coste alguno, puesto que son herramientas de código libre.

### 3.1.4.1 Eclipse

Este es el aplicativo utilizado como entorno de desarrollo para la ejecución de esta aplicación, utilizando la versión más actual (Eclipse Kepler). Dicho aplicativo está compuesto por un conjunto de herramientas de programación de código abierto multiplataforma.

Al tratarse de una herramienta de código abierto, no ha sido necesario ningún tipo de gasto económico para su uso, siendo accesible este recurso mediante la página oficial de Eclipse [ECLIPSE].

A su vez, para el desarrollo de aplicaciones Android, es necesaria la instalación de un plugin provisto por Google correspondiente al desarrollo de aplicaciones Android.

#### SDK Android

Una vez realizado el paso anterior, ya será posible desarrollar aplicaciones Android, no así ejecutarlas, para ello será necesario disponer del SDK de Android.

Posteriormente a su instalación, se podrán hacer pruebas sobre dispositivos Android emulados, por ello se asume que es necesario para el desarrollo de aplicaciones Android, la integración de Eclipse y SDK. Por ello, desde la web de desarrolladores de Android, se facilita el paquete completo [ECLIPSE+SDK].

#### Sistema de ficheros

Como se observa en la Ilustración 6, la jerarquía de archivos de la aplicación se divide en cuatro paquetes, todos ellos bajo los directorios “android.tfg”, el nombre contiguo a ese path, viene dado por la función que determina en la aplicación.

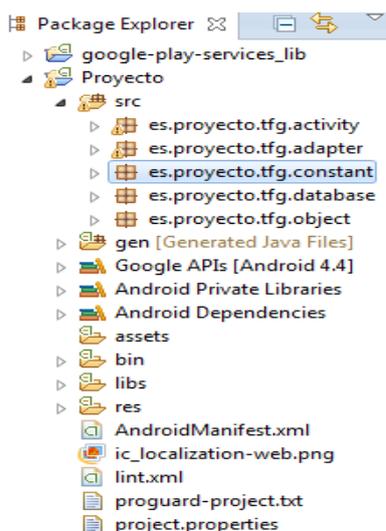


Ilustración 6 - Sistema de ficheros de la aplicación

El paquete “activity” compone la parte principal de la aplicación. Ya que está compuesto por las actividades que proporcionan la funcionalidad de la aplicación. Habiendo tantas actividades como interfaces haya.

El paquete “adapter”, se encarga de cambiar el comportamiento de un componente del layout (listView). Este componente será utilizado por las actividades en el momento de representar gráficamente la información.

El paquete “constant”, incluye las constantes más utilizadas a lo largo de la implementación de la aplicación.

El paquete “database”, contiene los ficheros Java encargados de la interacción con la base de datos.

El paquete “object”, incluye los ficheros Java encargados de crear los objetos que interactuarán en la aplicación (profile y rule).

### **3.1.5 Recursos del sistema**

En este apartado se detallan los recursos necesarios para el correcto funcionamiento de la aplicación, realizando una breve descripción de cada uno de ellos, así como la función que desempeñan dentro de la aplicación.

#### **3.1.5.1 Google Voice Recognizer**

Se trata de un sistema de reconocimiento de voz, integrado por Google en el sistema operativo Android desde la API 8. Dicha funcionalidad es de uso gratuito, aunque requiere conexión a internet por parte del dispositivo para el reconocimiento de la frecuencia de voz.

Este servicio será accesible por el usuario toda vez que se pulse sobre un determinado botón y finalizará la captura de muestras una vez el usuario termine de transmitir las.

#### **3.1.5.2 Google Places**

Google Places es un servicio que devuelve información sobre sitios, definidos en la API como establecimientos, ubicaciones geográficas o sitios de interés importantes, mediante solicitudes HTTP. En las solicitudes de sitios, las ubicaciones se especifican en forma de coordenadas de latitud/longitud.

Estas coordenadas permitirán la actualización automática de los perfiles, segundo la proximidad del dispositivo a la ubicación introducida previamente por el usuario.

### 3.1.5.3 SQLite

Para almacenar las preferencias de los usuarios en nuestra aplicación, es necesaria la creación de una base de datos. Para ello, se ha empleado el servicio integrado en la librería de Android (SQLite), cuyo lenguaje es prácticamente similar al SQL.

SQLite es un motor de base de datos, que implementa una pequeña librería desarrollada en lenguaje C. Al ser necesario acceder a las preferencias del usuario cada corto espacio de tiempo, es más óptimo establecer la base de datos en el propio dispositivo, reduciendo así el número de consultas a través de Internet.

A continuación, se exponen otras de las ventajas que han llevado a usar SQLite en el desarrollo de la aplicación:

- Es un recurso con un espacio reducido, lo cual resulta atractivo puesto que la memoria de los dispositivos no infinita, ni tan grande a la de un ordenador.
- Se ha optimizado la eficiencia de este motor de base de datos, frente a otros tradicionales como MySQL.
- No es necesario separar los procesos de escritura y lectura, puesto que lo realiza sobre archivos que se encuentra en el propio almacenamiento de dispositivo.
- Optimiza el tamaño de los registros, para usar únicamente el espacio requerido en cada momento.
- Es un motor multiplataforma y portable.

### 3.1.6 Implementación de las operaciones generales

En este apartado se describirán las operaciones más comunes utilizadas a lo largo de la implementación de la aplicación.

#### 3.1.6.1 Reconocimiento de voz

En dispositivos Android tenemos la posibilidad de utilizar el servicio Google Voice Recognizer, para el reconocimiento de voz, sin ningún tipo de coste, aunque requiere conexión a internet por parte del dispositivo. Este servicio se soporta desde el nivel de API 8.

Este servicio será accesible por el usuario cada vez que se pulse un determinado botón o se capture una gramática preconcebida.

En primer lugar se den proporcionar los permisos necesarios para que la aplicación pueda acceder al servicio descrito.

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

A continuación, se define el proceso que realizará el procesamiento del audio de entrada a texto, definiendo el lenguaje del hablante:

```
private void startVoiceRecognitionActivity() {  
    // Definición del intent  
    Intent intent = new  
Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);  
    // Indicamos el modelo de lenguaje para el intent  
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL, "es-  
ES");}
```

Aunque no se ha llevado a cabo en esta aplicación, su ponemos el caso de pasar texto a voz. Para ello, es necesario indicar el idioma

```
tts.setLanguage(Locale.SPAIN);
```

Se puede establecer el nivel de tono mediante el uso de la función `setPitch`. Por defecto el valor es 1, pudiendo establecerse valores inferiores y superiores al inicial.

```
tts.setSpeechRate(2);
```

También se podrá establecer la velocidad mediante el uso de la función `setSpeechRate`. Por defecto toma el valor de 1, pudiendo establecerse valores inferiores y superiores al inicial.

```
tts.setPitch(0.6);
```

Por último, reproducimos el texto introducido con las características anteriormente mencionadas.

```
tts.speak(text, TextToSpeech.QUEUE_FLUSH, null);
```

### 3.1.6.2 Localización en Android

En los dispositivos Android tenemos la posibilidad de obtener la localización del dispositivo a través de dos métodos, usando los satélites GPS o mediante la red, ya sea a través de WIFI o triangulando la localización a partir de los puntos de acceso para móviles.

Para este proyecto se ha utilizada la localización por ambos métodos, permitiendo al usuario un ahorro de la batería y permitiéndole seguir localizado si no fuera accesible alguno de los métodos anteriormente mencionados.

Si el usuario se encuentra en una localización registrada en sus preferencias, se actualizará el estado de su perfil a la preferencia de dicho usuario. Si el sistema no pudiese calcular la posición de usuario, se mantendría el perfil en uso, a no ser que otra regla lo modificase.

En primer lugar se den proporcionar los permisos necesarios para que la aplicación pueda acceder a los servicios de localización.

```
<uses-permission
android:name="es.proyecto.tfg.permission.MAPS_RECEIVE" />
  <uses-permission
android:name="com.google.android.providers.gsf.permission.READ_G
SERVICES"/>
  <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION"/>
  <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

Dentro de la actividad, seleccionamos el proveedor con mayor precisión, garantizando que el dispositivo tenga algún proveedor de localización activado.

```
LocationManager lm = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);
// Creating a criteria object to retrieve provider
Criteria criteria = new Criteria();
// Getting the name of the best provider
provider = locationManager.getBestProvider(criteria, true);
```

Para evitar que se usen las llamadas "Mock Locations" y que el usuario no pueda introducir localizaciones falseadas a la aplicación, se utiliza el siguiente código:

```
if (Settings.Secure.getString(getContentResolver(),
Settings.Secure.ALLOW MOCK_LOCATION).equals("0"))
return false;
else return true;
```

A continuación, obtendremos la localización a través del mejor proveedor accesible por el dispositivo y la posicionará en el mapa.

```

public void onLocationChanged(Location location) {
    mLatitude = location.getLatitude();
    mLongitude = location.getLongitude();
    LatLng latLng = new LatLng(mLatitude, mLongitude);

    mGoogleMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));

    mGoogleMap.animateCamera(CameraUpdateFactory.zoomTo(12));
}

```

En la actividad se pide la localización actual al proveedor, esta petición volverá a repetirse cada dos minutos (medida tomada para el ahorro de batería y datos):

```

if (provider!=null)
locationManager.requestLocationUpdates(provider, 120*1000, 0,
this);

```

### 3.1.6.3 Gestión de bases de datos

Para la gestión de datos dentro de la aplicación se ha utilizado el motor de base de datos SQLite. Esta funcionalidad está contenida en el paquete "android.tfg.database" como anteriormente se ha mencionado.

Creemos la base de datos:

```

// Database Version
private static final int DATABASE_VERSION = 1;
// Database Name
private static final String DATABASE_NAME = "tfg_DB";

public BaseDeDatos(Context context) {
    super(context, DATABASE_NAME, null, DATABASE_VERSION);
}

```

Generamos las distintas tablas, así como los distintos campos que las componen:

```

private static final String[] COLUMNS_PROFILE =
{KEY_ID,KEY_NAME,KEY_VOLUMEN,KEY_VIBRACION,
    KEY_BRILLO,KEY_SLEEP,KEY_ROTAR,KEY_WIFI,
KEY_DATOS,KEY_BLUETOOTH, KEY_ENABLE};

private static final String[] COLUMNS_RULE = {KEY_ID, KEY_NAME,

```

```

KEY_HORA_INI, KEY_HORA_FIN, KEY_PROFILE});

private static final String[] COLUMNS_LOCATION = {KEY_ID, KEY_NAME,
KEY_LONGITUD,KEY_LATITUD,KEY_PROFILE};

@Override
public void onCreate(SQLiteDatabase db) { //Creacion de las tablas de
la base de datos
    db.execSQL("CREATE TABLE Profiles (id INTEGER PRIMARY KEY
AUTOINCREMENT, name TEXT, volumen TEXT, vibracion BOOLEAN, brillo
TEXT, sleep TEXT, rotar BOOLEAN, wifi BOOLEAN,datos BOOLEAN,bluetooth
BOOLEAN, activar BOOLEAN)");

    db.execSQL("CREATE TABLE Rules (id INTEGER PRIMARY KEY
AUTOINCREMENT, name TEXT, horaI TEXT, horaF TEXT, KEY_PROFILE TEXT");

    db.execSQL("CREATE TABLE Locations (id INTEGER PRIMARY KEY
AUTOINCREMENT, name TEXT, logitud TEXT, latitud TEXT, KEY_PROFILE
TEXT");
}

```

A continuación se detallan los campos de cada una de las tablas así como su finalidad en la aplicación.

#### **Tabla 'Profiles'**

En esta tabla se almacena toda la información relativa al perfil del usuario. Los siguientes campos se pueden ver actualizados en todo el transcurso de la administración del perfil (creación, edición y borrado).

- ID: Este campo contiene el identificador único del perfil.
- Name: Este campo contiene el nombre del perfil.
- Volumen: Este campo contiene las preferencias de volumen del dispositivo.
- Vibración: Este campo contiene el estado de la vibración del dispositivo.
- Brillo: Este campo contiene las preferencias de brillo de la pantalla.
- Sleep: Este campo contiene las preferencias del tiempo de espera de la pantalla.
- Rotar: Este campo contiene el estado de la rotación de la pantalla.
- Wifi: Este campo contiene el estado del Wifi del dispositivo.
- Datos: Este campo contiene el estado de los datos del dispositivo.
- Bluetooth: Este campo contiene el estado de los datos del dispositivo.

#### **Tabla 'Rules'**

En esta tabla se almacena toda la información relativa a las reglas implementadas por el usuario. Los siguientes campos se pueden ver actualizados en todo el transcurso de la administración del perfil (edición y borrado) y de la administración de la regla.

- ID: Este campo contiene el identificador único de la regla.
- Name: Este campo contiene el nombre de la regla.
- Horal: Este campo contiene la hora inicial de la regla.
- HoraF: Este campo contiene la hora final de la regla.
- Profile: Este campo contiene el nombre del perfil.

### Tabla 'Locations'

En esta tabla se almacena toda la información relativa a las localizaciones registradas por el usuario. Los siguientes campos se pueden ver actualizados en todo el transcurso de la administración del perfil (edición y borrado) y de la administración de la localización.

- ID: Este campo contiene el identificador único de la localización.
- Name: Este campo contiene el nombre de la localización.
- Longitud: Este campo contiene la longitud de la localización.
- Latitud: Este campo contiene la latitud de la localización.
- Profile: Este campo contiene el nombre del perfil.

Puesto que la información almacenada en la base de datos, no se trata de una información sensible del usuario, se ha tomado la decisión de no cifrar la base de datos. Con esta medida se quiere favorecer eficiencia de la aplicación, penalizando en un recurso superfluo para la misma.

Seguidamente implementamos alguna de las operaciones básicas de la base de datos, para ello solo utilizaremos como ejemplo la tabla "Profiles":

- Añadir perfil

```
public void addProfile(Profile profile){
    //for logging
    Log.d("addProfile", profile.toString());

    // 1. get reference to writable DB
    SQLiteDatabase db = this.getWritableDatabase();

    // 2. create ContentValues to add key "column"/value
    ContentValues values = new ContentValues();
    values.put(KEY_NAME, profile.getName());
    values.put(KEY_VOLUMEN, profile.getVolumen());
    values.put(KEY_VIBRACION, profile.isVibracion());
    values.put(KEY_BRILLO, profile.getBrillo());
    values.put(KEY_SLEEP, profile.getSleep());
    values.put(KEY_ROTAR, profile.isRotar());
    values.put(KEY_WIFI, profile.isWifi());
    values.put(KEY_DATOS, profile.isDatos());
    values.put(KEY_BLUETOOTH, profile.isBluetooth());
    values.put(KEY_ENABLE, profile.isActivar());

    // 3. insert
```

```

        db.insert(TABLE_ProfileS, // table
            null, //nullColumnHack
            values); // key/value -> keys = column names/ values = column
values

        // 4. close
        db.close();
    }

```

- Obtener perfil por ID

```

public Profile getProfile(int id){

    // 1. get reference to readable DB
    SQLiteDatabase db = this.getReadableDatabase();

    // 2. build query
    Cursor cursor =
        db.query(TABLE_ProfileS, // a. table
            COLUMNS_PROFILE, // b. column names
            " id = ?", // c. selections
            new String[] { String.valueOf(id) }, // d. selections
args
            null, // e. group by
            null, // f. having
            null, // g. order by
            null); // h. limit

    // 3. if we got results get the first one
    if (cursor != null)
        cursor.moveToFirst();

    // 4. build book object
    Profile profile = new Profile();
    profile.setId(Integer.parseInt(cursor.getString(0)));
    profile.setName(cursor.getString(1));
    profile.setVolumen(cursor.getString(2));

    profile.setVibracion(Boolean.parseBoolean(cursor.getString(3)));
    profile.setBrillo(cursor.getString(4));
    profile.setSleep(cursor.getString(5));
    profile.setRotar(Boolean.parseBoolean(cursor.getString(6)));
    profile.setWifi(Boolean.parseBoolean(cursor.getString(7)));
    profile.setDatos(Boolean.parseBoolean(cursor.getString(8)));

    profile.setBluetooth(Boolean.parseBoolean(cursor.getString(9)));

    profile.setActivar(Boolean.parseBoolean(cursor.getString(10)));

```

```

//log
Log.d("getProfile("+id+")", profile.toString());

// 5. return book
return profile;
}

```

- Obtener lista de perfiles

```

public List<Profile> getAllProfiles() {
    List<Profile> Profiles = new LinkedList<Profile>();

    // 1. build the query
    String query = "SELECT * FROM " + TABLE_ProfileS;

    // 2. get reference to writable DB
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery(query, null);

    // 3. go over each row, build Profile and add it to list
    Profile profile = null;
    if (cursor.moveToFirst()) {
        do {
            profile = new Profile();
            profile.setId(Integer.parseInt(cursor.getString(0)));
            profile.setName(cursor.getString(1));
            profile.setVolumen(cursor.getString(2));

            profile.setVibracion(Boolean.parseBoolean(cursor.getString(3)));
            profile.setBrillo(cursor.getString(4));
            profile.setSleep(cursor.getString(5));

            profile.setRotar(Boolean.parseBoolean(cursor.getString(6)));
            profile.setWifi(Boolean.parseBoolean(cursor.getString(7)));
            profile.setDatos(Boolean.parseBoolean(cursor.getString(8)));
            profile.setBluetooth(Boolean.parseBoolean(cursor.getString(9)));
            profile.setActivar(Boolean.parseBoolean(cursor.getString(10)));

            Profiles.add(profile);
        } while (cursor.moveToNext());
    }

    Log.d("getAllProfiles()", profile.toString());
}

```

```
        return Profiles;
    }
```

- Actualizar perfil

```
public int updateProfile(Profile profile) {

    // 1. get reference to writable DB
    SQLiteDatabase db = this.getWritableDatabase();

    // 2. create ContentValues to add key "column"/value
    ContentValues values = new ContentValues();
    values.put(KEY_NAME, profile.getName());
    values.put(KEY_VOLUMEN, profile.getVolumen());
    values.put(KEY_VIBRACION, profile.isVibracion());
    values.put(KEY_BRILLO, profile.getBrillo());
    values.put(KEY_SLEEP, profile.getSleep());
    values.put(KEY_ROTAR, profile.isRotar());
    values.put(KEY_WIFI, profile.isWifi());
    values.put(KEY_DATOS, profile.isDatos());
    values.put(KEY_BLUETOOTH, profile.isBluetooth());
    values.put(KEY_ENABLE, profile.isActivar());

    // 3. updating row
    int i = db.update(TABLE_ProfileS, //table
        values, // column/value
        KEY_ID+" = ?", // selections
        new String[] { String.valueOf(profile.getId()) });
//selection args

    // 4. close
    db.close();

    return i;
}
```

- Borrar perfil

```
public void deleteProfile(Profile profile) {

    // 1. get reference to writable DB
    SQLiteDatabase db = this.getWritableDatabase();

    // 2. delete
    db.delete(TABLE_ProfileS, //table name
        KEY_ID+" = ?", // selections
        new String[] { String.valueOf(profile.getId()) });
//selections args
}
```

```
// 3. close
db.close();

//log
Log.d("deleteProfile", profile.toString());
}
```

## 3.2 Descripción detallada de los módulos del sistema

En este apartado se describe al detalle las funcionalidades, arquitecturas y casos de uso para cada uno de los módulos que componen la aplicación.

### 3.2.1 Inicio

#### 3.2.1.1 Funcionalidad

Se trata del módulo principal de la aplicación. Desde él se accederá a todas las tres interfaces genéricas de la aplicación (Lista Perfiles, Lista Reglas y Lista Localizaciones).

En la pantalla de inicio, se cargará la lista con los tipos de objetos de la aplicación (Profile, Rule y Location). Permitiendo de este modo separar la funcionalidad de estos objetos.

Como se observa en la imagen anterior, pulsando sobre uno de los tres campos, se cambiará a su interfaz correspondiente. Permitiendo al usuario alcanzar una lista de objeto anteriormente pulsado.

#### 3.2.1.2 Arquitectura

A continuación se describirán las acciones que se llevan a cabo una vez el usuario haya pulsado sobre uno de los objetos de la lista de la interfaz inicio.

- Se pulsa sobre “Profile”

Una vez pulsado sobre el campo “Profile”, la aplicación realizará una transición hacia la interfaz “Profile”.

- Se pulsa sobre “Rule”

Una vez pulsado sobre el campo “Rule”, la aplicación realizará una transición hacia la interfaz “Rule”.

- Se pulsa sobre “Location”

Una vez pulsado sobre el campo “Location”, la aplicación realizará una transición hacia la interfaz “Location”.

### 3.2.1.3 Casos de uso

A continuación, se mostrará el flujo de aplicación una vez el usuario interactúa con la aplicación. La letra 'S' indicarán las acciones realizadas por el sistema, mientras que la 'U' muestra las acciones realizadas por el usuario.

S	->	Carga la interfaz gráfica correspondiente al módulo de inicio.
U	->	Selecciona uno de los objetos de la lista ("Profile", "Rule" o "Location").
S	->	Carga la interfaz gráfica correspondiente al módulo del objeto seleccionado.

### 3.2.2 Perfil

#### 3.2.2.1 Funcionalidad

Se trata de uno de los tres módulos básicos de la aplicación. Desde él se accederá a las funcionalidades correspondientes al objeto perfil de forma directa (Crear regla y Modificar/Eliminar regla) e indirectamente a las funcionalidades de los objetos regla y localización.

En la pantalla de perfiles, se cargará la lista con los objetos perfil que existan en cada momento, además de mostrar si está activado el perfil. Permitted mediante pulsación sobre ellos acceder a las funcionalidades de los mismos.

En la Ilustración 7, se muestra la interfaz correspondiente al módulo perfil.



Ilustración 7 - Interfaz correspondiente al módulo perfil.

Como se puede observar en la imagen anterior, pulsando sobre un objeto perfil, se permitirá realizar cualquier tipo de modificación sobre el mismo. A su vez, si pulsamos sobre "Nuevo profile", se permitirá crear un nuevo perfil.

### 3.2.2.2 Arquitectura

A continuación se describirán las acciones que se llevan a cabo una vez el usuario haya pulsado sobre los áreas anteriormente descritos.

- Se pulsa sobre un objeto perfil.

Una vez pulsado sobre el objeto perfil, la aplicación realizará una transición hacia la interfaz **Modificar/Eliminar** perfil.

- Se pulsa sobre “Nuevo Profile”

Una vez pulsado sobre el campo “Nuevo Profile”, la aplicación realizará una transición hacia la interfaz **Crear** perfil.

- Se mantiene pulsado el objeto perfil.

Manteniendo pulsado el objeto perfil, se producirá la activación manual del mismo.

### 3.2.2.3 Casos de uso

A continuación, se mostrará el flujo de aplicación una vez el usuario interactúa con la aplicación. La letra ‘S’ indicarán las acciones realizadas por el sistema, mientras que la ‘U’ muestra las acciones realizadas por el usuario.

S	->	Carga la interfaz gráfica correspondiente al módulo perfil.
U	->	Selecciona un objeto perfil.
S	->	Carga la interfaz gráfica correspondiente al módulo modificar/eliminar perfil.

S	->	Carga la interfaz gráfica correspondiente al módulo perfil.
U	->	Selecciona el campo “Nuevo Profile”.
S	->	Carga la interfaz gráfica correspondiente al módulo crear perfil.

S	->	Carga la interfaz gráfica correspondiente al módulo perfil.
U	->	Se mantiene pulsado el objeto perfil durante 1 segundo.
S	->	Se procesa la información, procediendo a activar el objeto perfil en cuestión. Realizando los cambios en el dispositivo correspondientes a las preferencias registradas en el objeto perfil.

### 3.2.3 Crear perfil

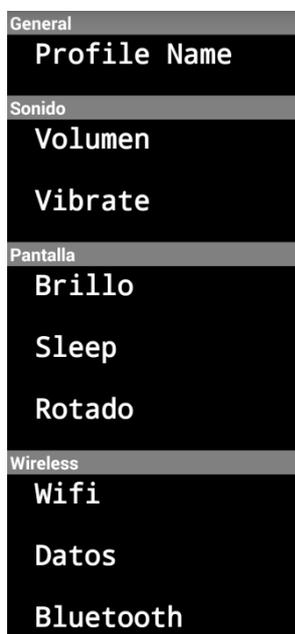
#### 3.2.3.1 Funcionalidad

Se trata de una interfaz del módulo 'Perfil'. Desde la cual se accederá a la funcionalidad de **Crear perfil**.

En la pantalla de crear perfil, se cargará una lista con las preferencias que el usuario podrá decidir que se modifiquen una vez este activado el perfil. Permitiendo mediante pulsación acceder a dichas preferencias para seleccionarl

Una vez, se han registrado todas las preferencias y se ha validado que eso es así, aparecerá el botón 'guardar'.

En la Ilustración 8, se muestra la interfaz correspondiente al módulo crear perfil



**Ilustración 8 - Interfaz correspondiente al módulo crear perfil.**

Como se puede observar en la imagen anterior, pulsando sobre cualquiera de las preferencias, se accederá a un pop up para proceder a su configuración.

En Ilustración 9, se muestra la interfaz correspondiente al módulo perfil



Ilustración 9 - Interfaz correspondiente al módulo crear perfil.

Como se puede observar en la imagen anterior, una vez se ha verificado que todas las preferencias han sido completadas, aparece el botón 'guardar'.

### 3.2.3.2 Arquitectura

A continuación se describirán las acciones que se llevan a cabo una vez el usuario haya pulsado sobre los áreas anteriormente descritos.

- Se pulsa sobre el campo "Profile Name".

Una vez pulsado sobre el campo "Profile Name", la aplicación mostrará un pop up para ingresar el nombre del perfil.

En la Ilustración 10, se muestra el pop up correspondiente al nombre del perfil.

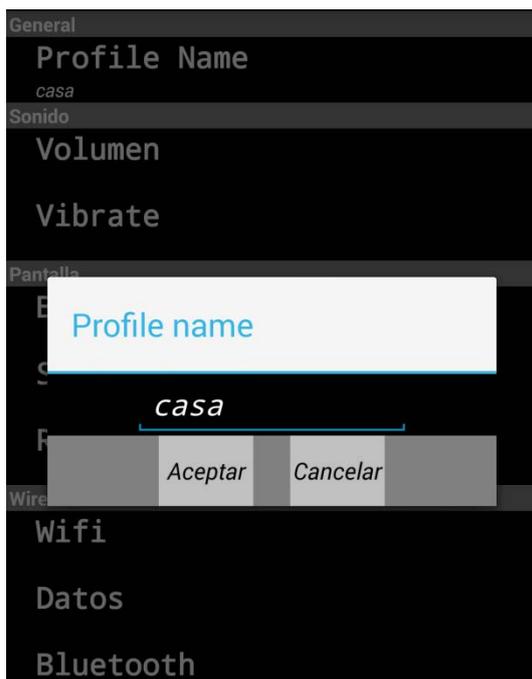


Ilustración 10 - Pop up correspondiente al nombre del perfil.

- Se pulsa sobre el campo “Volumen”.

Una vez pulsado sobre el campo “Volumen”, la aplicación mostrará un pop up para ingresar la configuración del volumen del perfil.

En la Ilustración 11, se muestra el pop up correspondiente al volumen del perfil.

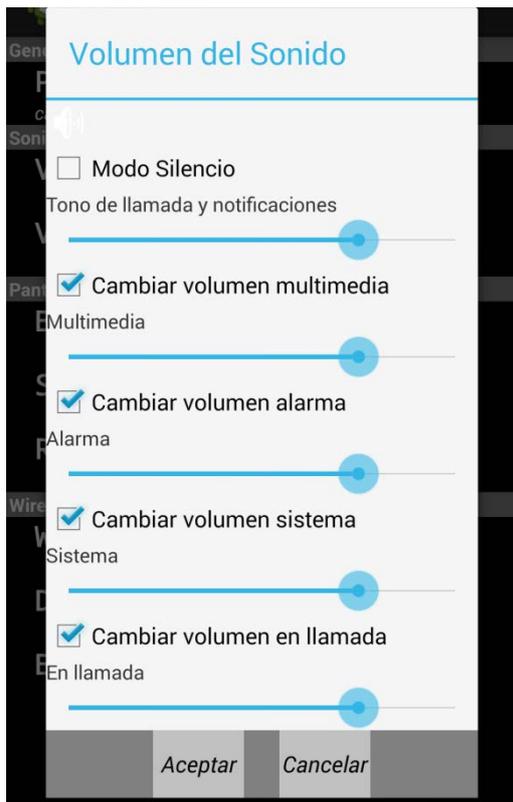


Ilustración 11 - Pop up correspondiente al volumen del perfil.

- Se pulsa sobre el campo “Vibrate”.

Una vez pulsado sobre el campo “Vibrate”, la aplicación mostrará un pop up para ingresar la configuración de la vibración del perfil.

En la Ilustración 12, se muestra el pop up correspondiente a la vibración del perfil.

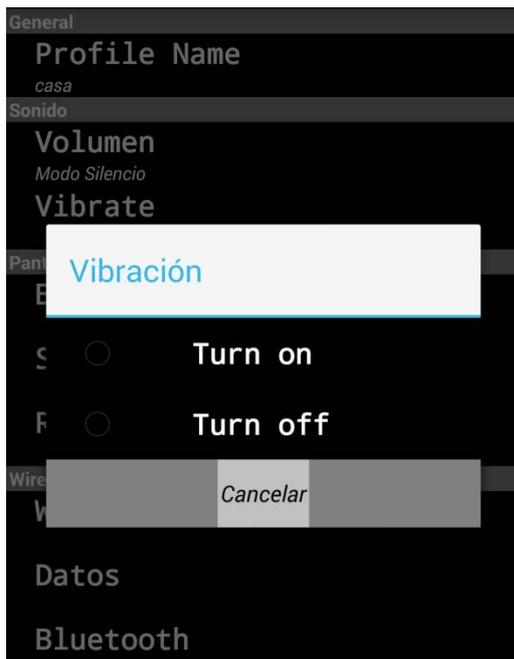


Ilustración 12 - Pop up correspondiente a la vibración del perfil.

- Se pulsa sobre el campo “Brillo”.

Una vez pulsado sobre el campo “Brillo”, la aplicación mostrará un pop up para ingresar la configuración del brillo de la pantalla del dispositivo.

En la Ilustración 13, se muestra el pop up correspondiente al brillo de la pantalla del dispositivo.

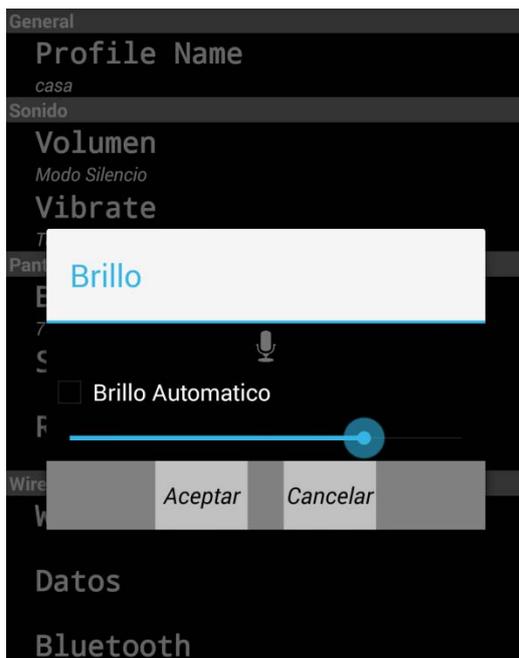


Ilustración 13 - Pop up correspondiente al brillo de la pantalla del dispositivo.

- Se pulsa sobre el campo “Sleep”.

Una vez pulsado sobre el campo “Sleep”, la aplicación mostrará un pop up para ingresar la configuración del tiempo de espera de la pantalla del dispositivo.

En la Ilustración 14, se muestra el pop up correspondiente al tiempo de espera de la pantalla del dispositivo.

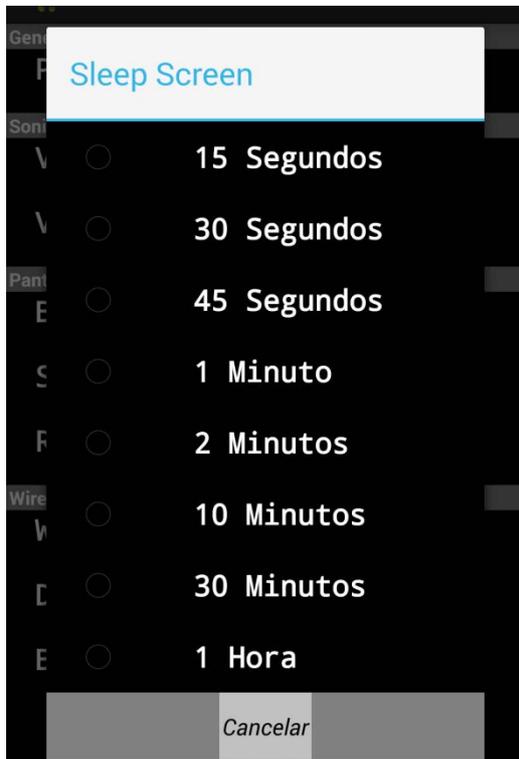


Ilustración 14 - Pop up correspondiente al tiempo de espera de la pantalla del dispositivo.

- Se pulsa sobre el campo “Rotate”.

Una vez pulsado sobre el campo “Rotate”, la aplicación mostrará un pop up para ingresar la configuración del rotado de la pantalla del dispositivo.

En la Ilustración 15, se muestra el pop up correspondiente al rotado de la pantalla del dispositivo.

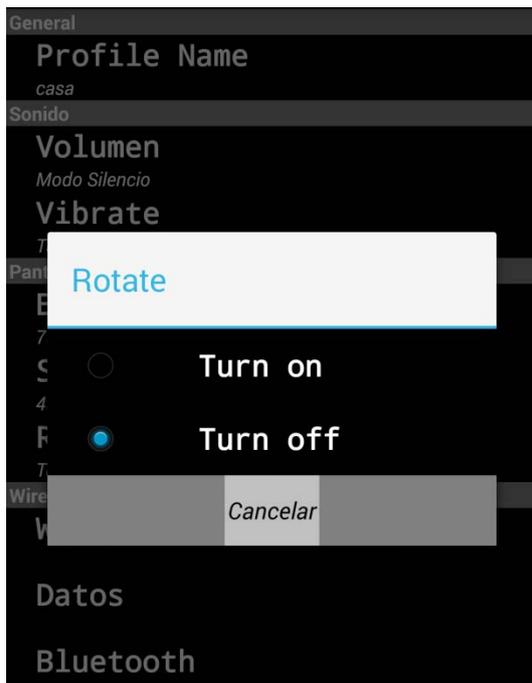


Ilustración 15 - Pop up correspondiente al rotado de la pantalla del dispositivo.

- Se pulsa sobre el campo “Wifi”.

Una vez pulsado sobre el campo “Wifi”, la aplicación mostrará un pop up para ingresar la configuración del Wifi del perfil.

En la Ilustración 16, se muestra el pop up correspondiente al Wifi del perfil.

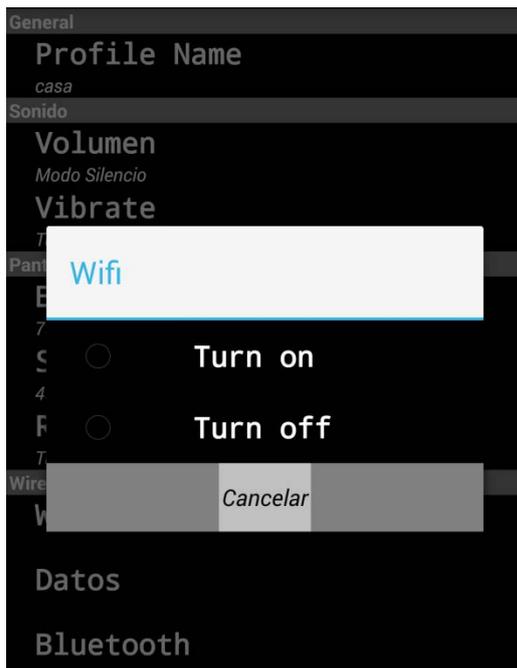


Ilustración 16 - Pop up correspondiente al Wifi del perfil.

- Se pulsa sobre el campo “Datos”.

Una vez pulsado sobre el campo “Datos”, la aplicación mostrará un pop up para ingresar la configuración de los Datos del perfil.

En la Ilustración 17, se muestra el pop up correspondiente al Datos del perfil.

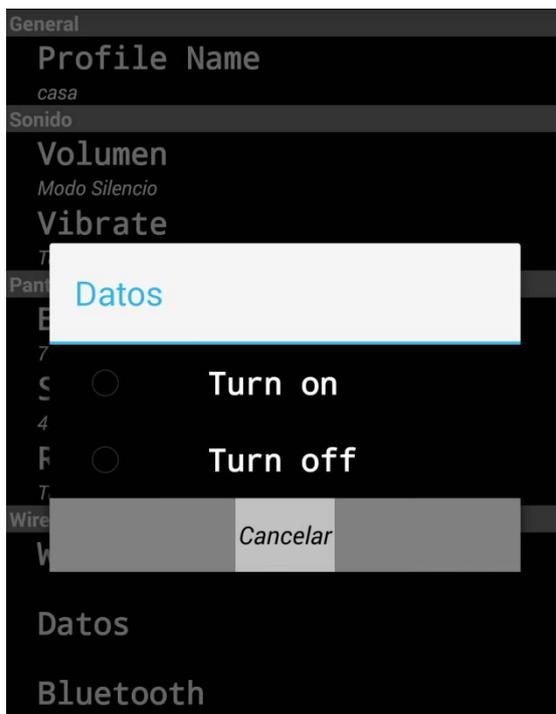


Ilustración 17 - Pop up correspondiente al Datos del perfil.

- Se pulsa sobre el campo “Bluetooth”.

Una vez pulsado sobre el campo “Bluetooth”, la aplicación mostrará un pop up para ingresar la configuración de los Bluetooth del perfil.

En la Ilustración 18, se muestra el pop up correspondiente al Bluetooth del perfil.

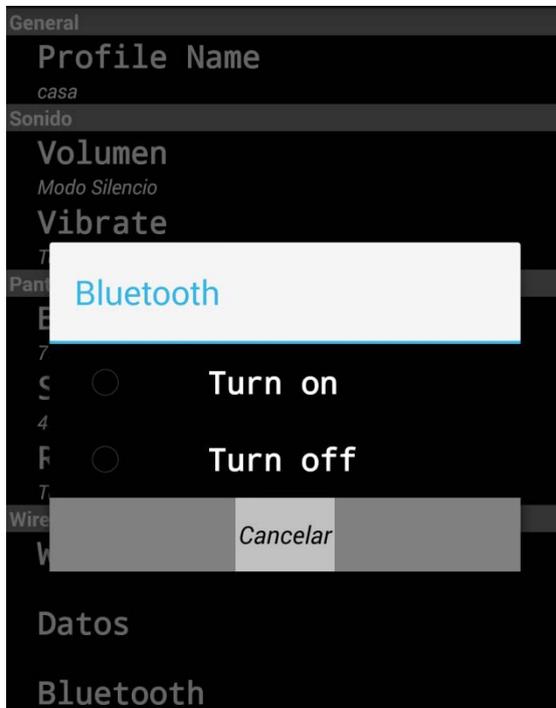


Ilustración 18 - Pop up correspondiente al Bluetooth del perfil.

### 3.2.3.3 Casos de uso

A continuación, se mostrará el flujo de aplicación una vez el usuario interactúa con la aplicación. La letra ‘S’ indicarán las acciones realizadas por el sistema, mientras que la ‘U’ muestra las acciones realizadas por el usuario.

S	->	Carga la interfaz gráfica correspondiente al módulo crear perfil.
U	->	Selecciona el campo el campo “Profile Name”.
S	->	Carga la interfaz gráfica correspondiente.
U	->	Introduce un nombre para el perfil.
S	->	Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar.
U	->	Pulsa el botón guardar para crear el perfil.
S	->	Se almacena dichas preferencias en la base de datos.

S	->	Carga la interfaz gráfica correspondiente al módulo crear perfil.
U	->	Selecciona el campo el campo “Volumen”.

S -> Carga la interfaz gráfica correspondiente.  
U -> Introduce la configuración deseada mediante el check box “Modo silencio” o mediante las barras que aparecen una vez marcas el check box “Cambiar volumen...”.  
S -> Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar.  
U -> Pulsa el botón guardar para crear el perfil.  
S -> Se almacena dichas preferencias en la base de datos.

S -> Carga la interfaz gráfica correspondiente al módulo crear perfil.  
U -> Selecciona el campo el campo “Vibrate”.  
S -> Carga la interfaz gráfica correspondiente.  
U -> Introduce la configuración deseada (activado o desactivado).  
S -> Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar.  
U -> Pulsa el botón guardar para crear el perfil.  
S -> Se almacena dichas preferencias en la base de datos.

S -> Carga la interfaz gráfica correspondiente al módulo crear perfil.  
U -> Selecciona el campo el campo “Brillo”.  
S -> Carga la interfaz gráfica correspondiente.  
U -> Introduce la configuración deseada mediante el check box “Brillo automático” o mediante la barra que aparecen estando desmarcado el check box “Brillo automático”.  
S -> Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar.  
U -> Pulsa el botón guardar para crear el perfil.  
S -> Se almacena dichas preferencias en la base de datos.

S -> Carga la interfaz gráfica correspondiente al módulo crear perfil.  
U -> Selecciona el campo el campo “Sleep”.  
S -> Carga la interfaz gráfica correspondiente.  
U -> Introduce la configuración deseada mediante un listado de opciones de tiempo.  
S -> Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar.  
U -> Pulsa el botón guardar para crear el perfil.  
S -> Se almacena dichas preferencias en la base de datos.

S -> Carga la interfaz gráfica correspondiente al módulo crear perfil.  
U -> Selecciona el campo el campo “Rotate”.  
S -> Carga la interfaz gráfica correspondiente.  
U -> Introduce la configuración deseada (activado o desactivado).

S	->	Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar.
U	->	Pulsa el botón guardar para crear el perfil.
S	->	Se almacena dichas preferencias en la base de datos.

S	->	Carga la interfaz gráfica correspondiente al módulo crear perfil.
U	->	Selecciona el campo el campo "Wifi".
S	->	Carga la interfaz gráfica correspondiente.
U	->	Introduce la configuración deseada (activado o desactivado).
S	->	Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar.
U	->	Pulsa el botón guardar para crear el perfil.
S	->	Se almacena dichas preferencias en la base de datos.

S	->	Carga la interfaz gráfica correspondiente al módulo crear perfil.
U	->	Selecciona el campo el campo "Datos".
S	->	Carga la interfaz gráfica correspondiente.
U	->	Introduce la configuración deseada (activado o desactivado).
S	->	Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar.
U	->	Pulsa el botón guardar para crear el perfil.
S	->	Se almacena dichas preferencias en la base de datos.

S	->	Carga la interfaz gráfica correspondiente al módulo crear perfil.
U	->	Selecciona el campo el campo "Bluetooth".
S	->	Carga la interfaz gráfica correspondiente.
U	->	Introduce la configuración deseada (activado o desactivado).
S	->	Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar.
U	->	Pulsa el botón guardar para crear el perfil.
S	->	Se almacena dichas preferencias en la base de datos.

### 3.2.4 Modificar o eliminar perfil

#### 3.2.4.1 Funcionalidad

Se trata de una interfaz del módulo 'Perfil'. Desde la cual se accederá a la funcionalidad de **Modificar/Eliminar** perfil y de forma indirecta a las interfaces **Modificar/Eliminar** regla y/o localización.

En la pantalla de modificar/eliminar perfil (misma interfaz gráfica que crear perfil), se cargará una lista con las preferencias registradas por el usuario en la creación del perfil, permitiendo mediante pulsación acceder a dichas preferencias para seleccionarlas.

Una vez, se han registrado todas las preferencias y se ha validado que eso es así, aparecerá el botón 'guardar' y 'eliminar'.

En la Ilustración 19, se muestra la interfaz correspondiente al módulo modificar/eliminar perfil.



Ilustración 19 - Interfaz correspondiente al módulo modificar/eliminar perfil.

Como se puede observar en la imagen anterior, a continuación del título de cada campo, aparece el valor introducido por el usuario. Una vez se ha verificado que todas las preferencias han sido completadas, aparece el botón 'guardar' y 'eliminar'.

### 3.2.4.2 Arquitectura

En cuanto a la arquitectura, se basa en la misma del módulo crear perfil.

### 3.2.4.3 Casos de uso

A continuación, se mostrará el flujo de aplicación una vez el usuario interactúa con la aplicación. La letra 'S' indicarán las acciones realizadas por el sistema, mientras que la 'U' muestra las acciones realizadas por el usuario.

S	->	Carga la interfaz gráfica correspondiente al módulo crear perfil.
U	->	Selecciona el campo el campo "Profile Name".
S	->	Carga la interfaz gráfica correspondiente.
U	->	Introduce un nombre para el perfil.
S	->	Procesa la información introducida y la muestra a continuación del título

del campo. Valida que todas las preferencias estén completas mostrando el botón guardar o eliminar.

U -> Pulsa el botón guardar o eliminar.

S -> Se actualizan o se eliminan dichas preferencias en la base de datos.

S -> Carga la interfaz gráfica correspondiente al módulo crear perfil.

U -> Selecciona el campo el campo "Volumen".

S -> Carga la interfaz gráfica correspondiente.

U -> Introduce la configuración deseada mediante el check box "Modo silencio" o mediante las barras que aparecen una vez marcas el check box "Cambiar volumen...".

S -> Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar o eliminar.

U -> Pulsa el botón guardar o eliminar.

S -> Se actualizan o se eliminan dichas preferencias en la base de datos.

S -> Carga la interfaz gráfica correspondiente al módulo crear perfil.

U -> Selecciona el campo el campo "Vibrate".

S -> Carga la interfaz gráfica correspondiente.

U -> Introduce la configuración deseada (activado o desactivado).

S -> Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar o eliminar.

U -> Pulsa el botón guardar o eliminar.

S -> Se actualizan o se eliminan dichas preferencias en la base de datos.

S -> Carga la interfaz gráfica correspondiente al módulo crear perfil.

U -> Selecciona el campo el campo "Brillo".

S -> Carga la interfaz gráfica correspondiente.

U -> Introduce la configuración deseada mediante el check box "Brillo automático" o mediante la barra que aparecen estando desmarcado el check box "Brillo automático".

S -> Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar o eliminar.

U -> Pulsa el botón guardar o eliminar.

S -> Se actualizan o se eliminan dichas preferencias en la base de datos.

S -> Carga la interfaz gráfica correspondiente al módulo crear perfil.

U -> Selecciona el campo el campo "Sleep".

S -> Carga la interfaz gráfica correspondiente.

U -> Introduce la configuración deseada mediante un listado de opciones de tiempo.

S -> Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar o eliminar.

U	->	Pulsa el botón guardar o eliminar.
S	->	Se actualizan o se eliminan dichas preferencias en la base de datos.

S	->	Carga la interfaz gráfica correspondiente al módulo crear perfil.
U	->	Selecciona el campo el campo "Rotate".
S	->	Carga la interfaz gráfica correspondiente.
U	->	Introduce la configuración deseada (activado o desactivado).
S	->	Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar o eliminar.
U	->	Pulsa el botón guardar o eliminar.
S	->	Se actualizan o se eliminan dichas preferencias en la base de datos.

S	->	Carga la interfaz gráfica correspondiente al módulo crear perfil.
U	->	Selecciona el campo el campo "Wifi".
S	->	Carga la interfaz gráfica correspondiente.
U	->	Introduce la configuración deseada (activado o desactivado).
S	->	Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar o eliminar.
U	->	Pulsa el botón guardar o eliminar.
S	->	Se actualizan o se eliminan dichas preferencias en la base de datos.

S	->	Carga la interfaz gráfica correspondiente al módulo crear perfil.
U	->	Selecciona el campo el campo "Datos".
S	->	Carga la interfaz gráfica correspondiente.
U	->	Introduce la configuración deseada (activado o desactivado).
S	->	Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar o eliminar.
U	->	Pulsa el botón guardar o eliminar.
S	->	Se actualizan o se eliminan dichas preferencias en la base de datos.

S	->	Carga la interfaz gráfica correspondiente al módulo crear perfil.
U	->	Selecciona el campo el campo "Bluetooth".
S	->	Carga la interfaz gráfica correspondiente.
U	->	Introduce la configuración deseada (activado o desactivado).
S	->	Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar o eliminar.
U	->	Pulsa el botón guardar o eliminar.
S	->	Se actualizan o se eliminan dichas preferencias en la base de datos.

## 3.2.5 Reglas

### 3.2.5.1 Funcionalidad

Se trata de uno de los tres módulos básicos de la aplicación. Desde él se accederá a las funcionalidades correspondientes al objeto regla de forma directa (Crear regla y Modificar/Eliminar regla).

En la pantalla de reglas, se cargará la lista con los objetos regla que existan en cada momento, además de mostrar una breve descripción de las preferencias del usuario. Permitiendo mediante pulsación sobre ellos acceder a las funcionalidades de los mismos.

A su vez, se encargará de modo interno de gestionar la activación automática de los perfiles, cuando el objeto regla con las preferencias descritas por el usuario (temporales), se corresponda con el momento temporal.

En la Ilustración 20, se muestra la interfaz correspondiente al módulo regla.



Ilustración 20 - Interfaz correspondiente al módulo regla.

Como se puede observar en la imagen anterior, pulsando sobre un objeto regla, se permitirá realizar cualquier tipo de modificación sobre el mismo. A su vez, si pulsamos sobre “Nueva regla”, se permitirá crear una nueva regla.

### 3.2.5.2 Arquitectura

A continuación se describirán las acciones que se llevan a cabo una vez el usuario haya pulsado sobre los áreas anteriormente descritos.

- Se pulsa sobre un objeto regla.

Una vez pulsado sobre el objeto regla, la aplicación realizará una transición hacia la interfaz **Modificar/Eliminar** regla.

- Se pulsa sobre “Nueva Regla”

Una vez pulsado sobre el campo “Nueva Regla”, la aplicación realizará una transición hacia la interfaz **Crear** regla.

### 3.2.5.3 Casos de uso

A continuación, se mostrará el flujo de aplicación una vez el usuario interactúa con la aplicación. La letra ‘S’ indicarán las acciones realizadas por el sistema, mientras que la ‘U’ muestra las acciones realizadas por el usuario.

S	->	Carga la interfaz gráfica correspondiente al módulo regla.
U	->	Selecciona un objeto regla.
S	->	Carga la interfaz gráfica correspondiente al módulo modificar/eliminar regla.

S	->	Carga la interfaz gráfica correspondiente al módulo regla.
U	->	Selecciona el campo “Nueva Regla”.
S	->	Carga la interfaz gráfica correspondiente al módulo crear regla.

## 3.2.6 Crear regla

### 3.2.6.1 Funcionalidad

Se trata de una interfaz del módulo ‘Regla’. Desde la cual se accederá a la funcionalidad de **Crear** regla.

En la pantalla de crear regla, se cargará una lista con las preferencias que el usuario podrá decidir para activar de manera automática el perfil correspondiente. Permitiendo mediante pulsación acceder a dichas preferencias para seleccionarlas.

Una vez, se han registrado todas las preferencias y se ha validado que eso es así, aparecerá el botón ‘guardar’.

### 3.2.6.2 Arquitectura

A continuación se describirán las acciones que se llevan a cabo una vez el usuario haya pulsado sobre los áreas anteriormente descritos.

- Se pulsa sobre el campo “Rule Name”.

Una vez pulsado sobre el campo “Rule Name”, la aplicación mostrará un pop up para ingresar el nombre de la regla (igual que ocurre para la interfaz crear perfil).

- Se pulsa sobre el campo “Time period”.

Una vez pulsado sobre el campo “Time period”, la aplicación mostrará un pop up para ingresar la hora de inicio y fin de la regla.

- Se pulsa sobre el campo “Profile”.

Una vez pulsado sobre el campo “Profile”, la aplicación mostrará un pop up para ingresar el nombre del perfil al que se asocia la regla.

### 3.2.6.3 Casos de uso

A continuación, se mostrará el flujo de aplicación una vez el usuario interactúa con la aplicación. La letra ‘S’ indicarán las acciones realizadas por el sistema, mientras que la ‘U’ muestra las acciones realizadas por el usuario.

S	->	Carga la interfaz gráfica correspondiente al módulo crear regla.
U	->	Selecciona el campo “Rule Name”.
S	->	Carga la interfaz gráfica correspondiente.
U	->	Introduce un nombre para la regla.
S	->	Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar.
U	->	Pulsa el botón guardar para crear la regla.
S	->	Se almacena dichas preferencias en la base de datos.

S	->	Carga la interfaz gráfica correspondiente al módulo crear regla.
U	->	Selecciona el campo “Time period”.
S	->	Carga la interfaz gráfica correspondiente.
U	->	Introduce el periodo temporal de la regla.
S	->	Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar.
U	->	Pulsa el botón guardar para crear la regla.
S	->	Se almacena dichas preferencias en la base de datos.

S	->	Carga la interfaz gráfica correspondiente al módulo crear regla.
U	->	Selecciona el campo “Profile”.
S	->	Carga la interfaz gráfica correspondiente.
U	->	Introduce un objeto perfil de la lista facilitada.
S	->	Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar.
U	->	Pulsa el botón guardar para crear la regla.
S	->	Se almacena dichas preferencias en la base de datos.

### 3.2.7 Modificar o eliminar regla

#### 3.2.7.1 Funcionalidad

Se trata de una interfaz del módulo 'Regla'. Desde la cual se accederá a la funcionalidad de **Modificar/Eliminar** regla.

En la pantalla de modificar/eliminar regla (misma interfaz gráfica que crear regla), se cargará una lista con las preferencias registradas por el usuario en la creación de la regla, permitiendo mediante pulsación acceder a dichas preferencias para seleccionarlas.

Una vez, se han registrado todas las preferencias y se ha validado que eso es así, aparecerá el botón 'guardar' y 'eliminar'.

#### 3.2.7.2 Arquitectura

En cuanto a la arquitectura, se basa en la misma del módulo crear regla.

#### 3.2.7.3 Casos de uso

A continuación, se mostrará el flujo de aplicación una vez el usuario interactúa con la aplicación. La letra 'S' indicarán las acciones realizadas por el sistema, mientras que la 'U' muestra las acciones realizadas por el usuario.

S	->	Carga la interfaz gráfica correspondiente al módulo crear regla.
U	->	Selecciona el campo "Rule Name".
S	->	Carga la interfaz gráfica correspondiente.
U	->	Introduce un nombre para la regla.
S	->	Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar o eliminar.
U	->	Pulsa el botón guardar o eliminar.
S	->	Se actualizan o se eliminan dichas preferencias en la base de datos.

S	->	Carga la interfaz gráfica correspondiente al módulo crear regla.
U	->	Selecciona el campo "Time period".
S	->	Carga la interfaz gráfica correspondiente.
U	->	Introduce el periodo temporal de la regla.
S	->	Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar o eliminar.
U	->	Pulsa el botón guardar o eliminar.
S	->	Se actualizan o se eliminan dichas preferencias en la base de datos.

S	->	Carga la interfaz gráfica correspondiente al módulo crear regla.
---	----	--

U	->	Selecciona el campo "Profile".
S	->	Carga la interfaz gráfica correspondiente.
U	->	Introduce un objeto perfil de la lista facilitada.
S	->	Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar o eliminar.
U	->	Pulsa el botón guardar o eliminar.
S	->	Se actualizan o se eliminan dichas preferencias en la base de datos.

### 3.2.8 Localizaciones

#### 3.2.8.1 Funcionalidad

Se trata de uno de los tres módulos básicos de la aplicación. Desde él se accederá a las funcionalidades correspondientes al objeto localización de forma directa (Crear localización y Modificar/Eliminar localización).

En la pantalla de localizaciones, se cargará la lista con los objetos localización que existan en cada momento, además de mostrar una breve descripción de las preferencias del usuario. Permitiendo mediante pulsación sobre ellos acceder a las funcionalidades de los mismos.

A su vez, se encargará de modo interno de gestionar la activación automática de los perfiles, cuando el objeto localización con las preferencias descritas por el usuario (ubicación), se corresponda con la ubicación de dispositivo, con una precisión de +-250m.

En la Ilustración 21, se muestra la interfaz correspondiente al módulo localización.



Ilustración 21 - Interfaz correspondiente al módulo localización.

Como se puede observar en la imagen anterior, pulsando sobre un objeto localización, se permitirá realizar cualquier tipo de modificación sobre el mismo. A su vez, si pulsamos sobre "Nueva localización", se permitirá crear una nueva localización.

### 3.2.8.2 Arquitectura

A continuación se describirán las acciones que se llevan a cabo una vez el usuario haya pulsado sobre los áreas anteriormente descritos.

- Se pulsa sobre un objeto localización.

Una vez pulsado sobre el objeto regla, la aplicación realizará una transición hacia la interfaz **Modificar/Eliminar** localización.

- Se pulsa sobre “Nueva localización”

Una vez pulsado sobre el campo “Nueva localización”, la aplicación realizará una transición hacia la interfaz **Crear** localización.

### 3.2.8.3 Casos de uso

A continuación, se mostrará el flujo de aplicación una vez el usuario interactúa con la aplicación. La letra ‘S’ indicarán las acciones realizadas por el sistema, mientras que la ‘U’ muestra las acciones realizadas por el usuario.

S	->	Carga la interfaz gráfica correspondiente al módulo localización.
U	->	Selecciona un objeto localización.
S	->	Carga la interfaz gráfica correspondiente al módulo modificar/eliminar localización.

S	->	Carga la interfaz gráfica correspondiente al módulo localización.
U	->	Selecciona el campo “Nueva localización”.
S	->	Carga la interfaz gráfica correspondiente al módulo crear localización.

## 3.2.9 Crear localización

### 3.2.9.1 Funcionalidad

Se trata de una interfaz del módulo ‘Localización’. Desde la cual se accederá a la funcionalidad de **Crear** localización.

En la pantalla de crear localización, se cargará una lista con las preferencias que el usuario podrá decidir para activar de manera automática el perfil correspondiente. Permitiendo mediante la pulsación acceder a dichas preferencias para seleccionarlas.

Una vez, se han registrado todas las preferencias y se ha validado que eso es así, aparecerá el botón ‘guardar’.

### 3.2.9.2 Arquitectura

A continuación se describirán las acciones que se llevan a cabo una vez el usuario haya pulsado sobre los áreas anteriormente descritos.

- Se pulsa sobre el campo “Location Name”.

Una vez pulsado sobre el campo “Location Name”, la aplicación mostrará un pop up para ingresar el nombre de la localización (igual que ocurre para la interfaz crear perfil).

- Se pulsa sobre el botón “micrófono”.

Una vez pulsado sobre el botón “micrófono”, se capturará la entrada por voz del usuario, correspondiente a una dirección y la aplicación mostrará un mapa con la posición del dispositivo y su recorrido hasta la dirección indicada, validando que es correcta.

En la Ilustración 22, se muestra la interfaz correspondiente al módulo Crear localización.

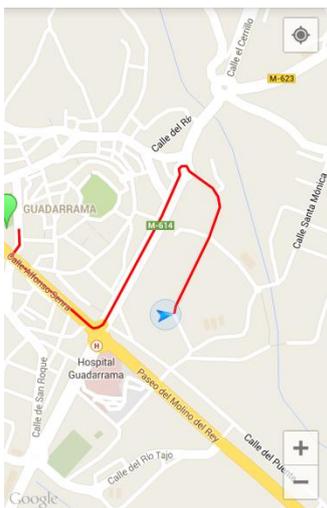


Ilustración 22 - Interfaz correspondiente al módulo Crear localización.

- Se pulsa sobre el campo “Profile”.

Una vez pulsado sobre el campo “Profile”, la aplicación mostrará un pop up para ingresar el nombre del perfil al que se asocia la localización.

### 3.2.9.3 Casos de uso

A continuación, se mostrará el flujo de aplicación una vez el usuario interactúa con la aplicación. La letra ‘S’ indicarán las acciones realizadas por el sistema, mientras que la ‘U’ muestra las acciones realizadas por el usuario.

S	->	Carga la interfaz gráfica correspondiente al módulo crear localización.
U	->	Selecciona el campo el campo “Location Name”.
S	->	Carga la interfaz gráfica correspondiente.

U	->	Introduce un nombre para la localización.
S	->	Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar.
U	->	Pulsa el botón guardar para crear la localización.
S	->	Se almacena dichas preferencias en la base de datos.

S	->	Carga la interfaz gráfica correspondiente al módulo crear localización.
S	->	Carga la interfaz gráfica correspondiente al módulo crear localización.
U	->	Selecciona el botón "micrófono".
S	->	Procesa la interacción e inicia el servicio Google Voice Recognizer.
U	->	Introduce la dirección mediante la voz.
S	->	Realiza consulta a la API de Google Places y esta le devuelve la longitud y latitud de la entrada. Además, carga una nueva interfaz gráfica, dibujando la ruta entre el origen y el destino, esperando su validación.
U	->	Se valida la localización.
S	->	Procesa la información obtenida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar.
U	->	Pulsa el botón guardar para crear la localización.
S	->	Se almacena dichas preferencias en la base de datos.

S	->	Carga la interfaz gráfica correspondiente al módulo crear localización.
U	->	Selecciona el campo "Profile".
S	->	Carga la interfaz gráfica correspondiente.
U	->	Introduce un objeto perfil de la lista facilitada.
S	->	Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar.
U	->	Pulsa el botón guardar para crear la localización.
S	->	Se almacena dichas preferencias en la base de datos.

### 3.2.10 Modificar o eliminar localización

#### 3.2.10.1 Funcionalidad

Se trata de una interfaz del módulo 'Regla'. Desde la cual se accederá a la funcionalidad de **Modificar/Eliminar** regla.

En la pantalla de modificar/eliminar regla (misma interfaz gráfica que crear regla), se cargará una lista con las preferencias registradas por el usuario en la creación de la regla, permitiendo mediante pulsación acceder a dichas preferencias para seleccionarlas.

Una vez, se han registrado todas las preferencias y se ha validado que eso es así, aparecerá el botón 'guardar' y 'eliminar'.

### 3.2.10.2 Arquitectura

En cuanto a la arquitectura, se basa en la misma del módulo crear localización.

### 3.2.10.3 Casos de uso

A continuación, se mostrará el flujo de aplicación una vez el usuario interactúa con la aplicación. La letra 'S' indicarán las acciones realizadas por el sistema, mientras que la 'U' muestra las acciones realizadas por el usuario.

S	->	Carga la interfaz gráfica correspondiente al módulo crear localización.
U	->	Selecciona el campo el campo "Location Name".
S	->	Carga la interfaz gráfica correspondiente.
U	->	Introduce un nombre para la localización.
S	->	Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar o eliminar.
U	->	Pulsa el botón guardar o eliminar.
S	->	Se actualizan o se eliminan dichas preferencias en la base de datos.

S	->	Carga la interfaz gráfica correspondiente al módulo crear localización.
S	->	Carga la interfaz gráfica correspondiente al módulo crear localización.
U	->	Selecciona el botón "micrófono".
S	->	Procesa la interacción e inicia el servicio Google Voice Recognizer.
U	->	Introduce la dirección mediante la voz.
S	->	Realiza consulta a la API de Google Places y esta le devuelve la longitud y latitud de la entrada. Además, carga una nueva interfaz gráfica, dibujando la ruta entre el origen y el destino, esperando su validación.
U	->	Se valida la localización.
S	->	Procesa la información obtenida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar o eliminar.
U	->	Pulsa el botón guardar o eliminar.
S	->	Se actualizan o se eliminan dichas preferencias en la base de datos.

S	->	Carga la interfaz gráfica correspondiente al módulo crear localización.
U	->	Selecciona el campo "Profile".
S	->	Carga la interfaz gráfica correspondiente.
U	->	Introduce un objeto perfil de la lista facilitada.
S	->	Procesa la información introducida y la muestra a continuación del título del campo. Valida que todas las preferencias estén completas mostrando el botón guardar o eliminar.
U	->	Pulsa el botón guardar o eliminar.
S	->	Se actualizan o se eliminan dichas preferencias en la base de datos.

# Capítulo 4

---

## 4 Conclusiones y Trabajos Futuros

### 4.1 Conclusiones

Finalmente, en este Trabajo Fin de Grado, se ha desarrollado una aplicación en el sistema operativo Android, basada en la inteligencia ambiental. Para ello, se ha hecho uso de sistemas de reconocimiento de voz y localización.

Esta aplicación permite al usuario automatizar la gestión de su perfil telefónico de forma automática, mediante la ubicación del propio dispositivo. Además, de permitirse consultas a través de la voz y el teclado del dispositivo.

Los servicios proporcionados por la aplicación, son prestados por una serie de módulos interconectados. De tal modo, el módulo Perfiles puede acceder al resto de módulos de la aplicación de forma directa o indirecta. Este módulo será el encargado de la gestión de los perfiles, ya sea crear, modificar, eliminar o gestionar el estado, de un perfil.

Desde el módulo Reglas, únicamente se podrá acceder a las funcionalidades de sus propios objetos (reglas). Por ello, este módulo nos ofrece la funcionalidad de crear, modificar y eliminar una regla. Estas reglas implementadas por el usuario. Son accesibles desde el módulo Perfiles, a través de su funcionalidad de gestión del estado del perfil, permitiendo de este modo la activación de perfil cuando el dispositivo del usuario se encuentre en la regla temporal predefinida.

A su vez, el módulo Localizaciones únicamente podrá acceder a las funcionalidades propias de sus objetos (localizaciones). Por ello, este módulo nos ofrece la funcionalidad de crear, modificar y eliminar una localización. Estas localizaciones introducidas por el usuario, mediante la voz. Son accesibles desde el módulo Perfiles, a través de su funcionalidad de gestión del estado del perfil, permitiendo de este modo la activación de perfil cuando el dispositivo del usuario se encuentre en la localización predefinida.

Como análisis previo a la implementación de la práctica, se ha llevado a cabo un estudio de los principales sistemas operativos móviles para definir qué sistema operativo era más atractivo y funcional para llevar a cabo el desarrollo de la aplicación.

Una vez elegido el sistema operativo Android como base de desarrollo de la aplicación, se ha procedido a realizar un estudio detallado de este mismo sistema operativo. Además, de los sistemas de diálogo y localización que son empleados en el desarrollo del proyecto.

Como sugerencias de mejora para la aplicación, podrían plantearse los siguientes aspectos:

- Incluir la posibilidad del uso de la aplicación en varios idiomas.
- Incluir una nueva funcionalidad para vehículos, pudiendo el usuario interactuar con la aplicación por medio de voz, incluyendo además la gestión de aplicaciones.
- Mejorar la interfaz gráfica.

## **4.2 Trabajos Futuros**

En este apartado, se proponen las líneas de trabajo futuro sobre el Trabajo Fin de Grado realizado. Estas líneas de trabajo se engloban en la inteligencia ambiental, destacando dos grandes tecnologías que pueden potenciar el proyecto realizado.

### **4.2.1 NFC**

Se trata de una tecnología de comunicación inalámbrica, de corto alcance y alta frecuencia que permite el intercambio de datos entre dispositivos.

Mediante esta tecnología podría implementarse desde servicios de pago seguros en ubicaciones predefinidas o implementación de mecanismos de acceso seguros a través de esta tecnología.

A su vez, también puede ser utilizada esta tecnología para la comunicación con otros dispositivos a corto alcance, desde nuestro dispositivo móvil.

### **4.2.2 Sistemas empotrados**

Se tratan de sistemas de computación diseñados para realizar una o algunas funcionalidades dedicadas, frecuentemente en un sistema a tiempo real.

Sería interesante integrar dispositivos domésticos desde un terminal móvil, con el fin de automatizar y potenciar las tareas para las cuales fueron creados.

# Presupuesto

En este apartado se expone el coste total de realizar el Trabajo Fin de Grado. En él se tiene en consideración la duración de las distintas fases y tareas, además de incluir un desglose con el coste humano, material y un resumen de los costes totales.

## 1. Fases

### ➤ Planificación

- Análisis general de los principales sistemas operativos móviles.  
🕒 Duración: 6 días.
- Estudio detallado del sistema operativo Android.  
🕒 Duración: 9 días.
- Estudio detallado de los sistemas de voz y localización.  
🕒 Duración: 11 días.
- Aprendizaje de las tecnologías necesarias.  
🕒 Duración: 26 días.
- Análisis de requisitos de la aplicación.  
🕒 Duración: 9 días.

### ➤ Desarrollo

- Análisis y diseño conceptual.  
🕒 Duración: 6 días.
- Análisis y diseño gráfico.  
🕒 Duración: 10 días.
- Implementación del sistema.  
🕒 Duración: 45 días.
- Pruebas unitarias.  
🕒 Duración: 11 días.
- Pruebas de integración y sistema.  
🕒 Duración: 23 días.

### ➤ Documentación

- Memoria del Trabajo Fin de Grado.  
🕒 Duración: 16 días.
- Preparación de la presentación.  
🕒 Duración: 7 días.

## 2. Medios

Para la realización de este proyecto se han empleado los siguientes medios hardware, software y humanos:

- **Dispositivos Hardware:**
  - Ordenador portátil: 850 €
  - Ordenador sobremesa: 550 €
  - Monitor: 100 €
  - Smartphone Android (2.3 o superior): 270 €
- **Aplicaciones Software:**
  - Entorno de desarrollo (Eclipse + plugins Android): Gratuito
  - Editor de texto (Notepad ++): Gratuito
  - Paquete Microsoft Office 2010: 160 €
  - Adobe Reader: Gratuito
  - Skype: Gratuito
- **Medios Humanos:**

En la realización de este proyecto han participado dos personas, el director de proyecto (tutor) y el desarrollador (alumno).

Las jornadas laborables dedicadas al proyecto han sido de cuatro horas de lunes a viernes y doce horas sábados y domingos.

- Coste de un ingeniero: 9,5 €/hora.

## 3. Resumen de costes

El coste de los medios humanos de la aplicación se resume en la Tabla 11.

Fases	Días	Importe
Planificación	58 días	4.408 €
Desarrollo	94 días	7.144 €
Documentación	23 días	1.748 €
<b>Subtotal</b>		<b>13.300 €</b>

Tabla 11 - Detalle de costes de recursos humanos.

El coste total del sistema se presenta en la Tabla 12:

<b>Concepto</b>	<b>Importe</b>
<b>Dispositivos Hardware</b>	1.680,00 €
<b>Aplicaciones Software</b>	160,00 €
<b>Medios Humanos</b>	13.300,00 €
<b>Subtotal</b>	15.140,00 €
<b>Impuestos (21% IVA)</b>	3.179,40 €
<b>Total</b>	<b>18.319,40 €</b>

Tabla 12 - Detalle de coste total

El presupuesto total de este proyecto asciende a la cantidad de **18.319,40 EUROS**.

Fdo. Adrián García Herías

Madrid a 1 de Julio de 2011

# Glosario

**API (Application Programming Interface):** conjunto de funcionalidades y procedimientos que provee un sistema operativo, aplicación o biblioteca, permitiendo añadir dichas funcionalidades al elemento que las implemente.

**Java:** se trata de un lenguaje de programación orientado a objetos, simplificando la codificación al realizarse por medio de módulos de código. A diferencia de otros lenguajes de programación, Java es un lenguaje interpretado, y por tanto, el tiempo de ejecución es mayor.

**Linux:** sistema operativo basado en software libre, por ello su código fuente puede ser utilizado, modificado y redistribuidos libremente bajo los términos de la GPL (Licencia Pública General de GNU).

**OpenGL:** se trata de un estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzca gráficos 2D y 3D.

**Plugin:** se trata de un componente software que añade una funcionalidad específica a una aplicación existente. Este componente se integra en la aplicación principal, interactuando a través de la librería correspondiente.

**STT (Speech To Text):** se trata de un tipo de sistema de dialogo, que permite a partir de una señal de audio, generar una palabra, oración o texto escrito, con una gramática correcta.

**SQL (Structured Query Language):** es un lenguaje declarativo de acceso a bases de datos relacionales, que permite especificar diversos tipos de operaciones en ellas. Una de sus principales funcionalidades es el manejo del algebra y calculo relacional, efectuando consultas con el fin de recuperar de forma sencilla información de la base de datos, así como hacer cambios en la misma.

**TTS (Text To Speech):** se trata de un tipo de sistema de dialogo, que permite a partir de una palabra, oración o texto generar una señal de audio correspondiente a la entrada.

**XML (eXtensible Markup Language):** es un lenguaje de marcas desarrollado por el World Wide Consortium (W3C), utilizado para almacenar datos de forma legible y definir la gramática de lenguajes específicos.

# Bibliografía

## **[ANDROID]**

Disponible:

<http://www.android.com/>

## **[ANDROIDEITY]**

Disponible:

<http://androideity.com/2011/07/04/arquitectura-de-android/>

## **[APPLE]**

Disponible:

<https://developer.apple.com/>

## **[ECLIPSE+SDK]**

Disponible:

<http://developer.android.com/sdk/index.html>

## **[DEVELOPERS]**

Disponible:

<https://developers.google.com/>

## **[GOOGLESCHOLAR]**

Disponible:

<http://scholar.google.es/>

## **[GPS]**

Disponible:

[http://www.oocities.org/es/foro\\_gps/infografia/gps5.pdf](http://www.oocities.org/es/foro_gps/infografia/gps5.pdf)

## **[GRIOL]**

Griol, David. Desarrollo y evaluación de Diferentes Metodologías para la Gestión Automática del Diálogo. Tesis Doctoral. Universidad Politécnica de Valencia, 2007.

## **[IAB]**

Disponible:

[http://www.iabspain.net/wp-content/uploads/downloads/2013/09/V\\_Estudio\\_Mobile\\_Marketing\\_version\\_corta.pdf](http://www.iabspain.net/wp-content/uploads/downloads/2013/09/V_Estudio_Mobile_Marketing_version_corta.pdf) [Mayo de 2014]

## **[LLIESTERRI]**

Llisterra, J. (2006) "Introducción a los sistemas de diálogo", in LLISTERRI, J.-MACHUCA, M. J. (Eds.) Los sistemas de diálogo. Bellaterra - Soria: Universitat Autònoma de Barcelona, Servei de Publicacions - Fundación Duques de Soria (Manuals de la Universitat Autònoma de Barcelona, Lingüística, 45), págs. 11-21.

**[ONSI]**

Disponible:

[http://www.ontsi.red.es/ontsi/sites/default/files/xlii\\_oleada\\_las\\_tic\\_en\\_los\\_hogares\\_espanol\\_es\\_4t\\_2013.pdf](http://www.ontsi.red.es/ontsi/sites/default/files/xlii_oleada_las_tic_en_los_hogares_espanol_es_4t_2013.pdf) [Mayo de 2014]