

# SISTEMA DE GUÍA POR VOZ EN INTERIORES

**Mariana Martín- Calderín de la Villa**

**FACULTAD DE INFORMÁTICA**

**DEPARTAMENTO DE INGENIERÍA DEL SOFTWARE E INTELIGENCIA ARTIFICIAL**

**UNIVERSIDAD COMPLUTENSE DE MADRID**



**TRABAJO DE FIN DE GRADO EN INGENIERIA DEL SOFTWARE**

**Madrid, junio de 2013**

Directores

Gonzalo Rubén Méndez Pozo

Pablo Gervás Gómez- Navarro



# **SISTEMA DE GUÍA POR VOZ EN INTERIORES**

Trabajo de Fin de Grado en Ingeniería del Software  
Departamento de Ingeniería del Software e Inteligencia Artificial  
Facultad de Informática  
Universidad Complutense de Madrid

## Autor

Mariana Martín-Calderín de la Villa

## Profesores Directores

Dr. Gonzalo Rubén Méndez Pozo

Dr. Pablo Gervás Gómez-Navarro

Curso 2012 / 2013

-



## **Autorización:**

Se autoriza a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, los contenidos audiovisuales incluso si incluyen imágenes de los autores, la documentación y/o el prototipo desarrollado.

---

Mariana Martín- Calderín de la Villa.



# Agradecimientos

Tras un año duro de dedicación absoluta, trabajo y esfuerzo ya está aquí, ¡por fin!, el ansiado proyecto, que no habría podido conseguir sin el apoyo y la ayuda de muchas personas a las que quiero agradecerse de todo corazón.

En primer lugar a mis directores de proyecto, Gonzalo Méndez y Pablo Gervás, por sus enseñanzas, sus ayudas con continuas reuniones, su apoyo y su orientación. Quizás haya sido el año en el que más habilidades y técnicas de trabajo he aprendido.

En segundo lugar a mis compañeros y amigos, por haber conseguido que ciertos momentos fueran más llevaderos y ofrecerme su ayuda y consejos.

En tercer lugar a mi familia, mis hermanos y mis padres por haberme soportado todo este tiempo y por el apoyo que me han proporcionado no sólo durante este año, sino desde el inicio de mis estudios universitarios, agradecer principalmente a mis padres el haber podido ofrecerme la educación que he recibido y sin la cual no habría podido llegar a conseguir todo esto.

Y por último quiero agradecer especialmente y con muchísimo cariño este trabajo a mi pareja, Roberto, por su apoyo a lo largo de todo este tiempo, por haberme aguantado esos momentos difíciles durante este curso, por su dedicación y su ayuda infinita en este proyecto, que sin ella no habría podido conseguir y alcanzar el éxito del mismo, por ser mi compañero de proyecto en la sombra y por estar siempre ahí, GRACIAS.



# Resumen

En los últimos años se ha producido un incremento del interés académico e industrial en los sistemas de localización para personas y objetos. Este crecimiento queda reflejado en la creación de aplicaciones comerciales, donde se ha recogido un aumento de la demanda en los sistemas de posicionamiento en interiores para las personas. En particular, dichos sistemas de localización en interiores, están teniendo una repercusión muy importante en fábricas, hospitales, grandes almacenes, entornos muy amplios y desconocidos en un primer momento para las personas que necesitan moverse en ellos. Unimos a este crecimiento el incremento mundial en el uso del sistema operativo Android, integrado en la mayoría de los dispositivos móviles de gran parte de la población.

Considerando todos estos datos, este proyecto de fin de carrera tiene como fin el diseño y desarrollo de un sistema de guía por voz en interiores. El sistema trata de guiar a un usuario en un escenario provisto de red Wi-Fi. En este caso, se utiliza como escenario, la primera planta de la Facultad de Informática de la Universidad Complutense de Madrid. Mediante el posicionamiento Wi-Fi, el sistema es capaz de captar la posición en la que se encuentra el usuario. Gracias al sistema de voz integrado en la aplicación, el usuario debe indicar a través de él el destino al que desea ir. Una vez queda localizado en el escenario el origen y el destino del usuario, el sistema generará una serie de instrucciones que serán transmitidas por voz al usuario e indicarán la ruta que debe seguir para alcanzar el destino solicitado.

El resultado de ese diseño y desarrollo es un sistema formado por tres aplicaciones conectadas entre sí y que en su conjunto son capaces de estimar la posición de un individuo en un entorno cerrado con cobertura Wi-Fi, calcular las rutas y generar las instrucciones necesarias para alcanzar un destino y reproducir la información en un dispositivo gestionado por un sistema operativo Android.

## Palabras clave:

- Posicionamiento en interiores
- Android
- Posicionamiento Wi-Fi
- Sistemas de guía por voz
- Reconocimiento y Reproducción de voz

# Abstract

In recent years there has been an increasing academic and industrial interest in localization systems for people and objects. This growth is reflected in the creation of business applications, where it has picked up an increase in demand in indoor positioning systems for people. In particular, such indoor location systems, are having a major impact in factories, hospitals, department stores, very large and unknown environments at first for people who need to move on them. We join this growing global increase in the use of the Android operating system, integrated in most mobile devices much of the population.

Considering all these data, this limit project aims to design and develop a voice guidance system indoors. The system tries to guide a user on a stage equipped with Wi-Fi network. In this case, it is used as a backdrop, the first floor of the Faculty of Informatics at the University Complutense of Madrid. Using Wi-Fi positioning, the system is able to capture the position where the user is located. Thanks to the voice built into the application, the user must indicate through it the destination you want to go. Once the stage is located in the origin and destination of the user, the system generates a set of instructions that will be transmitted to the user voice and indicate the path to follow to reach intended destination.

The result of that design and development is a system consisting of three interconnected applications and which together are able to estimate the position of an individual in a closed environment with Wi-Fi coverage, calculate routes and generate instructions for reach a destination and reproduce information on a device managed by an operating system Android.

## Keywords:

- Indoor Positioning
- Android
- Wi-Fi Positioning
- Voice Guidance Systems
- Voice Recognition and Reproduction

# Índice General

<b>Capítulo I Introducción</b>	<b>1</b>
1.1. Contexto .....	1
1.2. Objetivos.....	2
1.3. Estructura del documento.....	3
<b>Capítulo II Estado de la Cuestión</b>	<b>5</b>
2.1. Android.....	5
2.1.1. Introducción.....	5
2.1.2. Orígenes.....	6
2.1.3. Versiones .....	7
2.1.4. Arquitectura.....	8
2.2. Sistemas de voz .....	10
2.2.1. Introducción.....	10
2.2.2. Reconocimiento de voz .....	11
2.2.3. Reproducción de voz .....	11
2.2.4. Aplicaciones basadas en el Sistema de Voz.....	11
2.3. Sistemas de localización en interiores.....	14
2.3.1. Introducción.....	14
2.3.2. Posicionamiento GPS.....	15
2.3.3. Posicionamiento mediante Triangulación con estaciones base.....	15
2.3.4. Posicionamiento Wi-Fi.....	16
2.3.5. Posicionamiento Bluetooth .....	16
2.3.6. Ejemplos de sistemas de localización en interiores .....	17
<b>Capítulo III Trabajo Previo</b>	<b>21</b>
3.1. AVANTI: Sistema de asistencia a la evacuación de incendios. ....	21
3.2. Modelado de evacuación de multitudes mediante agentes y transcripción de comportamientos. ....	22
3.3. Generador de Instrucciones en Lenguaje Natural.....	24

## **Capítulo IV Características del Proyecto** **26**

---

4.1. Introducción .....	26
4.2. Cliente.....	28
4.2.1. Análisis de redes y posicionamiento.....	29
4.2.2. Sistema de voz .....	31
4.3. Servidor .....	32
4.3.1. Cálculo de rutas .....	32
4.3.2. Algoritmo de generación de instrucciones.....	34
4.4. Modelo de la base de datos .....	35

## **Capítulo V Gestión del Proyecto** **37**

---

5.1. Análisis de requisitos.....	37
5.1.1. Aplicación móvil.....	37
5.1.2. Servidor.....	38
5.2. Gestión de riesgos .....	39
5.2.1. Análisis de los Riesgos .....	40
5.2.2. Priorización de los Riesgos.....	41
5.2.3. Estudio de los Riesgos.....	41
5.3. Planificación del proyecto .....	45

## **Capítulo VI Diseño e Implementación** **47**

---

6.1. Introducción .....	47
6.2. Diseño e Implementación Cliente .....	47
6.3. Diseño e Implementación Servidor .....	49
6.3.1. Obtención del destino a partir de la entrada por voz .....	51
6.3.2. Cálculo de la ruta mínima .....	52
6.4. Conexión Cliente-Servidor.....	54

---

**Capítulo VII Resultados, Conclusiones y Trabajo Futuro** **55**

7.1. Descripción de los resultados obtenidos y alcanzados .....	55
7.1.1. Resultados positivos .....	56
7.1.2. Resultados negativos .....	56
7.2. Conclusiones.....	57
7.3. Trabajo futuro.....	57

---

**Apéndice A Manual de instalación del Proyecto** **59**

A.1. Requisitos .....	59
A.2. Instalación en Eclipse .....	59
A.3. Instalación en un terminal móvil.....	60
A.4. Instalación del servidor .....	60
A.5. Puesta en Funcionamiento.....	60

---

**Apéndice B Manual de usuario** **62**

B.1. Pantalla de Inicio de la Aplicación .....	62
B.2. Pantalla de Registro de la Aplicación .....	63
B.3. Menú principal de la Aplicación .....	64

---

**Bibliografía** **70**

---

# Índice de Figuras

Figura 1: Evolución de Android a lo largo del tiempo .....	7
Figura 2: Arquitectura de Android .....	8
Figura 3: Ejemplos de la Aplicación Siri.....	12
Figura 4: Ejemplo de la Aplicación Sherpa .....	13
Figura 5: Location-Based Services .....	14
Figura 6: Wi-Fi Tags Ekahau.....	17
Figura 7: Sistema AeroScout .....	18
Figura 8: Google Indoor Maps.....	20
Figura 9: Esquema de Funcionamiento a nivel de módulos .....	27
Figura 10: División de la Facultad en áreas .....	33
Figura 11: Origen de coordenadas que se ha establecido en la Facultad.....	34
Figura 12: Modelo E-R de la Base de Datos .....	36
Figura 13: Diagrama de Clases Aplicación Cliente.....	48
Figura 14: Diagrama de Clases de la Aplicación Servidor.....	50
Figura 15: Diagrama de Secuencia de la Obtención del Destino a partir de la entrada por voz .....	52
Figura 16: Diagrama de Secuencia del cálculo de la ruta mínima.....	53
Figura 17: Diagrama de Secuencia de la conexión entre aplicaciones .....	54
Figura 18: Pantalla de Inicio de la Aplicación .....	63
Figura 19: Pantalla de Registro de la Aplicación .....	64
Figura 20: Pantalla del menú principal de la Aplicación .....	65
Figura 21: Pantalla Indicar Ruta hacia el destino .....	67
Figura 22: Pantalla del Reconocimiento de Voz.....	68
Figura 23: Pantalla con la Ruta Calculada .....	69

# Índice de Tablas

Tabla 1: Evolución temporal del Sistema Operativo Android .....	7
Tabla 2: Relación Probabilidad-Severidad SQAS-SEI .....	40
Tabla 3: Nivel de Criticidad de los Riesgos en función de la Probabilidad-Severidad.....	40
Tabla 4: Priorización de los Riesgos .....	41

---

# CAPÍTULO I

## INTRODUCCIÓN

---

### 1.1. CONTEXTO

La problemática asociada a la localización en interiores es un tema que está siendo objeto de estudio e investigación en los últimos años. Hasta el momento, ninguna de las soluciones planteadas para resolver esta situación ha alcanzado el éxito que han tenido los sistemas de localización en exteriores, como puede ser el Sistema de Posicionamiento Global (*Global Positioning System* o GPS).

Conocer la posición en la que nos encontramos, es muchas veces una necesidad diaria, y la tecnología ha jugado siempre un papel muy importante en este tema. El empleo de dispositivos portátiles e integrados en los coches equipados con tecnología GPS nos ha traído una visión mucho más sencilla de la localización y, sobre todo, del sistema de guía con el que venía incorporado y que tanto éxito ha tenido en el mundo.

Actualmente nos encontramos con una nueva tecnología mucho más equipada, los Smartphones, teléfonos móviles con el GPS e Internet ya incorporados, los cuales nos ofrecen muchos más servicios en la localización que los anteriores. Estos han hecho que el resto de los dispositivos con GPS pasen a un segundo plano. A pesar de todo, esta tecnología presenta sus limitaciones, puesto que sólo tenemos cobertura GPS cuando tenemos línea directa con los satélites, es decir, esta tecnología no es válida para el posicionamiento en interiores. Debido a esto surge una clara necesidad de desarrollo de sistemas de posicionamiento o localización en interiores.

A día de hoy, las redes inalámbricas no sólo están orientadas a dar cobertura a las necesidades empresariales y particulares, sino también a las necesidades públicas, como universidades, hospitales, hoteles, metro, centros de ocio, etc. Es por ello, que se plantea como una posible solución al posicionamiento en interiores, el uso de las redes inalámbricas para captar la localización en el interior de todos estos lugares. Una solución de bajo coste, al estar basado en infraestructuras existentes y al alcance de las nuevas tecnologías.

Otro factor a tener en cuenta para el desarrollo de este proyecto, ha sido el uso de la tecnología Android, cuyo crecimiento mundial ha sido altamente notable, obteniendo gran parte de la cuota de mercado y pasando a ser uno de los grandes competidores de los sistemas operativos en el mercado tecnológico.

Crear un sistema de guía en interiores es necesario para ayudar a las personas a desenvolverse y moverse con facilidad en el interior de los grandes edificios que no conocen, como pueden ser hospitales, grandes almacenes, fábricas o aeropuertos por ejemplo. De esta forma se ofrecen facilidades al usuario y se evita estar dando vueltas innecesariamente alcanzado el destino deseado de forma sencilla y rápida.

## **1.2. OBJETIVOS**

Este proyecto se enmarca dentro del Proyecto Miles (TIN2009-14659-C03) del Departamento de Ingeniería del Software e Inteligencia Artificial de la Facultad de Informática de la UCM y tiene como objetivo integrar la funcionalidad realizada en otros proyectos anteriores, así como la implementación de nueva funcionalidad para lograr la realización final de un sistema de guía por voz en interiores.

Se desarrollará un sistema capaz de localizar personas en el interior de un edificio, utilizando como infraestructura la red inalámbrica del propio edificio. Para el desarrollo de la aplicación se utilizará Android, sistema operativo que nos permitirá el reconocimiento de voz, para que el usuario pueda indicar el lugar deseado, y la reproducción de voz, que se utilizará, una vez calculada la trayectoria, para indicar al usuario la ruta que ha de seguir.

Para el desarrollo de este proyecto se dispone del proyecto “AVANTI: Sistema de asistencia a la evacuación de incendios”, realizado por Enrique López Mañas, Francisco Javier Moreno, Javier Plá Herrero [1]. En dicho proyecto se creó un asistente para la evacuación en caso de incendio basado en posicionamiento Wi-Fi, donde el sistema a grandes rasgos combina el Posicionamiento Wi-Fi y la Realidad Aumentada.

En el presente proyecto, se realiza una adaptación del proyecto anterior a las nuevas y últimas versiones de Android, heredando la aplicación y lo referente al posicionamiento en el interior del edificio, basado en posicionamiento Wi-Fi. Se continúa utilizando como escenario la primera planta de la Facultad de Informática de la Universidad Complutense de Madrid. Se implementarán,

además, nuevas funcionalidades en este nuevo proyecto: la aplicación dispondrá de un sistema de guía por voz, tanto de reconocimiento de voz como de reproducción de voz. Además, para obtener las rutas y generar las instrucciones, se realizarán y utilizarán algoritmos de cálculo de trayectoria así como algoritmos de generación de instrucciones ubicados en un servidor.

La idea general es que un usuario desea ir a un lugar en concreto. El sistema podrá calcular la situación de este en el edificio. El usuario, mediante el sistema de voz, indicará el lugar al que desea ir. A continuación y gracias a los algoritmos de generación de instrucciones y cálculos de trayectoria, el sistema indicará, por voz, el camino mínimo que debe seguir para llegar a su destino.

### 1.3. ESTRUCTURA DEL DOCUMENTO

Este documento se encuentra organizado en diferentes capítulos. Este primer capítulo contiene una introducción y objetivos del proyecto a realizar.

El segundo capítulo, el **Estado de la Cuestión**, recoge en primer lugar una visión general y evolución del sistema operativo Android, seguido de un estudio de la situación actual y aplicaciones que podemos encontrar de los sistemas por voz, y por último la situación actual de los sistemas de localización en interiores, así como las diferentes compañías que trabajan en proyectos referentes a ello.

Tras describir el entorno en que se desarrolla este proyecto, se pasa al tercer capítulo, **Trabajos Relacionados**, que contiene un breve resumen de los proyectos de los que se han podido reutilizar parte de su funcionalidad para lograr la funcionalidad de este proyecto.

El cuarto capítulo, **Características del Proyecto**, muestra una visión del proyecto en sí, se describe la arquitectura modular y funcional del proyecto, entrando en detalle en cada uno de sus componentes.

El quinto capítulo, **Gestión del Proyecto**, define el proceso que se ha llevado a cabo para desarrollar el proyecto, formalizada según las distintas propuestas de *Pressman* [2] para metodologías de desarrollo de software.

El sexto capítulo, **Diseño e Implementación**, explica más detalladamente las partes de la aplicación para lograr la funcionalidad del sistema.

El séptimo capítulo, **Resultados y Trabajos Futuros**, evalúa los resultados obtenidos y propone una línea de trabajos futuros para este proyecto.

A continuación se encuentran los Apéndices. El, **Apéndice A**, recoge un manual de instalación de la aplicación en los distintos entornos así como la puesta en funcionamiento del mismo. El **Apéndice B**, recoge un manual de usuario.

Por último, se encuentra la **Bibliografía** donde se recogen las referencias utilizadas para la realización de este proyecto.

---

# CAPÍTULO II

## ESTADO DE LA CUESTIÓN

---

### 2.1. ANDROID

#### 2.1.1. Introducción

Android es una plataforma de desarrollo libre basada en Linux y de código abierto. Una de sus grandes ventajas es que se puede usar y adaptar a las preferencias del usuario o propietario sin pagar ningún tipo de licencia.

Es adaptable a cualquier tipo de Hardware. Android no ha sido diseñado exclusivamente para su uso en teléfonos y tabletas. Hoy en día podemos encontrar relojes, cámaras, electrodomésticos y gran variedad de sistemas empujados que se basan en este sistema operativo.

Android tiene una portabilidad asegurada, ya que las aplicaciones finales son desarrolladas en Java, lo que nos asegura que podrán ser ejecutadas en cualquier tipo de CPU, tanto presente como futuro. Esto se consigue gracias al concepto de máquina virtual.

Posee una gran cantidad de servicios incorporados: localización basada en GPS, bases de datos con SQL, reconocimiento y síntesis de voz, multimedia, navegador, etc.

Está optimizado para consumo de baja potencia y poca memoria. Se refleja esto, por ejemplo, en que Android utiliza la *Máquina Virtual Dalvik*. Se trata de una implementación de Google de la máquina virtual de Java optimizada tanto para los dispositivos móviles como para su correcto funcionamiento sobre el kernel de Linux.

Tiene una alta calidad de gráficos y sonido, como son los gráficos vectoriales suavizados, animaciones inspiradas en Flash, gráficos en tres dimensiones basados en OpenGL. Además incorpora *codecs* estándar más comunes de audio y vídeo, incluyendo H.264 (AVC), MP3, AAC, etc.

En conclusión Android nos ofrece una forma sencilla y novedosa de implementar potentes aplicaciones para diferentes tipos de dispositivo. Y es por esto y por todas sus características por lo que interesa Android como sistema operativo para desarrollar el proyecto. Cubre con todas las necesidades y requisitos que se necesitan para llevar adelante este proyecto.

### 2.1.2. Orígenes

Android fue desarrollado inicialmente por la empresa *Android Inc.* Fundada en octubre de 2003. El objetivo principal de la empresa era desarrollar dispositivos móviles inteligentes que supieran la localización del usuario y sus preferencias. En agosto de 2005, Google compra la empresa Android Inc. Y continúa con el desarrollo de la plataforma Android. Este mismo año empiezan a trabajar en la máquina virtual optimizada para los dispositivos móviles (*Dalvik VM*).

En el año 2007 se crea el consorcio *Handset Alliance*, con el objetivo de desarrollar estándares abiertos para móviles. Está formado por Google, Intel, Texas Instruments, Motorola, T-Mobile, Samsung, Ericsson, Toshiba, Vodafone, NTT DOCOMO, Sprint Nextel y otros. Una pieza clave de los objetivos de esta alianza es promover el diseño y difusión de la plataforma Android. Sus miembros se comprometieron a publicar una parte importante de su propiedad intelectual como código abierto bajo licencia Apache v2.0.

En noviembre del 2007 se lanza una primera versión del Android SDK. Al año siguiente aparece el primer móvil con Android (*T-Mobile G1*). En octubre Google libera el código fuente de Android principalmente bajo licencia de código abierto Apache (licencia GPL v2 para el núcleo). Ese mismo mes se abre *Android Market*, para la descarga de aplicaciones. En abril del 2009 Google lanza la versión 1.5 del SDK que incorpora nuevas características como el teclado en pantalla. A finales del 2009 se lanza la versión 2.0 y durante el 2010 las versiones 2.1, 2.2 y 2.3.

Durante el año 2010 Android se consolida como uno de los sistemas operativos para móviles más utilizados, con resultados cercanos al iPhone e incluso superando al sistema de Apple en EE.UU.

En el 2011 se lanzan la versión 3.0, 3.1 y 3.2 específica para tabletas y la 4.0 tanto para móviles como para tabletas. Durante este año Android se consolida como la plataforma para móviles más importante alcanzando una cuota de mercado superior al 50%.

En 2012 Google cambia su estrategia en su tienda de descargas online, reemplazando *Android Market* por *Google Play Store*. Donde en un solo portal unifica tanto la descarga de aplicaciones como de contenidos. En este año aparecen las versiones 4.1 y 4.2 del SDK. Android mantiene su espectacular crecimiento, alcanzando a finales de año una cuota de mercado del 75%.

### 2.1.3. Versiones

Versión	Nivel de API	Nombre	Fecha
Android 1.0	1	-	Septiembre de 2008
Android 1.1	2	-	Febrero de 2009
Android 1.5	3	Cupcake	Abril de 2009
Android 1.6	4	Donut	Septiembre de 2009
Android 2.0	5	Éclair	Octubre de 2009
Android 2.1	7	Éclair	Enero de 2010
Android 2.2	8	Froyo	Mayo de 2010
Android 2.3	9	Gingerbread	Diciembre de 2010
Android 3.0	11	Honeycomb	Febrero de 2011
Android 3.1	12	Honeycomb	Mayo de 2011
Android 3.2	13	Honeycomb	Julio de 2011
Android 4.0	14	Ice Cream Sandwich	Octubre de 2011
Android 4.0.3	15	Ice Cream Sandwich	Diciembre de 2011
Android 4.1	16	Jelly Bean	Julio de 2012
Android 4.2	17	Jelly Bean	Noviembre de 2012

Tabla 1: Evolución temporal del Sistema Operativo Android



Figura 1: Evolución de Android a lo largo del tiempo

## 2.1.4. Arquitectura

En el siguiente gráfico se muestra la arquitectura de Android. Como se puede ver está formada por cuatro capas. Una de las características más importantes es que todas las capas están basadas en *software* libre.

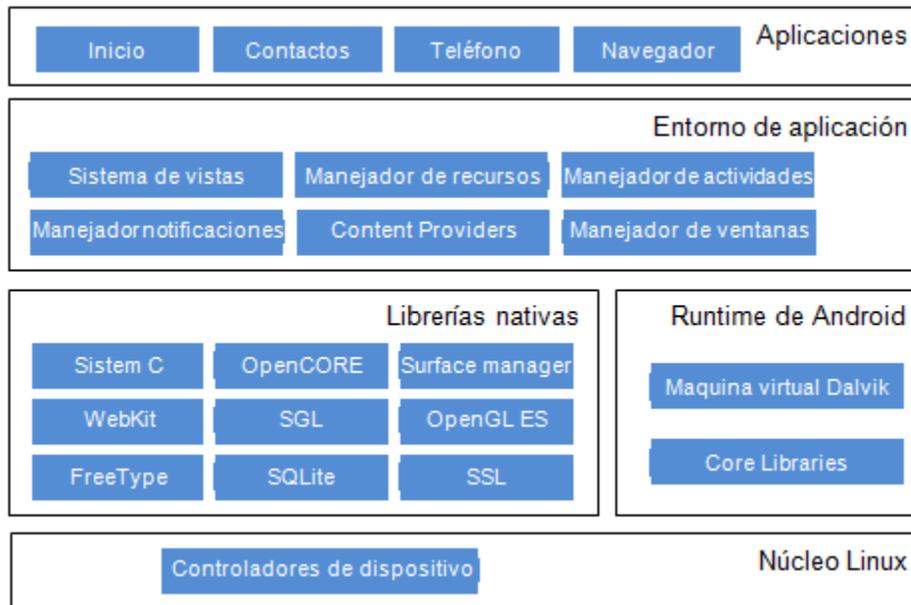


Figura 2: Arquitectura de Android

### 2.1.4.1. El núcleo Linux

El núcleo de Android está formado por el sistema operativo Linux versión 2.6. Esta capa proporciona servicios como la seguridad, el manejo de la memoria, el multiproceso, la pila de protocolos y el soporte de *drivers* para dispositivos.

Esta capa del modelo actúa como capa de abstracción entre el hardware y el resto de la pila. Por lo tanto, es la única que es dependiente del *hardware*.

### 2.1.4.2. Runtime de Android

Está basado en el concepto de máquina virtual utilizado en Java. Dado las limitaciones de los dispositivos donde ha de correr Android, poca memoria y procesador limitado, no fue posible utilizar una máquina virtual Java estándar. Google tomó la decisión de crear una nueva, la máquina virtual Dalvik, que respondiera mejor a estas limitaciones.

Algunas características de la máquina virtual Dalvik que facilitan esta optimización de recursos son: la ejecución de ficheros Dalvik ejecutables (.dex), es un formato optimizado para ahorrar memoria. Además, está basada en registros. Cada aplicación corre en su propio proceso Linux con su propia instancia de la máquina virtual Dalvik. Delega al kernel de Linux algunas funciones como *threading* y el manejo de la memoria a bajo nivel.

También se incluye en el Runtime de Android el “*core libraries*” con la mayoría de las librerías disponibles en el lenguaje Java.

### 2.1.4.3. Librerías nativas

Incluye un conjunto de librerías en C/C++ usadas en varios componentes de Android. Están compiladas en código nativo del procesador. Muchas de las librerías utilizan proyectos de código abierto. Algunas de estas librerías son:

- **System C library:** una derivación de la librería BSD de C estándar (*libc*), adaptada para dispositivos embebidos basados en Linux.
- **Media Framework:** librería basada en *PacketVideo's OpenCORE*; soporta *codecs* de reproducción y grabación de multitud de formatos de audio vídeo e imágenes MPEG4, H.264, MP3, AAC, AMR, JPG y PNG.
- **Surface Manager:** maneja el acceso al subsistema de representación gráfica en 2D y 3D.
- **WebKit:** soporta un moderno navegador web utilizado en el navegador Android y en la vista *webview*. Se trata de la misma librería que utiliza Google Chrome y Safari de Apple.
- **SGL:** motor de gráficos 2D.
- **Librerías 3D:** implementación basada en OpenGL ES 1.0 API. Las librerías utilizan el acelerador hardware 3D si está disponible, o el software altamente optimizado de proyección 3D.
- **FreeType:** fuentes en bitmap y renderizado vectorial.
- **SQLite:** potente y ligero motor de bases de datos relacionales disponible para todas las aplicaciones.
- **SSL:** proporciona servicios de encriptación *Secure Socket Layer*.

### 2.1.4.4. Entorno de aplicación

Proporciona una plataforma de desarrollo libre para aplicaciones con gran riqueza e innovaciones (sensores, localización, servicios, barra de notificaciones, etc.).

Esta capa ha sido diseñada para simplificar la reutilización de componentes. Las aplicaciones pueden publicar sus capacidades y otras pueden hacer uso de ellas (sujetas a las restricciones de seguridad). Este mismo mecanismo permite a los usuarios reemplazar componentes.

Una de las mayores fortalezas del entorno de aplicación de Android es que se aprovecha el lenguaje de programación Java. El SDK de Android no acaba de ofrecer todo lo disponible para su estándar del entorno de ejecución Java (JRE), pero es compatible con una fracción muy significativa de la misma.

Los servicios más importantes que incluye son:

- **Views:** extenso conjunto de vistas, son la parte visual de los componentes.
- **Resource Manager:** proporciona acceso a recursos que no son en código.
- **Activity Manager:** maneja el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellas.
- **Notification Manager:** permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.
- **Content Providers:** mecanismo sencillo para acceder a datos de otras aplicaciones, como por ejemplo, los contactos.

### 2.1.4.5. Aplicaciones

Este nivel está formado por el conjunto de aplicaciones instaladas en una máquina Android. Todas las aplicaciones han de correr en la máquina virtual Dalvik para garantizar la seguridad del sistema.

Normalmente las aplicaciones Android están escritas en Java. Para desarrollar aplicaciones en Java podemos utilizar el Android SDK. Existe otra opción consistente en desarrollar las aplicaciones utilizando C/C++. Para esta opción podemos utilizar el Android NDK (Native Development Kit).

## 2.2. SISTEMAS DE VOZ

### 2.2.1. Introducción

Los comandos o acciones por voz son parte de las herramientas incluidas dentro del sistema de Android. Este tipo de herramientas fueron diseñadas en un principio para ampliar el uso de los dispositivos portátiles y móviles a gente con alguna discapacidad visual o motora. En la actualidad, con el crecimiento de la demanda en los dispositivos móviles y sumando la búsqueda continua de una comodidad en el uso de estos, se opta por hacer una integración completa de los comandos por voz a los nuevos sistemas operativos, como ha sido en el caso de Android.

Los comandos por voz han pasado a ser una de las herramientas más utilizadas dentro del universo Android ayudando a personas como conductores u otro tipo de personas que estén continuamente con las manos ocupadas. Esta funcionalidad facilita en gran medida la utilización de numerosas aplicaciones, como pueden ser por ejemplo la escritura de mensajes de texto, mails y búsquedas. Pero no sólo se limita al uso de este tipo de aplicaciones, sino también al manejo total del equipo, pudiendo manipular acciones dentro del mismo, como son el contestar o colgar las llamadas, la búsqueda de contactos, navegación por los menús, etc.

### **2.2.2. Reconocimiento de voz**

El objetivo principal de los sistemas de reconocimiento de voz es desarrollar interfaces centradas en las necesidades del usuario aprovechando una de las capacidades que tiene el hombre para comunicarse, la expresión oral.

Las aplicaciones basadas en este tipo de reconocedores son, por ejemplo, servicios financieros, asistencia de directorio, transferencia de llamadas telefónicas, consultas de información. La ventaja que presenta este tipo de aplicaciones es que al interactuar, el usuario utiliza la eficiencia del habla y está libre de movimientos de las manos en caso de que las tenga ocupadas.

Existen otras aplicaciones que no se basan en el teléfono, por ejemplo, el dictado automático. También, el reconocimiento de voz es usado en compañías en donde la entrada de datos o comandos por voz es requerida tales como en desarrollo de inventarios, control de robots, etc.

### **2.2.3. Reproducción de voz**

El habla es una herramienta básica de comunicación entre los seres humanos y podría considerarse como el medio natural mediante el que interaccionamos entre nosotros. El desarrollo de aplicaciones informáticas capaces de comprender y sintetizar el habla supone un paso hacia adelante en el proceso de facilitar la interacción del hombre-máquina.

El realizar aplicaciones capaces de reproducir el contenido de las mismas, ayuda principalmente a personas con alguna discapacidad visual o motora, tal y como se ha mencionado anteriormente. Además de facilitar el uso y manejo de la aplicación a cualquier tipo de usuario.

### **2.2.4. Aplicaciones basadas en el Sistema de Voz**

En los últimos años han surgido aplicaciones que utilizan procesamiento del lenguaje y permiten a sus usuarios, mediante expresión oral, hacer prácticamente todas las iteraciones que harían con su dispositivo móvil de forma táctil.

Actualmente, las dos que tiene mayor relevancia dentro del mercado son Siri, en iOS, y Sherpa, para Android.

### 2.2.4.1. Siri

Siri es una aplicación asistente personal para iOS. Fue creada en 2007 y en apenas dos años ya había recuperado financiación por valor de 24 millones de dólares. Fue adquirida por Apple Inc. el 28 de Abril de 2010<sup>1</sup> e incluida por primera vez dentro de sus dispositivos iPhone 4S<sup>2</sup>.

La aplicación utiliza procesamiento del lenguaje natural para responder preguntas o hacer recomendaciones. Siri requiere de conexión a Internet, puesto que las instrucciones son procesadas en los servidores de Apple y no directamente por los dispositivos.

La aplicación para iOS es el primer producto enfocado en una aplicación de inteligencia virtual, ya que Siri se adapta a las preferencias individuales del usuario con el paso del tiempo y personaliza los resultados.

A continuación se muestra una imagen dónde se puede ver el uso de la aplicación en diferentes entornos y situaciones:



Figura 3: Ejemplos de la Aplicación Siri

<sup>1</sup> <http://scobleizer.com/2010/04/28/breaking-news-siri-bought-by-apple/>

<sup>2</sup> <http://www.telegraph.co.uk/technology/apple/8804922/Apple-iPhone-event-live.html>

### 2.2.4.2. Sherpa

Sherpa es un asistente de voz que nos permite interactuar con nuestro terminal Android desde la versión 2.2. La aplicación fue lanzada a finales de 2012 como el equivalente a Siri en dispositivos Android y en apenas año y medio de vida ya ha alcanzado 500.000 descargas. Actualmente las versiones para Windows 8 y BlackBerry están en desarrollo.

Sherpa al igual que Siri aprende de los hábitos de los usuarios. La mayoría de las ocasiones, Sherpa muestra la información que el usuario ha solicitado en su propia interfaz, pero si el sistema comprendiese la petición pero no dispusiese de la información relevante disponible en su propio sistema, se muestra una página de resultados de Google.

A continuación se muestra una imagen de la Aplicación Sherpa, que recoge distintas actividades que se pueden realizar con ella:



Figura 4: Ejemplo de la Aplicación Sherpa

## 2.3. SISTEMAS DE LOCALIZACIÓN EN INTERIORES

### 2.3.1. Introducción

La localización, el conocer en qué lugar estamos, ha sido siempre una necesidad para desenvolvernó día a día. De una forma abstracta, podemos definir también la localización como la determinación de relaciones espaciales entre objetos.

El GPS es la solución “casi universal” para obtener localización precisa y rápida en cualquier punto del planeta. Sin embargo, este sistema no funciona en algunos entornos exteriores y en la mayoría de los interiores. Es por ello que surge un nuevo concepto: *Local Positioning System (LPS)*, son sistemas de localización alternativos creados para funcionar en entornos locales. De las varias posibilidades tecnológicas para el diseño de LPS, las basadas en señales RF experimentan un gran auge en la actualidad.

Este gran auge se debe a que las tecnologías inalámbricas han experimentado una fuerte eclosión en el mercado tecnológico desde hace unos años. Queda reflejado este crecimiento en que estas tecnologías son usadas en la vida cotidiana por toda la sociedad como:

- Redes personales (*Personal Area Networks, PAN*). Son el Bluetooth, ratones y teclados inalámbricos.
- Redes locales (*Local Area Networks, LAN*). Conocida como Wi-Fi.
- Integración y convergencia con los sistemas celulares (“globales”), como GPRS y móviles de 4ª generación (All IP).

Gracias al desarrollo y estandarización de estas nuevas tecnologías, están surgiendo nuevas aplicaciones tecnológicas como la conectividad continua (voz, audio vídeos digitales), redes de sensores inalámbricos, sistemas de rescate y emergencias y servicios basados en la posición.

Los LPS dan soporte a los servicios basados en la posición (*Location-Based Services, LBS*).

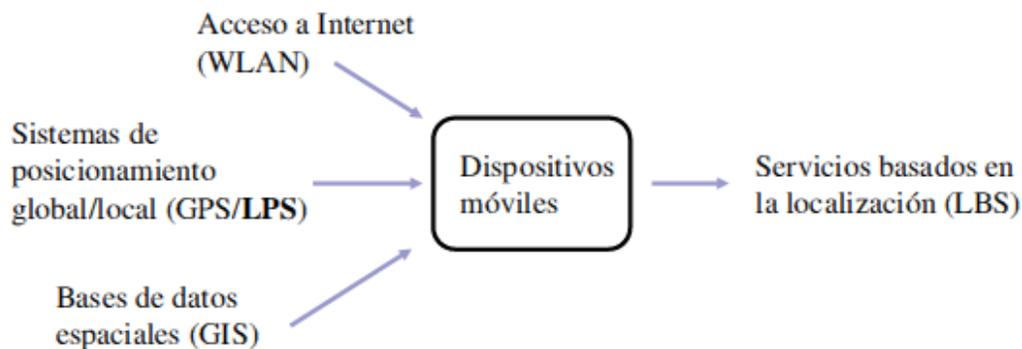


Figura 5: Location-Based Services

La localización en interiores tiene un futuro prometedor en el ámbito de la *Ubiquitous Computing*, que es una disciplina en la que se pretende que una persona use en su vida cotidiana múltiples dispositivos de computación, pero no tiene necesidad de darse cuenta de que lo está haciendo.

A continuación se muestran distintos tipos de sistemas de localización.

### 2.3.2. Posicionamiento GPS

GPS<sup>3</sup> (*Global Positioning System*), es un sistema de navegación por satélite que proporciona información precisa sobre la localización de un terminal, y está preparado para otra serie de usos como información meteorológica.

Este tipo de posicionamiento puede ser usado en cualquier lugar donde haya línea directa y sin obstáculos hacia por lo menos cuatro satélites de la red GPS. Esta tecnología está mantenida por el gobierno de los Estados Unidos, y es accesible de manera gratuita a cualquier individuo que disponga de un receptor GPS.

El GPS fue originalmente desarrollado por el Departamento Americano de Defensa, e inicialmente basaba su funcionamiento en la señal recibida de 24 satélites. Se empezó a utilizar en 1973 para hacer frente a todos los problemas de navegación de la época.

Algunos países permiten el uso de repetidores GPS para permitir la recepción de señales GPS en interiores, o en localizaciones donde la cobertura no llega como se desearía. Como es el ejemplo y podemos ver en la última actualización de *Google Maps* para móviles, han incluido en Japón y en EE.UU. varios mapas de centros comerciales y aeropuertos por los que Google te da la opción de guiarte hasta tu puerta de embarque, tu tienda de interés o simplemente la salida.

No obstante, las leyes de la Unión Europea y del Reino Unido prohíben explícitamente el uso de estos dispositivos, ya que las señales pueden causar interferencias en otros dispositivos.

Viendo los obstáculos que planteaba este tipo de tecnología a la hora de desarrollar el posicionamiento para este proyecto, fue descartada a favor de otra tecnología.

### 2.3.3. Posicionamiento mediante Triangulación con estaciones base

Este tipo de triangulación utiliza las estaciones base de telefonía GSM para proporcionar una ubicación aproximada con 200-1000 metros de precisión.

Generalmente se utiliza como un complemento a los sistemas de posicionamiento Wi-Fi (WPS) o al GPS allí donde la cobertura de estos últimos no es lo suficientemente potente.

---

<sup>3</sup> <http://www.gps.gov/>

La compañía *Skyhook*<sup>4</sup>, responsable de una API para usar posicionamiento por WPS, mantiene una base de datos mundial con información sobre estaciones base, lo que incrementa la cobertura de áreas usando XPS (sistemas híbridos de posicionamiento) y ayuda a mejorar el tiempo de adquisición de datos cuando se usa GPS.

Esta triangulación también puede ayudar en la optimización de recursos energéticos, activando o desactivando los distintos sistemas que participan el XPS dependiendo de las condiciones ambientales y de los sistemas disponibles.

Dado el margen de error de este tipo de triangulación, más cercano a los 1000 metros que a los 200, se descartó también esta tecnología de posicionamiento.

### 2.3.4. Posicionamiento Wi-Fi

El posicionamiento basado en Wi-Fi (*WPS, Wi-Fi Positioning System*) surge como una idea que pueda resolver los problemas anteriormente citados, aprovechando el rápido crecimiento de los puntos de acceso en áreas urbanas. *SkyHook Wireless* es la compañía puntera en este tipo de servicio, manteniendo una base de datos pública a la que se puede acceder a través de una API, y obtener el posicionamiento basado en los puntos de acceso accesibles desde un terminal. *SkyHook* proporciona su sistema a los dispositivos basados en *Apple*, aunque ha habido otras iniciativas similares, como la del *Fraunhofer Institute* [3] o Google<sup>5</sup>. La técnica de localización utilizada en posicionamiento con puntos de acceso inalámbricos se basa en medir la intensidad de la señal recibida (*Received Signal Strength, RSS*) y el método de “*fingerprinting*” [4] [5].

La precisión depende del número de posiciones que se hayan introducido en la base de datos, las posibles fluctuaciones de la señal que puedan ocurrir pueden llegar a introducir errores y faltas de precisión en la trayectoria del usuario. Para minimizar las fluctuaciones de la señal recibida, se aplican ciertas técnicas para filtrar ruidos. Hay nuevas leyes y regulaciones que se están imponiendo a operadores de VO- IP para que integren algún sistema en sus puntos de acceso que permita determinar la posición de los terminales en un entorno dado<sup>6</sup>.

### 2.3.5. Posicionamiento Bluetooth

Bluetooth es la especificación que define un estándar de comunicación para la transmisión de voz y de datos entre diferentes dispositivos mediante un enlace por radiofrecuencia omnidireccional. Está diseñado especialmente para dispositivos de bajo consumo con un rango de cobertura bajo.

---

<sup>4</sup> <http://www.skyhookwireless.com>

<sup>5</sup> [http://www.google.com/googleblogs/pdfs/google\\_submission\\_dpas\\_wifi\\_collection.pdf](http://www.google.com/googleblogs/pdfs/google_submission_dpas_wifi_collection.pdf)

<sup>6</sup> Abril 2005, <http://www.telecomweb.com/news/1112721769.htm>

Las principales aplicaciones de Bluetooth hoy en día son la transferencia de archivos, la sincronización de dispositivos y conectividad de periféricos. Además, existen diversas soluciones de sistemas de posicionamiento en interiores que emplean la tecnología Bluetooth. Soluciones propuestas en los últimos años como son:

El trabajo de *Feldmann et al* [6] que dispone de tres ordenadores portátiles que actúan como emisores de señal Bluetooth y un dispositivo móvil del tipo PDA HP iPAQ con Windows Mobile como receptor de la señal Bluetooth. El dispositivo móvil es quién calcula la posición dónde se encuentra mediante triangulación de la señal que emiten los tres portátiles.

Para poder realizar un posicionamiento basado en Bluetooth en edificios públicos habría que realizar una instalación previa de balizas que emitan este tipo de señal, lo que implicaría un gasto adicional, ya que la mayor parte de los edificios no suelen incorporarlas mientras que sí que lo hacen con las señales Wi-Fi.

## 2.3.6. Ejemplos de sistemas de localización en interiores

### 2.3.6.1. Ekahau

La empresa Ekahau, Inc. ha desarrollado un sistema llamado *Ekahau Positioning Engine*<sup>7</sup>. Es el primer sistema comercial cuyo método de localización es íntegramente software, al usar para este fin la RSS, *Received Signal Strength* (Potencia señal recibida), de los puntos de acceso de interés.

Las personas a localizar llevan unas etiquetas Wi-Fi (*Wi-Fi Tags*) que son las que reciben las potencias de señal de los puntos de acceso de interés.



Figura 6: Wi-Fi Tags Ekahau

<sup>7</sup> <http://www.ekahau.com/products/real-time-location-system/positioning-engine.html>

La forma de calibración es totalmente experimental, usando muestras de la RSS en cada localización de interés, las cuales se toman dividiendo la planta en una cuadrícula en lugar de dividirla por localizaciones de interés. Esto tiene la ventaja de que los errores de estimación del sistema se podrán caracterizar de forma absoluta. La precisión de la localización es, según la página web del fabricante, de 1 metro.

### 2.3.6.2. AeroScout

El sistema *AeroScout* es un sistema de localización tanto para interiores como para exteriores basado en Wi-Fi y en RFID, que usa las técnicas DToA y RSS para la localización, y por eso es uno de los sistemas de localización más avanzados tecnológicamente.

El funcionamiento se puede resumir en que las etiquetas RFID de *AeroScout* o los terminales móviles que quieren localizar emiten periódicamente una señal.

A continuación, los puntos de acceso o receptores de localización *AeroScout* reciben la señal emitida y la envían al motor de procesamiento de *AeroScout*. Luego, el motor de procesamiento usa métodos de tiempo de llegada o de potencia de señal recibida para determinar la situación del terminal móvil o de la etiqueta RFID emisora. Por último, se indica la posición de manera gráfica para que el usuario la conozca.

A continuación se muestra una imagen con los distintos dispositivos de los que consta la tecnología *AeroScout*.

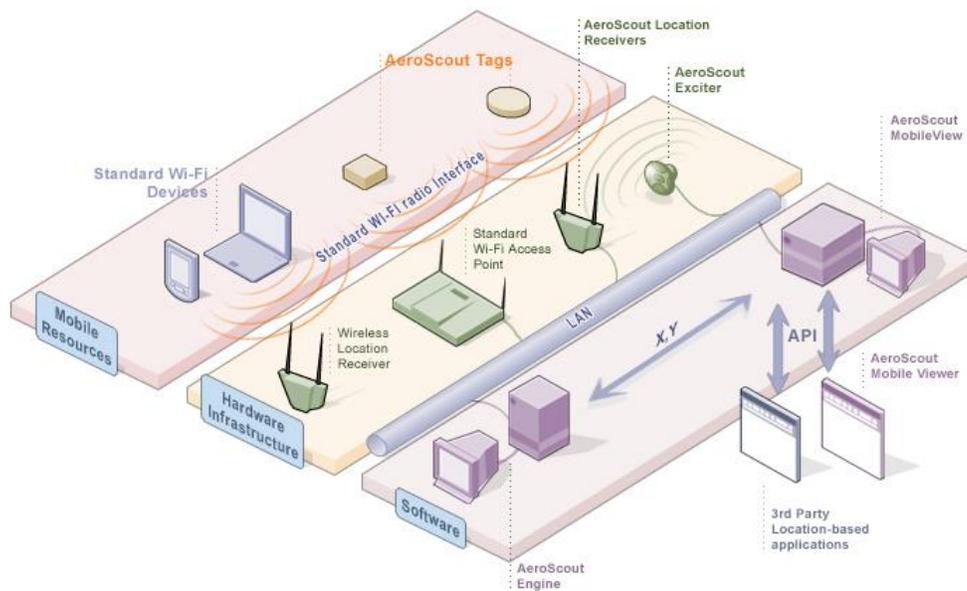


Figura 7: Sistema AeroScout

### 2.3.6.3. Telefónica

Es un proyecto desarrollado dentro del programa de I+D de Telefónica, se basa en la localización Wi-Fi en base a mediciones de RSS [7].

El método utilizado es el algoritmo Monte Carlo, parte de la potencia recibida, pero teniendo en cuenta otros factores como la información que proviene de la arquitectura del edificio y perfil del usuario. Se utiliza, además, como mecanismo de filtrado para mejorar la precisión. El algoritmo Monte Carlo se implementa con el algoritmo recursivo SIR (*Sampling Importance Resampling*), lo que permite tener en cuenta tanto datos estáticos como dinámicos, relativos al movimiento del usuario.

El sistema estaba inicialmente pensado para entornos centralizados, pero permite la localización distribuida de modo que cada terminal móvil se localice a sí mismo. En el caso más usual, los ordenadores de los usuarios transmiten la información de las señales recibidas al ordenador central, que se encarga de ejecutar el algoritmo de localización. En el segundo caso, la posición la estima el propio terminal.

### 2.3.6.4. Google Indoor Maps, versión 6.0

La nueva versión 6.0 de este software es ahora capaz de realizar el geo-posicionamiento en interiores e incluso guiar al usuario bajo techo<sup>8</sup>.

Esta nueva función sólo está disponible en algunos grandes centros comerciales y aeropuertos internacionales de Estados Unidos y Japón. En dichas ubicaciones, la aplicación puede localizar al usuario y guiarlo hasta un punto concreto (como una puerta de embarque) usando funciones que ya estaban disponibles para la navegación en exteriores, como el modo brújula o la perspectiva tridimensional.

La aplicación únicamente está capaz de cubrir el posicionamiento de determinados lugares, ya que es necesario disponer del mapa del interior del edificio que ha de ser almacenado en la aplicación previamente para poder realizar el posicionamiento.

Una vez está disponible el mapa de edificio para poder realizar dicha localización, la aplicación sitúa al usuario cuando está viendo el mapa, actualizando la posición acorde al movimiento del usuario y haciendo distinción entre las distintas plantas del edificio.

Para realizar este posicionamiento se utiliza la señal de red Wi-Fi de la misma forma que realizan el posicionamiento GPS para Google Maps, captando la intensidad de la señal. Como ocurre en este proyecto el posicionamiento para este tipo de señal no es muy preciso conteniendo un margen de error.

---

<sup>8</sup> <http://googleblog.blogspot.com.es/2011/11/new-frontier-for-google-maps-mapping.html>

En la siguiente imagen podemos ver el posicionamiento antes y después de activar el selector de piso en el que te encuentras.

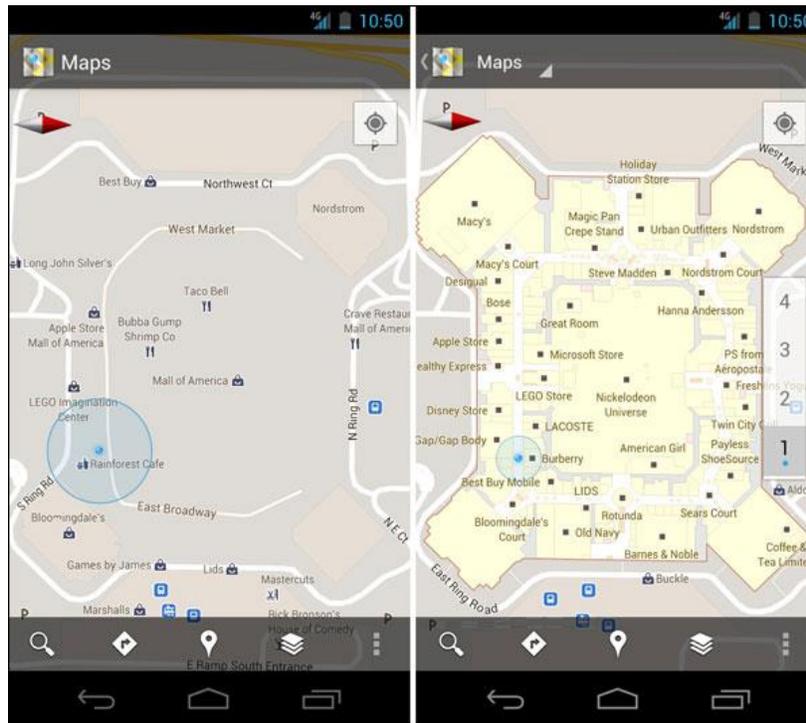


Figura 8: Google Indoor Maps

---

# CAPÍTULO III

## TRABAJO PREVIO

---

Para la realización de este trabajo se han tenido en cuenta varios proyectos englobados en el Proyecto MILES (TIN2009-14659-C03) del departamento de Ingeniería del Software e Inteligencia Artificial de la Facultad de Informática de la UCM, de los que se ha conseguido adaptar parte de su funcionalidad a la nueva funcionalidad de este proyecto.

A continuación se enumera y se exponen brevemente los proyectos de los que se ha conseguido heredar parte de su funcionalidad.

### **3.1. AVANTI: SISTEMA DE ASISTENCIA A LA EVACUACIÓN DE INCENDIOS.**

Proyecto de Sistemas Informáticos realizado por Enrique López Mañas, Francisco Javier Moreno, Javier Plá Herrero [1].

El objetivo de dicho proyecto era la creación de un sistema de evacuación de incendios. El proyecto trata de aportar una plataforma intuitiva pero completa y potente para usar durante los ensayos de evacuación. Se combina la tecnología de posicionamiento Wi-Fi, predicción de movimiento gracias a los sensores que proporcionan los terminales Android y Realidad Aumentada, para poder contribuir a una experiencia más visual y estimuladora. El proyecto intenta contribuir a que los ensayos de incendio puedan ser informatizados. De esta forma, conseguir que estudiantes e instructores puedan analizar fácilmente fallos en los protocolos a seguir, y que llegaba la situación de un incendio los ensayos puedan cumplir su objetivo: ser eficaces y evitar que ningún miembro de la Facultad de Informática sufra daños personales.

El proyecto proporcionó una solución que combinase las ventajas de las simulaciones virtuales con las ventajas de las evacuaciones reales. El sistema que se propuso utilizó técnicas Wireless para posicionamiento en interiores, y realidad aumentada para poder sobre-imprimir información en el entorno, mientras se puede observar la realidad a través de la pantalla del dispositivo móvil. El acelerómetro del teléfono se usa para recolectar información que permita estimar el movimiento del usuario, basado en estadísticas y mediciones previas, en aquellas áreas que no disponen de una señal certera, conocidas como áreas de sombra.

AVANTI se desarrolló con el objetivo de poder estudiar la combinación de tecnologías pertenecientes a la Inteligencia Ambiental con otras como la Realidad Aumentada y poder determinar que conflictos pueden surgir de combinar estas dos líneas. Empleando todas estas tecnologías se pretendió conseguir unos efectos de motivación sobre usuarios de la plataforma, que se tomen más en serio los ensayos para la evacuación de incendios para que puedan mejorar las habilidades sobre la evacuación de edificios, y tengan presentes todas y cada una de las reglas en el momento que haga ponerlas en marcha.

Parte de la funcionalidad del proyecto AVANTI ha sido reutilizada para la elaboración del actual proyecto, como ha sido la parte del posicionamiento en interiores utilizando la tecnología Wi-Fi. Dicha funcionalidad ha interesado en este proyecto para conseguir ubicar al usuario en una posición actual y controlar en todo momento la ubicación el usuario en el edificio. De esta forma dado el origen, que es la posición actual del usuario, y establecido el destino por el usuario, podemos generar la secuencia de instrucciones necesaria para que el usuario alcance el destino deseado. Gracias a este posicionamiento se consigue controlar también, que el usuario sigue el camino correcto y no se sale de la ruta establecida.

## **3.2. MODELADO DE EVACUACIÓN DE MULTITUDES MEDIANTE AGENTES Y TRANSCRIPCIÓN DE COMPORTAMIENTOS.**

Trabajo de Fin de Máster realizado por Elena Núñez González [8].

Este proyecto tuvo como fin la creación de un sistema multi-agente que simula el comportamiento de las personas que se ven involucradas en una evacuación de incendios de un edificio. La función principal de este proyecto fue la creación de distintos tipos de comportamiento en los agentes que actuaban de diferentes maneras en la evacuación del edificio. Se simularon tres tipos de comportamiento:

- El líder, que ayuda a los demás a salir del escenario.
- El dependiente, que no tiene capacidad para salir por su propia cuenta y que necesita que un líder le ayude.
- El independiente, con capacidad para evacuar por sí mismo, pero en ocasiones decide poner su vida en peligro saltándose las normas de evacuación.

La simulación del proyecto produce un extenso log con los eventos sucedidos. Se creó un sistema, *FromLog2StoryTelling*, que trata el log, lo resume y modifica para generar un informe con los sucesos más importantes ordenados, y preparado para traducirse a lenguaje natural.

El simulador consiste en un sistema multi-agente en el que cada individuo es un agente, con características propias, las cuales, junto al entorno, determinan su comportamiento. Se tuvieron en cuenta diversos factores que distinguieron un agente de otro, se hizo una distinción entre el rol, evacuador y evacuado, la independencia, que distingue a dependientes de independientes, y el liderazgo, que distingue a líderes de seguidores.

Se propuso la creación de tres tipos de agentes. Los primeros, que representan a la mayoría de los implicados en la evacuación, son agentes simples, con comportamientos similares entre ellos, y que cumplen normas de evacuación. El segundo y tercer tipo de agentes son mucho más complejos, y toman decisiones más complicadas. El segundo corresponde a los líderes, responsables de la evacuación. El tercer grupo lo formarían los independientes, con comportamientos complejos que pueden llegar a desobedecer explícitamente las normas de evacuación.

Cada acción individual de cada personaje de la simulación da lugar a una entrada de un log en que se explica sus movimientos o reacciones. Además, el fuego también genera un texto sobre sus propiedades y la evolución que tomaba el incendio en la simulación. Toda esta información genera unos registros con un elevado número de datos.

Todos estos datos son posteriormente tratados y se generan unos resúmenes con la información relevante sobre los acontecimientos ocurridos durante la simulación.

La parte que ha interesado de este proyecto para la elaboración del sistema de guía ha sido el diseño para realizar las trayectorias que seguían los agentes en la evacuación del edificio. Realizado gracias al algoritmo de *Dijkstra*, algoritmo de caminos mínimos, que servía para calcular las rutas mínimas de los agentes desde la posición en la que se iban encontrando hasta la salida más cercana teniendo en cuenta no encontrar el fuego de por medio.

### 3.3. GENERADOR DE INSTRUCCIONES EN LENGUAJE NATURAL

Hasta la fecha, aquellas aplicaciones en que ha resultado necesaria la generación de instrucciones o descripciones verbales han recurrido por lo general a soluciones basadas en textos enlatados (el sistema muestra mensajes cableados directamente en el código del programa), o basadas en plantillas (fragmentos de texto con los elementos variables identificados – nombres, verbos - para poder remplazarlos al reutilizarlos en contextos distintos). Estas soluciones resultaban apropiadas porque su eficiencia y su simplicidad encajaban bien con las restricciones prioritarias del contexto de operación.

Los recientes avances en potencia de cómputo y capacidad de almacenamiento, así como en tecnologías de GLN, permiten replantear estas decisiones. Hoy en día, el uso de soluciones basadas en conocimiento (tanto lingüístico como del mundo en forma de ontología o jerarquía conceptual) ya no presenta problemas de eficiencia, y la flexibilidad que proporciona permite soluciones más ágiles y naturales. El uso de tecnologías basadas en conocimiento permite optimizar el tamaño de los mensajes generados, garantizando la identificación correcta de los referentes y la fluidez natural de los mensajes.

El problema fundamental de los generadores tradicionales es que su construcción es compleja y está muy ligada al dominio de aplicación. Los sistemas de GLN desarrollados presentaban distintas arquitecturas, aunque todas tenían en común una organización modular de las tareas: determinación de contenido - qué se va a decir -, planificación del documento - cómo se va a organizar -, generación de expresiones de referencia - cómo describir cada objeto -, agregación - cómo agrupar lo que se dice -, lexicalización - qué término concreto usar en cada caso -, y realización superficial - cómo dar forma lingüística al resultado.

El desarrollo de soluciones software reutilizables para la etapa de la generación de lenguaje natural correspondiente a la realización superficial (por ejemplo FUF, KPLM, YAG o RealPro) disparó el número de sistemas de GLN desarrollados para aplicaciones reales o de investigación. Esta expansión ha llevado incluso a la creación de empresas que ofrecen comercialmente servicios de descripción textual automática de contenidos digitales no textuales (como CoGenTex).

Para el resto de las tareas de GLN se han explorado soluciones en sistemas de investigación, y empieza a aparecer un consenso sobre la manera en que deben llevarse a cabo, pero no se han desarrollado todavía módulos reutilizables como los que existen para la realización superficial. El trabajo de investigación en este campo en la actualidad estudia la manera en que la ingeniería del software podría facilitar el desarrollo de este tipo de módulos.

Como resultado de la investigación del grupo de investigación NIL en el proyecto GALANTE (TIN2006-14433-C02-01), se ha desarrollado el framework TAP, diseñado para facilitar el desarrollo de aplicaciones de generación automática de lenguaje natural de forma reutilizable. Para ello se combinó trabajo previo sobre arquitecturas genéricas para el desarrollo de

aplicaciones de generación de lenguaje natural [9] y [10], con ideas de patrones de diseño [11] y *frameworks* [12].

El generador de lenguaje natural TAP (*Text Arranging Pipeline*) proporciona funcionalidades de generación de expresiones de referencia, elección sintáctica, y lexicalización, lo que permite conseguir una especificación del discurso a realizar a partir de una entrada definida en términos de conceptos abstractos. La realización superficial de ese plan de discurso en forma de texto gramaticalmente válido se lleva a cabo utilizando SurReal [13], un realizador superficial desarrollado en Java que utiliza los mecanismos básicos de FUF [14] sobre una gramática para el castellano definida según la notación de SURGE [15]. Esta solución de generación de lenguaje natural ya se ha utilizado en tareas de valoración competitivas [16] [17] y [18]. Para el problema específico de generación de instrucciones para guiar a los largo de trayectorias espaciales, se han desarrollado soluciones utilizando TAP en el contexto del proyecto NOVA (Navegación basada en Ontologías mediante la Verbalización de mensajes de Ayuda, TIN2009-14659-C03-01).

---

# CAPÍTULO IV

---

## CARACTERÍSTICAS DEL PROYECTO

---

### 4.1. INTRODUCCIÓN

En este apartado se van a describir con más detalle todas las características del proyecto. En primer lugar, explicar que la funcionalidad de este proyecto se ha dividido en dos partes principales que se comunican entre sí para lograr el objetivo deseado: la generación de unas instrucciones que se puedan transmitir al usuario para conseguir llegar al destino solicitado. Dichas partes se organizan con una arquitectura cliente-servidor.

La aplicación móvil, gracias a la cual el usuario interactúa con la aplicación, constituye el cliente. Esta aplicación permite al usuario establecer el destino al que desea ir. Una vez introducido, transmitirá por voz la secuencia de instrucciones generadas.

El servidor, además del almacenamiento de datos de la aplicación cliente contiene dos aplicaciones, que son las encargadas del cálculo de ruta y la generación de instrucciones que posteriormente serán transmitidas al usuario.

En este capítulo se explicará que funcionalidad es característica del cliente y que funcionalidad es característica del servidor así como la comunicación que se ejecuta durante las acciones típicas.

La parte de posicionamiento bajo redes Wi-Fi es reutilizada del proyecto “AVANTI: Sistema de asistencia a la evacuación de incendios.” [1], donde la solución propuesta es una implementación basada en el algoritmo de los k-vecinos más cercanos y en un filtro de partículas. De esta forma se obtienen resultados más favorables para el posicionamiento en el interior del edificio.

El servidor recoge la información de los puntos de acceso, el identificador y la potencia de la señal, que se han introducido por el usuario según se va desplazando con el terminal, en el caso de este proyecto, por la primera planta del edificio de la Facultad de Informática.

Al iniciar sesión en la aplicación, el terminal recibe del servidor la base datos con toda la información sobre localizaciones y realiza el cálculo de datos del posicionamiento en el propio terminal.

El terminal sigue en contacto con el servidor para recibir la trayectoria calculada y que el usuario debe seguir para llegar al destino. Dado que la potencia de cálculo en los dispositivos móviles es mucho menor, se ha optado por la utilización de un servidor centralizado, dispuesto de mayor potencia, para realizar el cálculo de dichas trayectorias. De esta forma se obtiene mayor eficiencia en los resultados y se evitan bloqueos en la ejecución de la aplicación o la ralentización de la misma. Se ha querido establecer una comunicación cliente-servidor, donde el servidor realiza el cálculo de los datos y el dispositivo se encarga únicamente de reproducir la información.

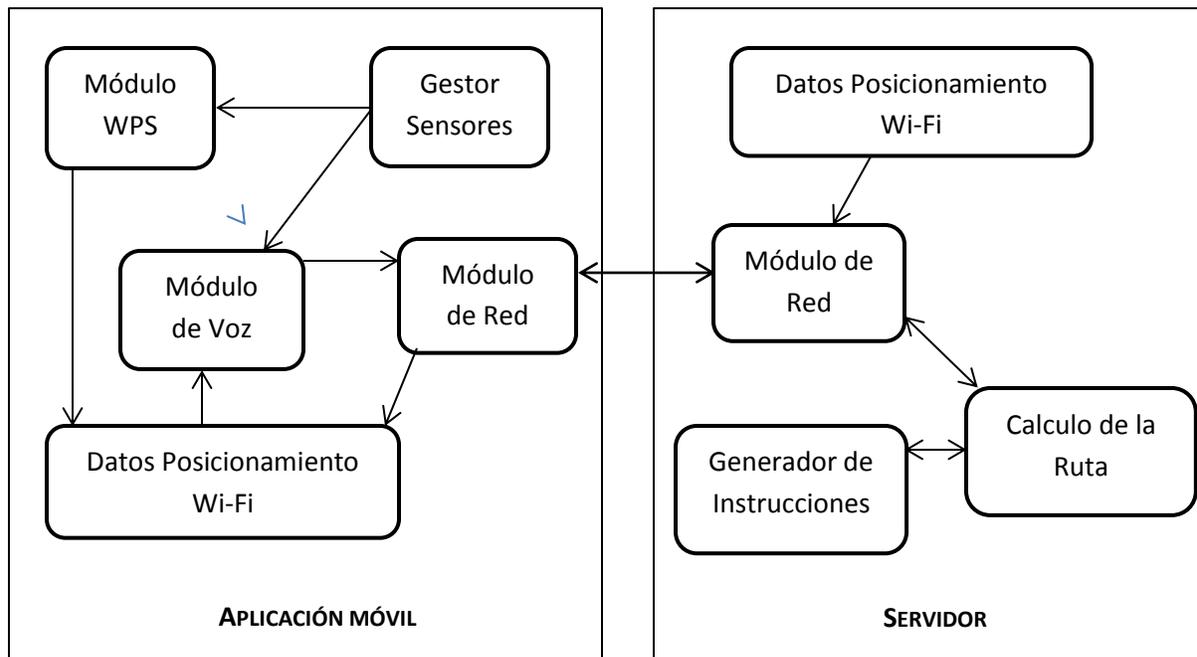


Figura 9: Esquema de Funcionamiento a nivel de módulos

La **Figura 9** explica cómo funciona la aplicación desde un punto de vista más modular. La comunicación entre la Aplicación móvil y el Servidor se realiza a través del módulo de red, dónde se realiza el intercambio de datos necesarios para poder realizar el cálculo de ruta correctamente y que ésta llegue al terminal para poder ser transmitida al usuario. Inicialmente, tal y como se ha explicado anteriormente, los datos para el posicionamiento Wi-Fi se cargan en el dispositivo móvil. El módulo WPS, en base a la entrada que recibe de los puntos de acceso que están a la vista y los datos almacenados, genera unas coordenadas geográficas que se corresponden con la posición actual del usuario. El módulo de Voz, es el encargado de la recopilación de todos los datos

iniciales, en los que se incluye solicitar por Voz, el destino al usuario. A través del módulo de red, se envían estos datos a la aplicación que se encuentra ejecutando de manera permanente en el servidor, dicha aplicación es la encargada del cálculo de la ruta mínima entre la posición actual y la posición destino solicitada por el usuario. La aplicación en el servidor, genera una serie de instrucciones que son devueltas al terminal y transmitidas al usuario, el cual debe seguirlas para llegar al destino que ha solicitado. Se ha contemplado el caso en el que el usuario se salga de la ruta indicada, debiendo generar la nueva ruta recalculada. Tras realizar pruebas, se ha llegado a la conclusión de que la mejor opción era tomar la posición actual en la que se encontraba el usuario cada diez segundos, debiendo ésta de coincidir con alguna posición contenida en la ruta indicada. En caso contrario, se procedía al cálculo de la nueva ruta.

## 4.2. CLIENTE

El desarrollo del cliente inicialmente consistió en una adaptación de proyecto “AVANTI: Sistema de asistencia a la evacuación de incendios” [1], a las nuevas versiones de Android. Dicho proyecto estaba implementado para utilizar la versión 1.5 este sistema operativo. La adaptación consistió en conseguir que la aplicación funcionara para todas las versiones de Android existentes actualmente en el mercado.

Por lo tanto para el desarrollo del cliente se ha continuado con la tecnología Google Android<sup>9</sup>, adaptándolo a las versiones más recientes existentes y ofrecidas en el mercado. Las ventajas de continuar con este sistema operativo, es que se trata de una aplicación totalmente abierta soportada por una tecnología emergente que pretende hacerse fuerte en el mercado de la tecnología portátil. Por ello, se puede afirmar que lo que se busca con este proyecto es realizar una aplicación fiable para un gran número de usuarios.

Una vez realizada esta adaptación, como ya se explicó en esta memoria en el capítulo 3, Trabajos Relacionados, punto 3.1, toda la funcionalidad de esta aplicación no es necesaria para la realización de este proyecto, por lo que se procedió a hacer una selección de la parte necesaria, el posicionamiento en interiores mediante red Wi-Fi.

Posteriormente, se añadió la nueva funcionalidad necesaria en este proyecto, la incorporación del sistema de voz a la aplicación para lograr interactuar con el usuario por voz y la comunicación entre las distintas aplicaciones que constituyen este proyecto para lograr el intercambio de datos entre el cliente y el servidor.

A continuación se describen los elementos principales, pertenecientes al cliente, que se han utilizado para el desarrollo de este proyecto.

---

<sup>9</sup> <http://developer.android.com>

### 4.2.1. Análisis de redes y posicionamiento

La parte de la localización en el interior del edificio, fue estudio del proyecto mencionado en el punto anterior [1].

Se usaron como referencia los trabajos [19] y [20]. El primero propone una solución basada en el algoritmo de los k-vecinos más cercanos, mientras que el segundo propone una solución basada en un filtro de partículas. Para una solución más favorable se realizó una implementación de ambos sistemas.

Se ha diseñado también un sencillo sistema de control para evitar posicionar al usuario en áreas que carecen de lógica, como puede ser el interior de paredes o puertas. Se ha utilizado un mapa de bits que es una representación del mapa de bits de la facultad en blanco y negro, siendo los píxeles negros las “zonas prohibidas”. De manera que si en algún momento los datos del posicionamiento nos indican que el usuario se encuentra atravesando alguna de estas zonas prohibidas, automáticamente se recoloca al usuario en la zona válida más próxima a la posición actual.

#### 4.2.1.1. K-Closest Neighbors

Este algoritmo recupera las K posiciones de la base de datos que más concuerdan con la señal de los puntos de acceso que se pueden ver desde la posición del terminal.

El criterio usado para seleccionar las mejores posiciones es la distancia euclídea (medida como la diferencia en la intensidad de la señal). Si  $Z = (RSS_1, \dots, RSS_M)$  el vector de señales observadas desde el dispositivo, compuesto por M puntos de acceso en una posición desconocida  $X = (x, y)$  (aquellas del terminal del cual deseamos obtenerla) y  $Z_i$  son las señales registradas en la base de datos para la posición  $X_i = (x_i, y_i)$ , entonces la distancia euclídea es:

$$d(Z, Z_i) = \frac{1}{M} \times \sqrt{\sum_{j=1}^M (RSS_j(x, y) - RSS_j(x_i, y_i))^2}, \quad \text{siendo:}$$

1.  $RSS_j(x_i, y_i)$ , la fuerza de la señal recibida para el punto de acceso con la MAC  $j$ , localizado en  $x_i, y_i$ .
2. M, el número de diferentes puntos de acceso que se encuentran guardados en la base de datos.

El conjunto  $N_k$  de posiciones de la base de datos, que contiene el menor rango de error  $m$  en términos de RSS se construye según el siguiente proceso iterativo:

$$N_k = \left\{ \operatorname{argmin}_{x_i \in \Omega} [d(Z, Z_i)] \mid x_i \notin N_k \right\}$$

Esto recupera las K posiciones de la base de datos con el menor error, y sin ningún tipo de repetición (recupera un conjunto único).  $\mathcal{Z}$  es el conjunto de posiciones registradas dentro de la base de datos. Finalmente la posición del dispositivo se calculara como el baricentro de las K posiciones:

$$X = \frac{\sum_{j=1}^k (1/d(Z, Z_i)) \cdot X_j}{\sum_{j=1}^k (1/d(Z, Z_i))}$$

La principal ventaja de este algoritmo es la sencillez de implementación y uso. No obstante, la precisión depende de la granularidad de la base de datos de referencia. Cuanto mayor se desee que sea la precisión, mayor debería ser la base de datos, lo que puede dar lugar a un tiempo de acceso mayor para localizar el dispositivo, y eventualmente conducir hacia una latencia inaceptable.

La selección del tamaño para el conjunto N puede variar según el número de puntos que estemos usando para calcular la posición del terminal, pudiendo modificarse para intentar alcanzar algún valor óptimo.

#### 4.2.1.2. Filtro de partículas

Para la implementación del filtro de partículas se siguió la descripción y metodologías explicadas en [7]. Acorde a ello, se establece la definición de “filtro de partículas”. La idea principal de este algoritmo es combinar el modelo de movimiento y el mapa de información en el filtro para obtener un modelo de aproximación más realístico a la posición del dispositivo.

El filtro de partículas, basado en un conjunto de ejemplos medidos aleatoriamente, es decir las partículas, representa la función de densidad de la posición del dispositivo. Cada partícula explora el entorno de acuerdo al modelo de movimiento y al mapa de información. Sus pesos son actualizados cada vez que se recibe una nueva medida. El filtro de partículas intenta estimar la distribución probabilística  $Pr [x_k | Z_{0:k}]$ , donde  $x_k$  es el vector de estado del dispositivo en el instante  $k$ , y  $Z_{0:k}$  es el conjunto de medidas recolectadas hasta la medida  $(k+1)$ . Cuando el número de partículas es lo suficientemente alto puede ser equiparado a:

$$Pr [x_k | Z_{0:k}] = \sum_{i=1}^{N_s} w_k^i \delta(x_k - x_k^i)$$

El filtro está compuesto de 5 pasos: predicción, corrección, actualización de la partícula, muestreo y estimación.

1. Predicción. Se predice, mediante el uso del acelerómetro, que rumbo tomará el usuario una vez haya iniciado el desplazamiento.

2. Corrección. En esta etapa se corrigen los errores de cálculo que hayan podido ocurrir durante el cálculo de la posición mediante una estimación, para evitar errores con un desvío importante del valor real.
3. Actualización de la partícula. Se actualiza el peso de la partícula.
4. Re-muestreo. Analiza las partículas existentes.
5. Estimación. Se estima la nueva posición del dispositivo.

Estos cinco pasos quedan explicados de manera más detallada en el proyecto mencionado [1].

## 4.2.2. Sistema de voz

El sistema de generación de voz tiene dos características principales, una de ellas es la reproducción por parte del dispositivo de las instrucciones que debe seguir el usuario para llegar al destino deseado y la otra es la posibilidad de dar comandos por voz a la aplicación para solicitar el destino deseado.

### 4.2.2.1. Reproducción de voz

Para reproducir la información por voz, se emplea la tecnología *Text-To-Speech*, que ofrece el sistema operativo Android desde su versión 2.6<sup>10</sup>. Su funcionamiento es muy sencillo, tan sólo requiere la inicialización de una instancia de *TextToSpeech*, comprobar que se cuenta con el motor *TextToSpeech*, especificar el idioma y proveer a la instancia de *TextToSpeech* inicializada de una cadena de texto para que este la lea en voz alta.

### 4.2.2.2. Reconocimiento de voz

El sistema operativo Android ofrece la posibilidad de recibir instrucciones por voz y transformarlas a texto<sup>11</sup>. Su funcionamiento consiste en lanzar una intención (*Intent*) al sistema, para emplear el motor de reconocimiento de voz, comprobando que se cuenta con este. Una vez el usuario ha hablado, se devuelve en una estructura de datos el correspondiente texto equivalente a lo dicho por el usuario.

---

<sup>10</sup> <http://android-developers.blogspot.com/2009/09/introduction-to-text-to-speech-in.html>

<sup>11</sup> [http://developer.android.com/reference/android/speech/RecognizerIntent.html#ACTION\\_RECOGNIZE\\_SPEECH](http://developer.android.com/reference/android/speech/RecognizerIntent.html#ACTION_RECOGNIZE_SPEECH)

### 4.3. SERVIDOR

El servidor se encuentra corriendo de manera permanente en una máquina. Para mayor eficiencia y práctica del proyecto, toda la funcionalidad referente a todo lo específico del edificio queda en el servidor, localizado en el propio edificio en el que se realiza el posicionamiento, de tal manera que se ha establecido una comunicación cliente-servidor continua.

Una de las funciones del servidor es el almacenamiento de datos, tanto de los usuarios registrados en el sistema como de las redes Wi-Fi que se toman para poder realizar el posicionamiento.

El cálculo de rutas es una de las funciones principales de este proyecto corriendo de manera permanente en el servidor. El algoritmo que realiza este cálculo de la ruta mínima se encuentra siempre en permanente escucha, a la espera de los datos con los que calcular las rutas mínimas entre el origen y el destino. Este algoritmo genera una serie de instrucciones, en datos numéricos, que posteriormente deben ser convertidas a un lenguaje natural.

Esta función de generar las instrucciones a lenguaje natural, es otra de las funciones principales de este proyecto. Calculadas las instrucciones numéricas, el algoritmo anterior debe conectar con este nuevo algoritmo encargado de transformar todas estas coordenadas a un lenguaje que el usuario sea capaz de entender.

#### 4.3.1. Cálculo de rutas

Para realizar el cálculo de ruta ha de plantearse primero la forma de obtener la ruta mínima que el usuario debe seguir para llegar al destino, dicha ruta mínima se calcula mediante el Algoritmo de *Dijkstra* (algoritmo de caminos mínimos). Para lograr realizar el cálculo de esta ruta mínima se necesita una representación del edificio en el que nos encontramos, para obtener un punto inicial y un punto final gracias a los cuales conseguimos realizar el cálculo de dicha ruta.

En este caso, las pruebas se han realizado en la primera planta de la Facultad de Informática de la UCM, en la que se realizó una división de la primera planta en distintos cuadrantes conectados entre sí. Cada uno de estos cuadrantes engloba varias posiciones, lo que nos permite situar al usuario en un cuadrante y calcular la distancia mínima entre el cuadrante inicial en el que se encuentra situado y el cuadrante final que recoge la posición destino del usuario. La principal ventaja de este modelo es la sencillez de implementación y de cálculo de las rutas.

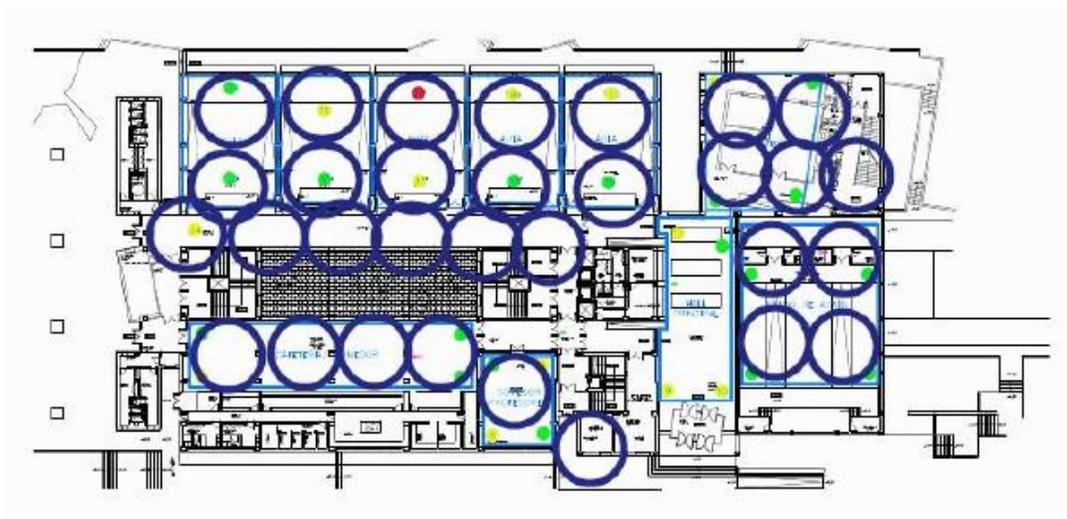


Figura 10: División de la Facultad en áreas

Establecida la representación para el cálculo de rutas y el algoritmo para buscar la ruta más corta entre el origen y el destino, el Algoritmo de *Dijkstra*, se procesa a detallar la implementación del mismo.

Para aplicar este algoritmo, se ha representado la primera planta en forma de grafo, para ello se ha realizado una división en cuadrantes de los pasillos de la primera planta. También se ha planteado una división en zonas para la Sala de Juntas y la Sala de Grados, situadas en esta primera planta.

Internamente, cada cuadrante está definido por las posiciones superior izquierda e inferior derecha del mismo y un identificador único para cada uno de ellos, así como un indicador del tipo de cuadrante según la localización en el mapa. El conjunto de cuadrantes resultante de las divisiones de la primera planta forma el conjunto de nodos del grafo.

Para almacenar las aristas de dicho grafos, se utiliza una matriz de adyacencia, que guarda las conexiones entre los distintos cuadrantes con coste unitario. Se utiliza también una lista de aristas, cada una de ellas definidas por su origen, destino y dirección. Cada arista tiene ocho posibles direcciones: arriba, arriba-derecha, derecha, abajo-derecha, abajo, abajo-izquierda, izquierda y arriba-izquierda. El uso de direcciones para las aristas facilita el cálculo de la ruta que el usuario debe seguir. Se establecen diferentes tramos establecidos según las diferentes direcciones de aristas que pueda contener la ruta generada.

El cálculo de posibles rutas se realiza en el servidor. Ahí es donde se inicializa la lista de cuadrantes, para ello, se han medido sobre el plano de la Facultad la longitud y anchura de cada uno de los pasillos, la Sala de Juntas y la Sala de Grados. De esta forma, se ha determinado sobre dicho plano la posición y dimensiones de cada uno de los cuadrantes. Una vez establecida la lista de cuadrantes, se han conectado mediante aristas y se ha completado adecuadamente la matriz de adyacencia.

Las coordenadas de cada uno de los cuadrantes se han tomado en base al origen establecido. Se puede apreciar en la siguiente imagen:

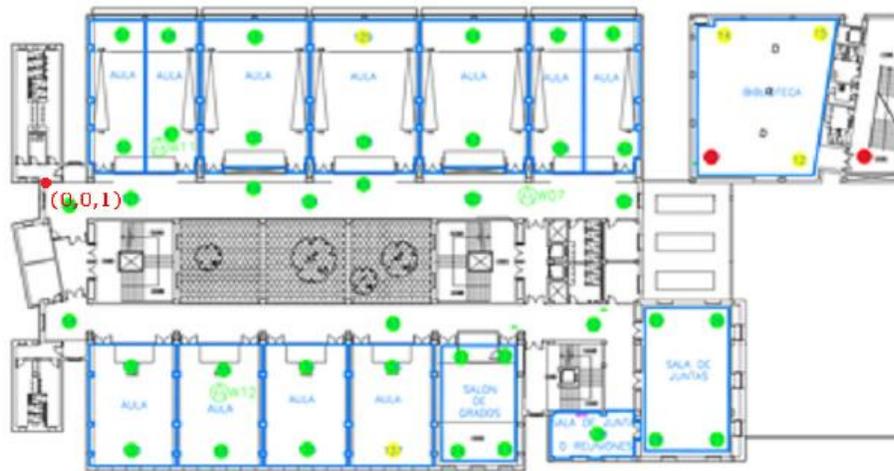


Figura 11: Origen de coordenadas que se ha establecido en la Facultad

Obtenida la posición inicial del usuario, mediante el posicionamiento Wi-Fi, y establecido el destino por parte del usuario, se mandan los datos a la aplicación del servidor “CalculaRuta”, encargada de calcular esta ruta mínima entre el origen y el destino establecidos.

### 4.3.2. Algoritmo de generación de instrucciones

El algoritmo de generación de instrucciones, es el encargado de transformar la ruta calculada numéricamente a una secuencia de instrucciones en lenguaje natural.

Una vez calculada la ruta, gracias a la aplicación anterior, queda almacenada como una lista de instrucciones, donde para cada instrucción se han almacenado los siguientes datos: la dirección que se debe tomar, diferenciando los siguientes valores: norte, noreste, este, sureste, sur, suroeste, oeste, noroeste, así como la distancia, expresada en metros, a recorrer en dicha dirección.

Inicialmente debemos obtener la orientación inicial del usuario, como ya se ha explicado, para elaborar la ruta en el sentido correcto en el que se encuentra situado el usuario.

Una vez se han obtenido todos los valores necesarios para generar la instrucción, la orientación inicial del usuario y la lista de instrucciones, se pasa a una solución de generación de instrucciones para guiar a lo largo de trayectorias espaciales basada en TAP, descrita en el capítulo 3 de esta memoria en el apartado 3.3.

Dicha llamada genera un String, que contiene toda la secuencia de instrucciones necesarias para transmitir al usuario.

EL generador de instrucciones inicialmente debe convertir los datos al formato necesario, diferenciando los distintos tramos de la ruta que se le ha pasado como parámetro de entrada. En el primer tramo, genera la instrucción teniendo en cuenta la orientación inicial del usuario. En los tramos posteriores dicha orientación se va obteniendo según la dirección que tenga que tomar.

En cada instrucción se indicará el giro que debe hacer, pudiendo expresar dicho resultado tanto en horas como en grados, así como la distancia que debe recorrer expresada en metros. Una vez se han generado todas las instrucciones para los distintos tramos, se devuelve la cadena de caracteres a la aplicación que calcula la ruta, que a su vez será la encargada de enviar dicha cadena de instrucciones a la aplicación móvil para poder ser transmitida al usuario.

#### 4.4. MODELO DE LA BASE DE DATOS

La base de datos de este proyecto está realizada en MySQL<sup>12</sup>, por ser el gestor de bases de datos más utilizado para desarrollos de código libre. Existen otras alternativas, como Oracle<sup>13</sup>, que no se ha utilizado principalmente por ser un producto de pago y disponer de presupuesto para pagar la licencia. En cualquier caso, dada la naturaleza de este proyecto y el controlado número de transacciones que se van a efectuar, no es necesario una alternativa más potente que MySQL.

A continuación, se presenta la base de datos del sistema formada por las siguientes tablas:

- Tabla **users**, que almacena la información personal del usuario al registrarse, su nombre y apellidos, el nombre de usuario y la contraseña, así como la última vez que se accedió al sistema y la última IP referente la posicionamiento que estuvo la última vez.
- Tabla **position**, guarda la información de las coordenadas en la que está el usuario, así como el usuario al que hace referencia y el tiempo en el que se produce.
- Tabla **repository**, almacena la información de todas las direcciones MAC correspondientes al mapeo del edificio. Se guarda la dirección MAC, así como la intensidad con la que se recibe y las coordenadas en la que se ha tomado.

En la siguiente imagen se presenta el Diagrama Entidad-Relación aplicado a la base de datos del sistema:

---

<sup>12</sup> <http://www.mysql.com/>

<sup>13</sup> <http://www.oracle.com>

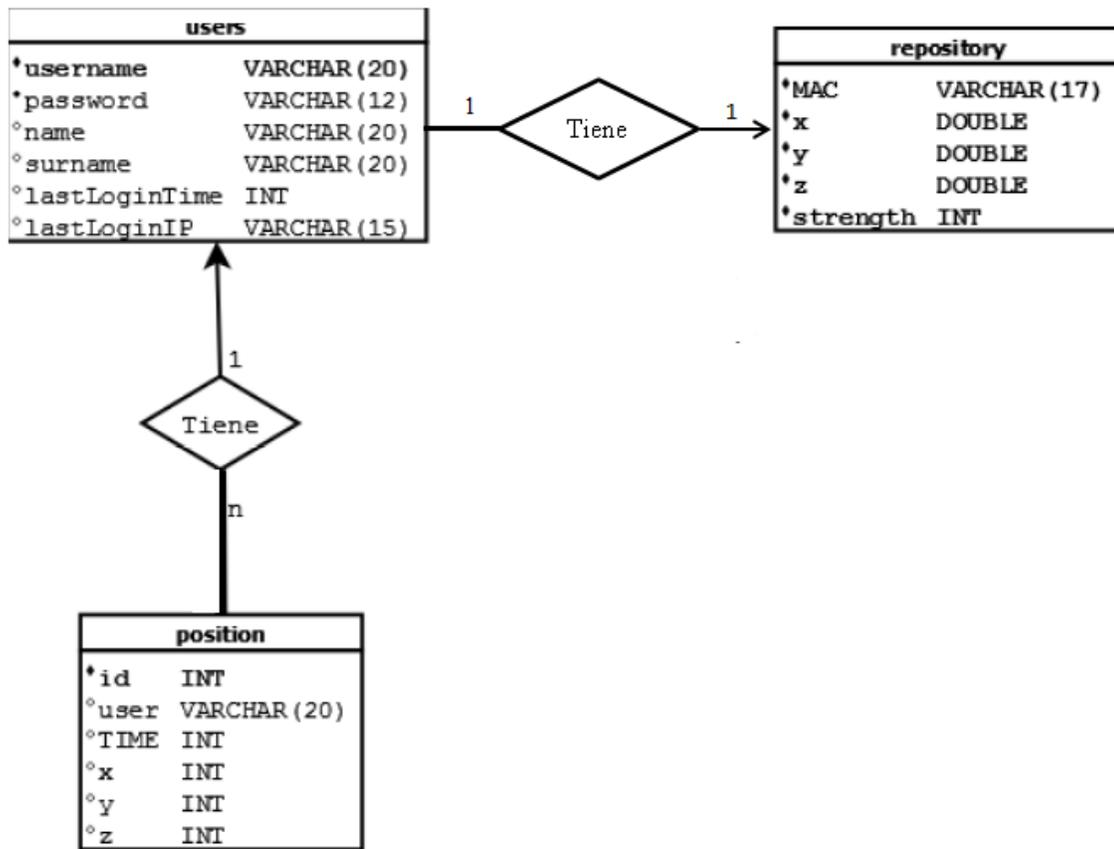


Figura 12: Modelo E-R de la Base de Datos

---

# CAPÍTULO V

---

## GESTIÓN DEL PROYECTO

---

### 5.1. ANÁLISIS DE REQUISITOS

En esta sección se van a explicar los requisitos del proyecto, haciendo distinción entre la aplicación cliente, que es la aplicación Android, y la aplicación del servidor, que sirve para gestionar los usuarios conectados, calcular las rutas mínimas y generar las instrucciones. Se hace también una distinción entre los requisitos funcionales y los no funcionales. Los requisitos funcionales describen los servicios que proporciona el sistema (funciones), la respuesta del sistema ante determinadas entradas y el comportamiento del sistema en situaciones particulares, es decir toda la funcionalidad del producto final. Los requisitos no funcionales, son restricciones de los servicios o funciones que ofrece el sistema, es decir, las condiciones en las que debe funcionar el proyecto. Se ha dividido a su vez, en dos categorías diferenciando entre software y hardware.

#### 5.1.1. Aplicación móvil

##### 5.1.1.1. Análisis de requisitos funcionales

- **Debe ser capaz de acceder a bajo nivel a la capa de comunicación:** para poder lograr el posicionamiento Wi-Fi, la aplicación necesitará acceder a datos de bajo nivel sobre los puntos de acceso con el fin de realizar la triangulación necesaria.
- **Debe ser capaz de procesar la entrada, el destino, que el usuario indique por voz al sistema:** para lograr el sistema por voz, la aplicación deberá realizar el reconocimiento de voz correctamente sobre el destino que indique el usuario

- **Debe ser capaz de reproducir la ruta mínima al usuario para llegar al destino:** igual que el requisito anterior para que el sistema de voz funcione correctamente, la aplicación debe ser capaz de reproducir la ruta mínima calculada al usuario.
- **Debe ser capaz de posicionarme en el mapa con la máxima precisión posible:** para lograr un control más exacto de la situación actual del usuario el sistema deberá realizar el posicionamiento actual con la máxima precisión posible.

#### 5.1.1.2. Análisis de requisitos no funcionales: Software

- **Versiones de Android, desde la 2.3 hasta la 4.2:** Estas son las versiones para las que se ha establecido el funcionamiento de la aplicación.

#### 5.1.1.3. Análisis de requisitos no funcionales: Hardware

- **Samsung Galaxy Mini:** es el dispositivo del que se ha dispuesto para desarrollar la aplicación. Debido a las limitaciones presupuestarias no se ha dispuesto de un mejor terminal para el desarrollo, pero se han hecho pruebas en otros terminales móviles para verificar el funcionamiento en las versiones superiores de Android.

### 5.1.2. Servidor

#### 5.1.2.1. Análisis de requisitos funcionales

- **Debe proporcionar conexión en tiempo real con el terminal móvil:** para que el servidor pueda ser usado en tiempo real ha de asegurarse una conexión entre el terminal móvil y el servidor en tiempo real y sin interrupciones.
- **Debe calcular y generar la trayectoria mínima que debe seguir el usuario, establecidos un origen y un destino:** la funcionalidad principal de este proyecto es la generación de las rutas y las trayectorias que el usuario debe seguir, por lo que es un requisito trivial para el éxito del proyecto.
- **Debe ser capaz de generar una serie de instrucciones en lenguaje natural a partir de la ruta calculada:** al igual que el anterior, se convierte en un requisito indispensable para el éxito del proyecto, pues necesitamos que el usuario obtenga la lista de instrucciones para poder llegar al destino.
- **Debe hacer sencillo una posible ampliación del proyecto:** en un futuro puede ser posible una ampliación del proyecto o añadir nueva funcionalidad, por lo que se debe proporcionar una arquitectura lo más modular que sea posible. Se intentará definir los módulos lo mejor posible y separar lógicamente cada capa de acceso para facilitar la tarea de los futuros programadores.

### 5.1.2.2. Análisis de requisitos no funcionales: Software

- **Versión de java 1.7.0\_09:** es la versión de Java que se ha utilizado en el ordenador para desarrollar el proyecto.

## 5.2. GESTIÓN DE RIESGOS

Riesgo, es todo aquella que puede afectar negativamente al proyecto. Es el motivo por el que antes de comenzar, debemos realizar un análisis de riesgos, para evitar un desastre en la gestión del proyecto y poder detectar los posibles riesgos y adoptar una estrategia en caso de que estos ocurran [2] [21].

Para la gestión de riesgos, se decide adoptar una estrategia proactiva, identificando los riesgos antes de caer en ellos y proponiendo alternativas para evitar que estos puedan ocurrir, así como posibles soluciones si finalmente los riesgos se llegan a producir.

Para aquellos riesgos que no se hayan podido evitar o prever, se adoptará una estrategia reactiva, identificando el riesgo así como las causas que lo han producido. Se asignaran recursos por si los riesgos se convierten en problemas o vuelven a surgir.

En su artículo de 1991 [22], *Boehm* fija las bases para la gestión del riesgo en el software, y es esta técnica la que se adopta para este proyecto. Primero se realiza una valoración del riesgo y luego se hace un control del riesgo.

Identificamos los posibles riesgos que pueden ocurrir durante el proyecto clasificándolos en tres tipos posibles:

- Riesgos del proyecto, son lo que amenazan al plan del proyecto.
- Riesgos técnicos, son los que amenazan la calidad del software.
- Riesgos del negocio, son lo que amenazan la viabilidad del proyecto.

Para cada uno de los riesgos que a continuación se presentan, se realiza en un primer momento un listado recogiendo el nivel de criticidad que presenta cada riesgo en base a la relación entre la probabilidad y la severidad de los mismos. Este nivel de criticidad se toma en base a la técnica SQAS-SEI [23] dónde se establecen los siguientes valores:

Probabilidad/ Severidad	Frecuente	Probable	Ocasional	Remoto	Improbable
Catastrófico	IN	IN	IN	A	M
Crítico	IN	IN	A	M	B
Serio	A	A	M	B	T
Menor	M	M	B	T	T
Insignificante	M	B	T	T	T

IN: INTOLERABLE: ALTO M: MEDIO B: BAJO T: TOLERABLE

Tabla 2: Relación Probabilidad-Severidad SQAS-SEI

### 5.2.1. Análisis de los Riesgos

Seguidamente se resume la lista de riesgos identificados en formato tabla, recogiendo la probabilidad y la severidad de cada riesgo y el nivel de criticidad que provoca. A riesgo uno se le ha asignado un código de forma “R-XX”, donde XX establece el número de riesgo que se asigna al requisito según el orden en que se identificó. Asimismo, se asigna un nombre “textual” para referirse a él de manera más formal.

Código: Nombre	Probabilidad	Severidad	Nivel de Criticidad
R-01: Abandono de miembros del equipo	Frecuente	Critica	INTOLERABLE
R-02: Malas decisiones en la organización	Ocasional	Seria	MEDIO
R-03: Retraso en la entrega	Remota	Seria	BAJO
R-04: Falta de dedicación al proyecto en periodos de vacaciones y exámenes	Probable	Critica	INTOLERABLE
R-05: Problemas de rendimiento con el terminal móvil	Frecuente	Menor	MEDIO
R-06: Incompatibilidad funcional entre las versiones de Android	Remoto	Catastrófico	ALTO
R-07: Falta de formación y conocimiento en las tecnologías utilizadas	Probable	Critico	INTOLERABLE

Tabla 3: Nivel de Criticidad de los Riesgos en función de la Probabilidad-Severidad

### 5.2.2. Priorización de los Riesgos

La priorización de los riesgos se realizará en base a su nivel de criticidad. Los riesgos más graves serán aquellos de nivel de criticidad más alto:

Prioridad	Nombre del Riesgo	Nivel de Criticidad
1	R-01: Abandono de miembros del equipo	INTOLERABLE
2	R-04: Falta de dedicación al proyecto en periodos de vacaciones y exámenes	INTOLERABLE
3	R-07: Falta de formación y conocimiento en las tecnologías utilizadas	INTOLERABLE
4	R-06: Incompatibilidad funcional entre las versiones de Android	ALTO
5	R-02: Malas decisiones en la organización	MEDIO
6	R-05: Problemas de rendimiento con el terminal móvil	MEDIO
7	R-03: Retraso en la entrega	BAJO

Tabla 4: Priorización de los Riesgos

### 5.2.3. Estudio de los Riesgos

A continuación se detalla, además de la clasificación, la especificación detallada de cada riesgo, abarcando una descripción, causas, estrategias para evitarlo, medidores, estrategias para minimizarlo y una plan de contingencia.

#### [R - 01] Abandono de miembros del equipo

**Tipo de Riesgo:** Proyecto

**Probabilidad:** Frecuente

**Severidad:** Crítica

**Nivel de Criticidad:** Intolerable

**Descripción:** Algún o algunos componentes del equipo dejan el proyecto.

**Causas:**

- Dedicación a otras actividades incompatibles con la realización del proyecto.
- Prioridad en otras actividades externas al proyecto.

**Estrategias para evitarlo:**

- Comprometerse realizar el proyecto establecido.

**Medidores:**

- Falta de interés en la realización del proyecto.
- Falta de dedicación al proyecto.
- No asistencia a las reuniones.
- No utilización del repositorio.

**Estrategias para minimizarlo:**

- Realizar una buena distribución de las tareas a realizar, minimizando la carga de trabajo a los miembros cuando sea posible.
- Para cada actividad que haya que realizar, todos los integrantes del grupo deben conocerla completamente.

**Plan de contingencia:** Reorganización del proyecto así como de las tareas a realizar.

**[R - 02] Malas decisiones en la organización**

**Tipo de Riesgo:** Proyecto

**Probabilidad:** Ocasional

**Severidad:** Seria

**Nivel de Criticidad:** Medio

**Descripción:**

- Se toman malas decisiones en la organización de objetivos para el desarrollo del proyecto, no se tienen claros los objetivos a cumplir o existen dudas sobre si es viable o no conseguir los objetivos.

**Causas:**

- No se conoce lo suficiente el proyecto para su desarrollo.

**Estrategias para evitarlo:**

- Se establece un tiempo extra para prevenir retrasos.
- Se revisa la planificación del proyecto en cada reunión con el director de proyecto para ir estableciendo los objetivos cumplidos y los que queda por cumplir.

**Medidores:**

- No se realizan las tareas a tiempo.
- Falta de coherencia en los documentos.
- Cuando queda poco tiempo para la entrega queda demasiadas tareas por realizar.

**Estrategias para minimizarlo:**

- Realizar una revisión de la planificación cada dos semanas, para comprobar que no existan problemas serios.

**Plan de contingencia:** Reestructuración de la planificación.

**[R - 03] Retraso en la entrega****Tipo de Riesgo:** Proyecto**Probabilidad:** Remota**Severidad:** Seria**Nivel de Criticidad:** Bajo**Descripción:** El proyecto o las entregas no dan tiempo a entregarse.**Causas:**

- Debido a la complejidad del proyecto.
- Falta de compromiso por parte de algún componente del equipo.

**Estrategias para evitarlo:**

- Establecer fechas de entregas.
- Ir cumpliendo objetivos establecidos por el jefe de proyecto.
- Mantener comunicación con el jefe de proyecto para aclarar si se está en plazo con los objetivos establecidos.

**Medidores:**

- No se cumple la planificación.
- La planificación fue errónea.
- Falta de compromiso de los miembros del equipo.

**Estrategias para minimizarlo:**

- Si el retraso no es muy grande intentar remediar y realizar el trabajo pendiente.
- Si el retraso es importante hablar con el jefe de proyecto.

**Plan de contingencia:** Reorganizar las tareas.**[R - 04]: Falta de dedicación al proyecto en periodos de vacaciones y exámenes****Tipo de Riesgo:** Proyecto**Probabilidad:** Probable**Severidad:** Crítica**Nivel de Criticidad:** Intolerable**Descripción:**

- No realizar ni avanzar el proyecto en los periodos de vacaciones o épocas de exámenes.

**Causas:**

- Falta de tiempo en época de exámenes.
- Falta de dedicación en la época de vacaciones.

**Estrategias para evitarlo:**

- Confianza en el compromiso de los miembros del equipo.

**Medidores:**

- No se cumple la planificación establecida para el proyecto.

**Estrategias para minimizarlo:**

- Mayor realización y dedicación al proyecto en los periodos no vacacionales y épocas de exámenes.
- Avanzar el proyecto en la medida de lo posible en las épocas vacacionales.

**Plan de contingencia:** Reorganizar las tareas para conseguir los objetivos establecidos.

**[R - 05] Problemas de rendimiento con el terminal móvil**

**Tipo de Riesgo:** Técnico

**Probabilidad:** Frecuente

**Severidad:** Menor

**Nivel de Criticidad:** Medio

**Descripción:** Disponer de un terminal móvil que no se caracteriza por sus grandes prestaciones, como es este caso, teniendo un *Samsung Galaxy Mini* para realizar las pruebas.

**Causas:**

- El teléfono no responde como se espera a la aplicación implementada.

**Estrategias para evitarlo:**

- Disponer de un terminal con mejores prestaciones.

**Medidores:**

- Antigüedad del dispositivo móvil.
- El móvil no responde como es debido, incluso hay veces que ni responde.

**Estrategias para minimizarlo:**

- Realizar las pruebas en la medida de lo posible en terminales con mejores prestaciones para comprobar que la aplicación funciona correctamente.

**Plan de contingencia:**

- Puesto que es un riesgo externo al desarrollo de la aplicación, no se puede hacer nada al respecto, ya que se dispone de fondos para la adquisición de un nuevo terminal.

**[R - 06] Incompatibilidad funcional entre las versiones de Android**

**Tipo de Riesgo:** Técnico

**Probabilidad:** Remoto

**Severidad:** Catastrófico

**Nivel de Criticidad:** Alto

**Descripción:** El acceso a los servidores, desde la versión 4.0 en adelante, se realiza de forma diferente a como se estaba planteando.

**Causas:**

- Se ha establecido una nueva forma de implementación de acceso a los servidores para las nuevas versiones de Android<sup>14</sup>.

**Estrategias para evitarlo:**

- Encontrar la forma de conectar correctamente con el servidor.

**Medidores:**

- La aplicación no responde como se espera.

**Estrategias para minimizarlo:**

- No es un riesgo que se pueda minimizar de alguna manera.

**Plan de contingencia:**

- Encontrar la solución a este problema es imprescindible para la continuación del proyecto, pues sin acceso al servidor la aplicación no hace nada.

---

<sup>14</sup> <http://www.androiddesignpatterns.com/2012/06/app-force-close-honeycomb-ics.html>

**[R - 07] Falta de formación y conocimiento en las tecnologías utilizadas**

**Tipo de Riesgo:** Proyecto

**Probabilidad:** Probable

**Severidad:** Crítica

**Nivel de Criticidad:** Intolerable.

**Descripción:** No tener los conocimientos necesarios para el desarrollo del proyecto.

**Causas:**

- No haber recibido la formación necesaria para el desarrollo de este proyecto.
- Falta de interés en querer buscar información de los conocimientos que no se dominan.

**Estrategias para evitarlo:**

- Buscar información en libros e internet.

**Medidores:**

- No realiza las tareas según los cánones pactados.
- No realiza las tareas a tiempo.
- No busca información en libros, apuntes, internet...

**Estrategias para minimizarlo:**

- Las personas que tengan más conocimientos del tema pueden ayudar en la medida de lo posible a resolver dichas dudas.

**Plan de contingencia:**

- Realización de curso y búsqueda de la información necesaria.

### 5.3. PLANIFICACIÓN DEL PROYECTO

La planificación del proyecto se ha ido realizando por objetivos, estableciendo unos objetivos mínimos para el proyecto que había cumplir que establecer las siguientes tareas.

El primer objetivo planteado consistió en la adaptación del proyecto AVANTI [1], a las nuevas versiones de Android, lo que conllevaba todo el aprendizaje de esta tecnología, desconocida para mí. Esta primera etapa vino acompañada de varios contratiempos. Por un lado, inicialmente el proyecto fue planteado para ser elaborado por tres personas, que finalmente ha sido sacado adelante por una sola. Por otro lado, la adaptación a las nuevas versiones de Android no era tan trivial como se creía en un primer momento, pues las nuevas versiones de este sistema operativo tienen como premisa preestablecida el empleo de multitareas para evitar esperas en el sistema y este hecho no estaba contemplado para las primeras versiones.

Cumplido el primer objetivo y reestructurado el proyecto, se plantea la realización del sistema de guía por voz, en dónde el usuario debe ser capaz de poder comunicar al terminal el destino deseado y el terminal debe ser capaz de transmitir la información necesaria para poder alcanzar dicho destino.

El tercer objetivo planteado fue la creación del algoritmo que calculase la ruta mínima que debía seguir el usuario para alcanzar el destino solicitado. Para realizar dicho algoritmo se debió tener en cuenta numerosos factores, entre ellos están el modo de realizar el cálculo de dicha ruta, la comunicación que existiría entre esta aplicación y el terminal móvil, el caso en el que el usuario se saliese de la ruta establecida, etc.

Una vez conseguida la ruta mínima de forma numérica, se plantea la transformación de dicha ruta a un lenguaje natural, para poder ser transmitida al usuario. Dicha transformación se consigue gracias a un generador de lenguaje natural. Para poder utilizar dicho generador se debe realizar una conversión previa de los datos que contienen la ruta mínima, de esta forma se consigue una adaptación entre los datos generados y los datos que el generador acepta para realizar dicha conversión.

Realizados todos los objetivos para el funcionamiento del sistema, se procede a la última tarea, que consiste en la unificación de todas las partes que se han ido realizando y comprobar que el funcionamiento del sistema es el adecuado.

---

# CAPÍTULO VI

---

## DISEÑO E IMPLEMENTACIÓN

---

### 6.1. INTRODUCCIÓN

En este capítulo se va a realizar un análisis más detallado de la función principal que caracteriza a este proyecto, se explicará cómo está diseñado e implementado el funcionamiento interno del sistema. Se comenzará detallando el diseño y la implementación tanto para el cliente como para el servidor, se explicará la conexión entre las aplicaciones cliente y servidor para el intercambio de datos, así como la obtención del destino por parte del sistema gracias a la entrada por voz del usuario.

### 6.2. DISEÑO E IMPLEMENTACIÓN CLIENTE

La aplicación móvil, es decir el cliente, contiene toda la funcionalidad Android, es la que se encarga de interactuar con el usuario gracias a las actividades propias de este sistema operativo.

Esta aplicación cliente es la encargada de la obtención de los datos necesarios para poder realizar el cálculo de la ruta que permita al usuario alcanzar el destino. Como ya se ha mencionado en esta memoria, la aplicación Android está formada, en parte, por la funcionalidad reutilizada del proyecto AVANTI [1], modificada y adaptada para su correcto funcionamiento en la nuevas versiones del sistema operativo Android.

Por otro lado se ha añadido nueva funcionalidad a la que ya se disponía, como es el Sistema de Voz, que permite al usuario interactuar con la aplicación sin necesidad de utilizar el teclado, así como la conexión con el servidor que permita realizar el intercambio de datos entre el cliente y el servidor, y que se explicará más adelante en este capítulo.

A continuación se muestra el diagrama de clases, que recoge las clases de las actividades principales de la Aplicación Android:

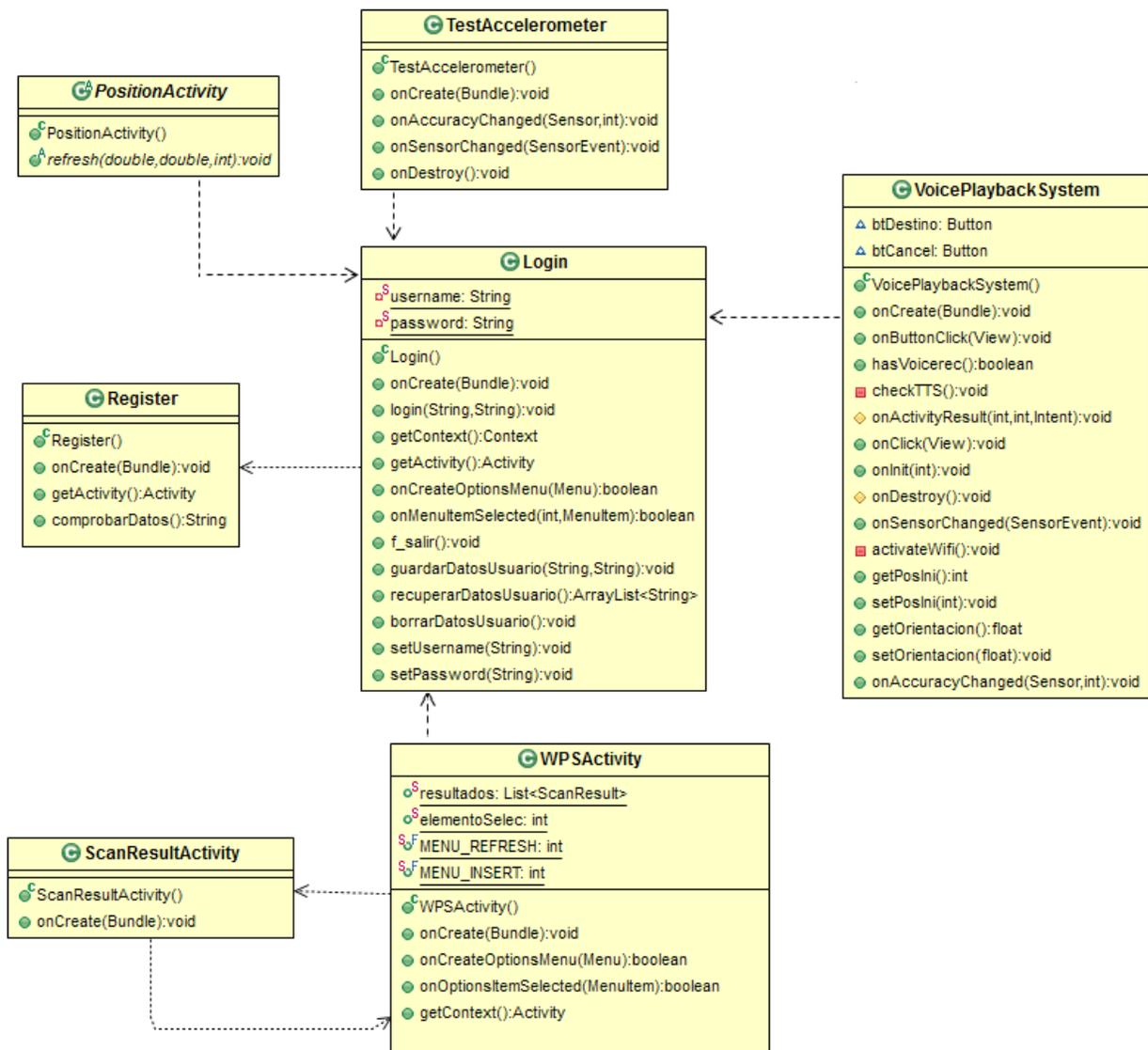


Figura 13: Diagrama de Clases Aplicación Cliente

Engloba el conjunto de clases de la aplicación que hereda de Activity:

- La clase **Login** se refiere a la pantalla de acceso a la aplicación. La función *login* realiza el acceso a la aplicación, comprobando si hay algún usuario en la base de datos con la contraseña introducida. Las funciones *guardarDatosUsuario* y *recuperarDatosUsuario* permiten almacenar los datos del último usuario que ha accedido a la aplicación, evitando tener que escribirlos la próxima vez que acceda a la aplicación. Por otra parte, la función *borrarDatosUsuario* elimina los datos almacenados del último usuario que accedió a la aplicación.

- la clase **Register** prepara la pantalla de registro de la aplicación, donde se encuentran los distintos campos a rellenar.
- **TestAccelerometer** engloba la prueba del acelerómetro.
- La clase **Position** actualiza los datos de posición del usuario (coordenadas X e Y, y cuadrante en el que se encuentra) mediante la función *refresh*.
- **WPSActivity** comprende todo lo relativo a la actividad Wi-Fi: el método *refresh* actualiza las redes detectadas en el escaneo y el método *CreateInserDialog* prepara una pantalla para insertar una nueva posición en la base de datos.
- La clase **ScanResultActivity** prepara los datos de la red seleccionada en la pantalla de Actividad Wi-Fi, mostrándolos en una pantalla nueva.
- **VoicePlaybackSystem** demuestra el uso del sistema de voz. Tanto el reconocimiento de voz como la reproducción de voz están en la misma actividad. El método *onActivityResult* es el encargado de diferenciar entre el reconocimiento o la reproducción, lanzando en cada caso la intención deseada. El método *onSensorChanged* es el encargado de la recopilación inicial de los datos referentes a la posición actual y la orientación del usuario, posteriormente este método es el que establece la comunicación con el servidor para el intercambio de datos.

### 6.3. DISEÑO E IMPLEMENTACIÓN SERVIDOR

La aplicación encargada del cálculo de rutas, como se ha explicado, se encuentra ejecutándose de forma permanente en el servidor. Se trata de una aplicación Java, "CalculaRuta", que se encarga una vez recibidos los datos necesarios del cliente, de calcular la ruta mínima establecidos un origen y un destino.

A continuación se presente el diagrama de clases de la aplicación servidor "CalculaRuta":

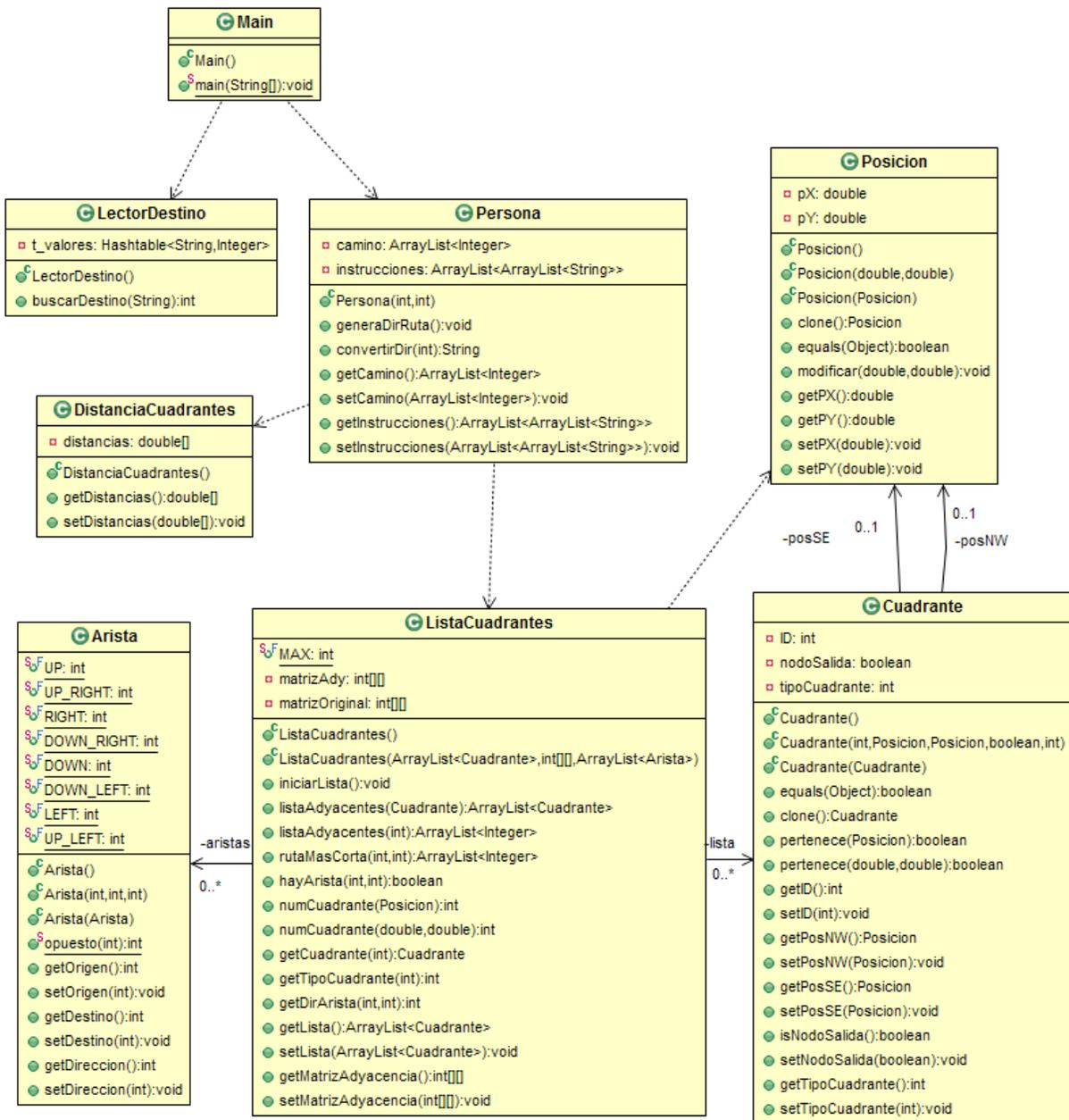


Figura 14: Diagrama de Clases de la Aplicación Servidor

Esta aplicación contiene todas las clases necesarias para el cálculo de posibles rutas:

- Los objetos de la clase **Cuadrante** vienen identificados por las coordenadas de su esquina superior izquierda y de su esquina inferior derecha, así como un identificador único para cada uno de ellos.
- La clase **ListaCuadrantes** contiene la representación del conjunto de cuadrantes en que se han dividido los pasillos de la primera planta de la Facultad de Informática. Para ello, se ha representado el conjunto de cuadrantes como un grafo mediante una matriz de adyacencia en el atributo *matrizAdy*. En el método *iniciarLista*, se crean el conjunto de

cuadrantes de los pasillos, la Sala de Grados y la Sala de Juntas, y las conexiones entre los mismos. Otro de los métodos fundamentales de esta clase es *rutaMasCorta*, que calcula el camino que hay que seguir para alcanzar la salida más cercana.

- La clase **Arista** almacena el identificador del cuadrante de origen y de destino, así como la dirección de la misma.
- La clase Posición es una clase auxiliar para representar las coordenadas en los ejes X e Y.
- La clase **DistanciaCuadrantes** almacena para cada tipo de cuadrante la media distancia del alto, ancho y diagonal expresada en metros.
- La clase **LectorDestino** es la encargada de la obtención de la posición destino a partir de la entrada por voz del usuario gracias a la función *buscarDestino*.
- **Persona** es la clase encargada de la generación de la lista que contiene las instrucciones de la ruta, esto se realiza gracias al método *generarDirRuta*.

Para mayor entendimiento del sistema, se presentan a continuación los diagramas de secuencia de la funcionalidad más importante de la aplicación servidor, realizando una explicación de cómo se ha realizado dicha funcionalidad.

### 6.3.1. Obtención del destino a partir de la entrada por voz

Una vez el usuario ha introducido el destino por voz, este queda almacenado como una variable *String* que posteriormente será enviada al servidor para localizar la posición de este destino. Para lograr obtener la posición en el servidor a partir del *String* que se recibe se ha empleado el formato ligero **JSON**<sup>15</sup>, con el que conseguimos tener almacenados en el propio servidor los posibles destinos en la primera planta de la Facultad de Informática. Se ha empleado el uso de este formato para obtener mayor eficiencia, evitando la continua lectura de un *array*, que almacenarían la información de un fichero de texto, y al que habría que estar recorriendo para buscar la posición que necesitamos buscar.

Para evitar esto, como ya se ha comentado, utilizamos el formato JSON, que nos permite realizar un único recorrido del fichero para almacenar los datos en un **Hashtable**<sup>16</sup>, utilizando como clave el destino y como valor el cuadrante al que pertenece. De esta forma conseguimos la posición en la que se encuentra el destino del usuario con una llamada de consulta por clave. En caso de no existir el destino introducido por el usuario, no se realiza el cálculo de la ruta, sino que se manda al usuario el mensaje de error, notificando el destino inexistente y el impedimento al cálculo de la ruta.

---

<sup>15</sup> <http://www.json.org/>

<sup>16</sup> <http://docs.oracle.com/javase/6/docs/api/java/util/Hashtable.html>

Se puede observar en el siguiente diagrama de secuencia el desarrollo de esta actividad:

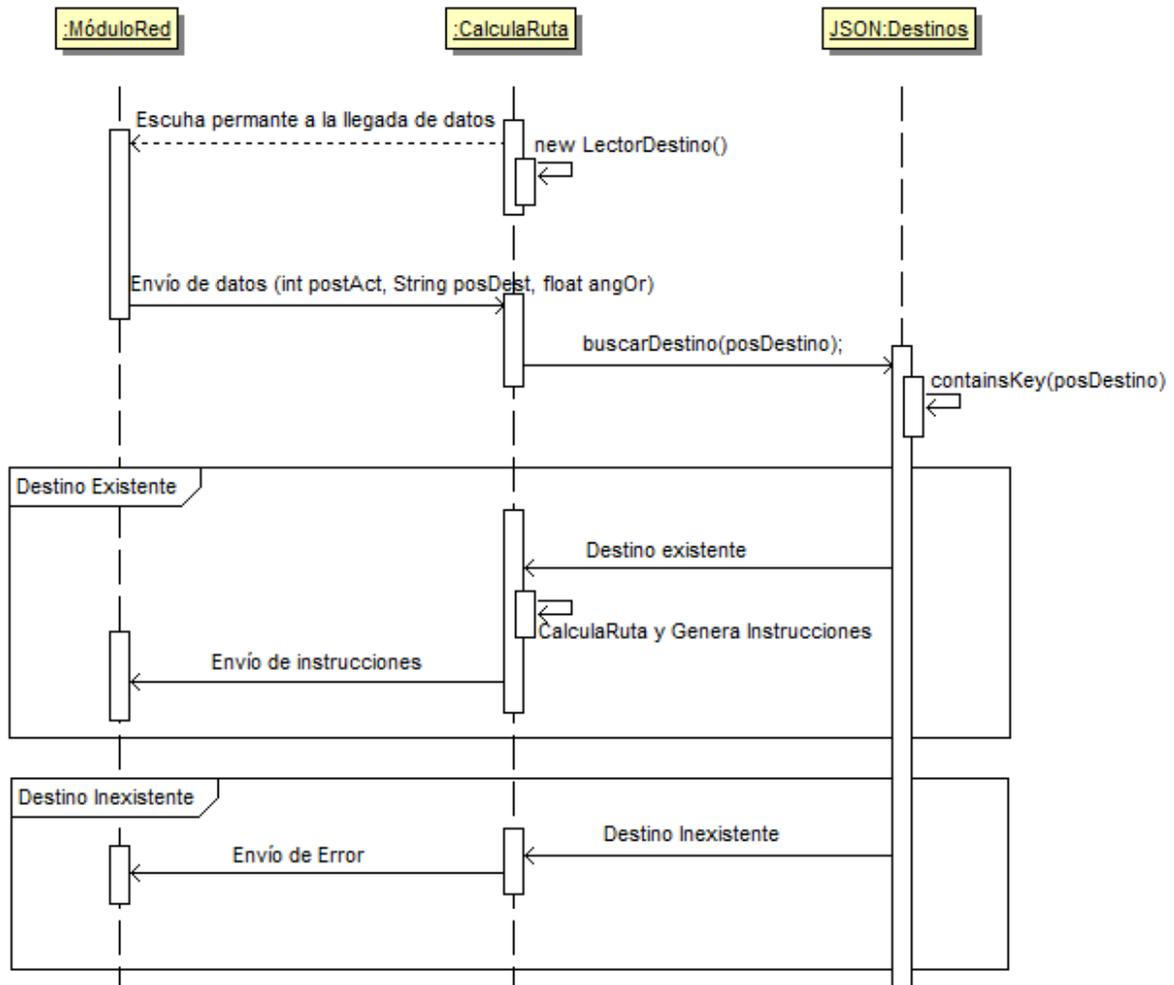


Figura 15: Diagrama de Secuencia de la Obtención del Destino a partir de la entrada por voz

### 6.3.2. Cálculo de la ruta mínima

El cálculo de la ruta mínima, cómo ya se explicó en el capítulo anterior, se consigue gracias a la aplicación del algoritmo de Dijkstra, que permite calcular la distancia mínima entre dos puntos.

Para calcular dicha distancia mínima, inicialmente se posiciona al usuario en el cuadrante que contiene la posición actual del usuario, y gracias a la obtención del destino a partir de la entrada por voz, explicada en el punto anterior, obtenemos el cuadrante destino al que el usuario desea ir. Lo único que quedaría sería realizar el cálculo de esta ruta mínima entre estos dos cuadrantes.

Obtenida la lista de cuadrantes por los que el usuario debería pasar para llegar al objetivo. Se comienza con el algoritmo para calcular las direcciones y distancias, pues si el usuario debe seguir al norte durante cuatro cuadrantes, no se generarán cuatro instrucciones, sino que generaríamos

una instrucción equivalente a esa indicando la dirección que debe seguir en esos cuatro cuadrantes y la distancia total que supondría recorrer dichos cuadrantes. Para ello se utiliza un `<ArrayList<ArrayList<String>>`, e almacenará la lista de instrucciones que se deben seguir, indicando para cada instrucción la dirección que debe seguir así como la distancia que debe recorrer en dicha instrucción.

Una vez obtenida se ha obtenido la lista de instrucciones, se realiza la llamada al generador de lenguaje natural, que será el encargado de transformar esta lista numérica de instrucciones en una cadena de instrucciones en lenguaje natural. Dicha llamada al generador se realiza pasando como parámetros la lista generada y el ángulo de orientación inicial del usuario, necesario para saber a dónde está mirando el usuario para generar una trayectoria correcta y que el usuario no se desvíe hacia otro lado.

El siguiente diagrama de secuencia muestra las llamadas principales de toda la actividad anteriormente descrita:

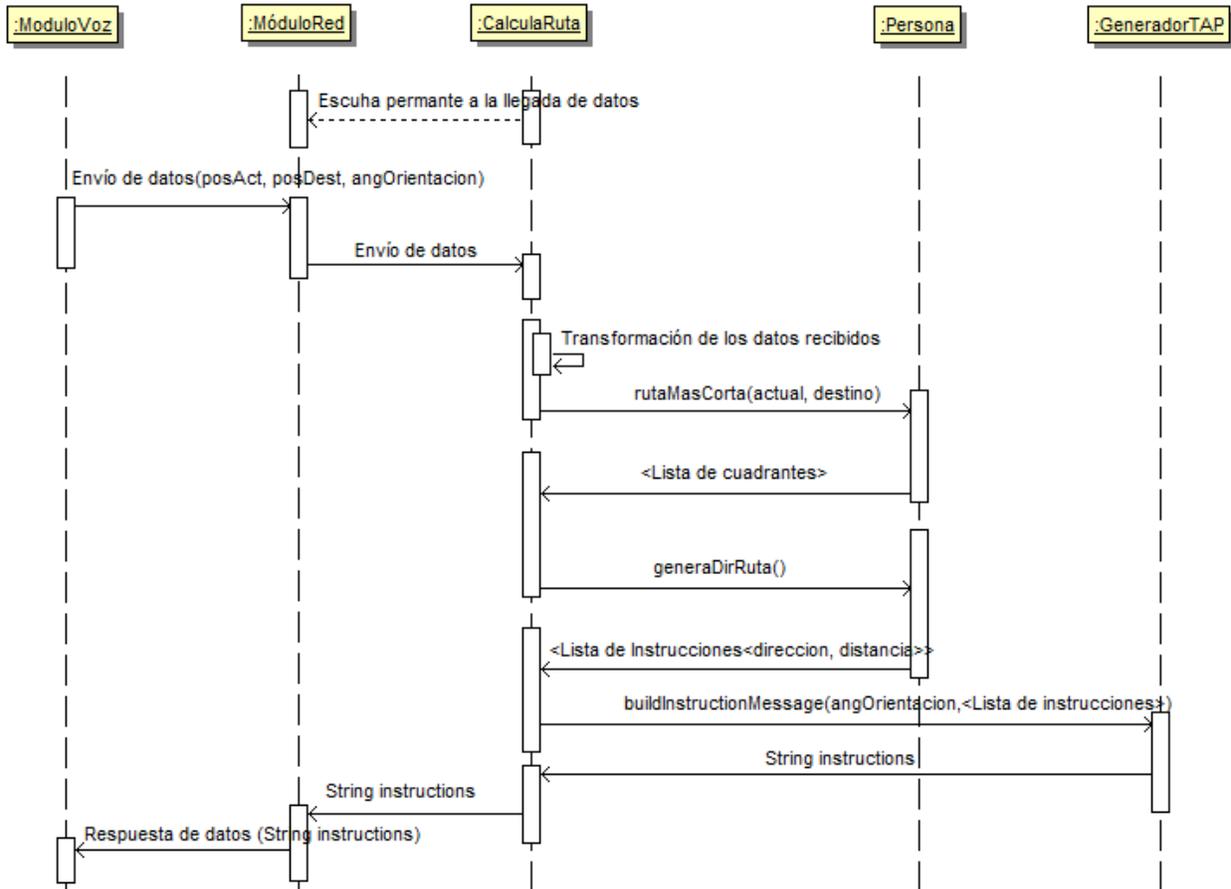


Figura 16: Diagrama de Secuencia del cálculo de la ruta mínima

## 6.4. CONEXIÓN CLIENTE-SERVIDOR

La manera de conectar la aplicación Android con la aplicación servidor para permitir el continuo intercambio de datos se ha conseguido gracias a la clase **java.net.Socket**<sup>17</sup>, que nos permite instanciar un objeto de tipo Socket en cada una de las aplicaciones y poder realizar de esta forma, el intercambio de datos entre las mismas, mediante un protocolo TCP<sup>18</sup> (*Transmission Control Protocol*).

Podemos ver con más detalle, en el siguiente diagrama de secuencia, el desarrollo de la actividad principal de este proyecto, “VoicePlayBackSystem” que es donde se realiza, en primer lugar la recopilación de datos necesarios para el cálculo de la ruta (posición actual, posición destino y ángulo de orientación), en segundo lugar la conexión entre las dos aplicaciones, ella misma y el servidor y por último la recogida de la secuencia de instrucciones para su posterior reproducción por voz.

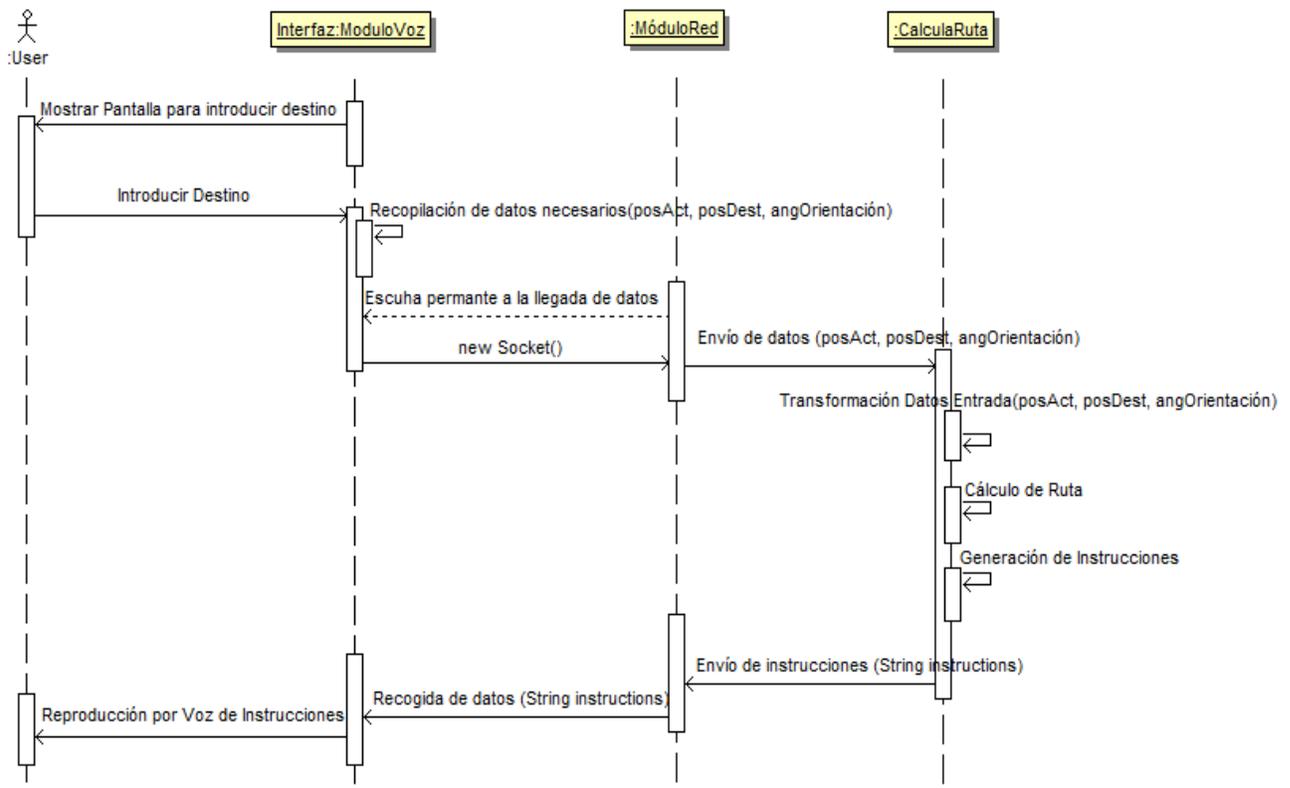


Figura 17: Diagrama de Secuencia de la conexión entre aplicaciones

<sup>17</sup> <http://developer.android.com/reference/java/net/Socket.html>

<sup>18</sup> <http://www.faqs.org/rfcs/rfc793.html>

---

# CAPÍTULO VII

## RESULTADOS, CONCLUSIONES Y TRABAJO FUTURO

---

### 7.1. DESCRIPCIÓN DE LOS RESULTADOS OBTENIDOS Y ALCANZADOS

La solución propuesta utilizando el terminal móvil, coordinándose con un servidor central para poder llevar a cabo simulaciones de guía en el interior del edificio, traza un puente entre las ventajas de una solución virtual y llevar a cabo una guía real del usuario.

El uso del posicionamiento Wi-Fi permite al sistema detectar el movimiento del usuario a través de un edificio, controlando la ruta que va siguiendo, pudiendo mandar un aviso cuando el usuario se sale de la ruta generada para llegar al destino que ha solicitado. Se puede tener en cuenta que el hecho de que el usuario se esté realmente moviendo a través del edificio asegura que la velocidad de movimiento es realista y que no habría problemas entre la representación virtual y la posición real.

El uso de Android como plataforma de desarrollo para el proyecto ha ayudado, en cierto modo, a la implementación del mismo, pues actualmente Android ha conseguido una madurez como sistema operativo pudiendo llegar a competir en el mercado con teléfonos como el iPhone de Apple, y captando una mayoría notable en la cuota de mercado. Esto provoca que surjan numerosas comunidades de desarrollo y recursos libres en Internet que hacen que el desarrollo de las aplicaciones sea un poco más sencillo.

Sin embargo, se ha comprobado al realizar algunas pruebas de la aplicación en diferentes terminales que no se ha conseguido una compatibilidad total del sistema operativo Android. Al testear la aplicación en modelos *Xperia* de la marca *Sony Ericsson*, surgieron problemas intentando combinar la vista de la cámara, lo que hacía que la aplicación fallara.

Otro hecho que ya dio problemas en la aplicación inicial y sigue ocurriendo, son las limitaciones que ofrecen el hardware de sensores en los dispositivos móviles, que presentan bastante margen de error. Este error se ha visto reflejado, por ejemplo, al acercar el terminal a algún campo magnético. Se puede ver como la señal que muestra la orientación inicial del usuario fluctúa, generando orientaciones que no son muy precisas. Esto puede presentar un error para el cálculo de ruta, pues la ruta generada para el usuario se calcula conforme a una orientación inicial que nos indica hacia dónde está mirando el usuario.

Por último, respecto al sistema de voz integrado en la aplicación, es un sistema muy desarrollado ya para todas las versiones de Android, y que no ha presentado problemas a la hora del reconocimiento y reproducción de voz. Tras varios modelos, se decidió integrar el reconocimiento y la reproducción en la misma actividad, pues necesitamos el destino que debe introducir el usuario para la generación de la ruta, en caso contrario no hay ruta que especificar.

### **7.1.1. Resultados positivos**

Tras finalizar toda la aplicación, podemos concluir unos resultados positivos de la misma que se detallan a continuación:

- Desarrollar un modelo que combinase la información de la brújula, del posicionamiento Wi-Fi y fuese capaz de integrar un sistema de voz por medio, es posible. Ha sido una tarea un poco complicada y puesto que los datos de entrada no son todo lo precisos que se hubieran deseado, debido a las limitaciones que presentan los sensores y el posicionamiento, proporcionan un buen punto de entrada para poder realizar el resto de la actividad del proyecto.
- El intercambio continuo de datos entre la aplicación cliente y la aplicación servidor, generó muy buenos resultados desde un primer momento. Gracias a esto, se consiguió lo que desde un primer momento se pretendía: que la aplicación móvil únicamente se encargase de recopilar los datos de entrada y mandárselos al servidor, el cual generaría la ruta que a su vez transformaría en una lista de instrucciones para que posteriormente, fueran reproducidas al usuario por el terminal. Este intercambio continuo permitió también controlar que el usuario no se saliese de la ruta establecida, mandando cada diez segundos, como ya se explicó, la posición al servidor para controlar que sigue la ruta establecida.

### **7.1.2. Resultados negativos**

Los resultados negativos que han limitado el proyecto se detallan a continuación:

- El tema del posicionamiento Wi-Fi, como ya fue explicado en [1], no es todo lo exacto que se desearía, lo que causa, en ocasiones, un mal posicionamiento inicial del usuario que provoca un cálculo erróneo de la ruta a seguir.

- El trabajo que conlleva realizar la captación de la señal de los puntos de acceso para poder realizar el posicionamiento, a lo que se añade la variación de la señal por factores externos, que conlleva a empeorar este posicionamiento pues la señal que se tiene almacenada, coincidirá en muy pocos casos con la señal que reciba el usuario en ese momento.

## **7.2. CONCLUSIONES**

Con el Proyecto Sistema de Guía por Voz en Interiores finalizado y tras exponer su contexto, bases y trabajo realizado se procede a evaluar el mismo de manera crítica.

Haciendo una vista global y general del proyecto, se puede concluir que los objetivos que se han establecido para la realización del presente proyecto se han cumplido en su mayor parte de forma satisfactoria. La idea de realizar un sistema de guía por voz en interiores utilizando la tecnología Android ha resultado un éxito.

Con el trabajo realizado en este proyecto de fin de carrera se han consolidado las bases para un sistema de guiado por voz para todo tipo de infraestructuras, como pueden ser hospitales o centros comerciales, gracias a que el sistema ha sido diseñado para poder ser aplicado a cualquier tipo de edificio que presente cierta necesidad de guiar a los usuarios por zonas en las que no están familiarizados.

El empleo de Android como plataforma ha resultado exitoso, una tecnología cuyo crecimiento mundial ha sido altamente notable pasando a ser uno de los grandes competidores de los sistemas en el mercado tecnológico, por lo que la experiencia adquirida durante el desarrollo de este proyecto será muy valorada.

Cabe destacar el trabajo de documentación realizado así como el proceso de aprendizaje de las nuevas tecnologías de desarrollo, desconocidas para mí. Este trabajo ha sido de gran valor e incluso indispensable para la consecución de los objetivos fijados para este proyecto.

En definitiva, creo que el trabajo ha merecido la pena y que gracias a la elaboración de este proyecto se puede afirmar que los sistemas de guías en interiores pueden ser posibles.

## **7.3. TRABAJO FUTURO**

Tras concluir este trabajo como un éxito al haber logrado lo que se pretendía, realizar un sistema de guía por voz que generase rutas a partir de los destino establecidos por el usuario, y se transmitiese al mismo por un sistema de voz, se pueden plantear diferentes líneas de desarrollo para aumentar o robustecer el sistema actual.

Una posible ampliación para este proyecto podría ser la incorporación de un sistema de guía visualmente. A través de la vista real, generar una serie de flechas según la ruta indicada que guíen al usuario hasta el destino.

Se podría plantear también un sistema de guía interactiva con el usuario, así como la segmentación de la trayectoria generada, ya que actualmente la trayectoria calculada es transmitida íntegramente al usuario, lo que puede producir complicaciones de entendimiento en rutas demasiado largas o complejas.

En cuanto a la localización en el edificio, sería interesante ampliar el escenario aumentando el número de plantas del mismo e ir ampliando hasta llegar a cubrir todas ellas para tener un modelo completo del edificio.

Otra posible ampliación podría ser en el cálculo de las rutas, estableciendo puntos visuales y conocidos por el usuario, es decir, generar rutas teniendo en cuenta el entorno en el que se encuentra localizado, pudiendo indicarle por ejemplo “gire cuando llegue al banco” o “baje las escaleras situadas a su derecha”. Esto llevaría también a un refinamiento de la generación de instrucciones a lenguaje natural, pudiendo llegar a una adaptación de las instrucciones según las características del usuario.

Otra línea futura estaría en el hecho de mejorar la localización en interiores. Se podría hacer un estudio de la mejor manera de distribuir los puntos de acceso en un edificio, para que se pueda cubrir la mayor cantidad de superficie con el menor número de puntos de acceso.

---

# APÉNDICE A

## MANUAL DE INSTALACIÓN DEL PROYECTO

---

A continuación se detalla un manual de instalación del proyecto actual. Se explicará tanto la instalación en un terminal de la aplicación móvil como la aplicación servidor.

### A.1. Requisitos

La aplicación ha sido testeada en su mayoría con el terminal *Samsung Galaxy Mini* con la versión 2.3.4 de Android y con el terminal *Newman N1* con la versión 4.0.4 de Android, donde la aplicación funcionaba correctamente. Se han realizado pruebas en otros terminales de la marca Samsung con versiones superiores del sistema operativo, llegando a la versión 4.2. En estos casos el funcionamiento siempre era correcto. No se puede garantizar el funcionamiento correcto en todas las marcas de teléfonos, existen limitaciones hardware, que hemos comprobado en el caso de probar la aplicación en un terminal de la marca *Sony Ericsson Xperia*.

### A.2. Instalación en Eclipse

Es necesario tener instalado el SDK de Android en Eclipse para poder ejecutar la aplicación. El proyecto soporta las versiones 2.3 a 4.2 de Android. Se puede instalar la versión que quieras que quede comprendida en ese rango para que el proyecto pueda ejecutarse.

Se proporciona la carpeta donde se encuentra el proyecto tal y como se usó en el IDE Eclipse. Para poder usar este mismo proyecto basta con realizar una importación de este al entorno. Esto se puede realizar haciendo clic en “File” y en el menú que despliega seleccionar “Import”. Una vez en ese menú hacer clic en “Android -> Existing Android Code into Workspace”, seleccionamos la ruta del proyecto en la ventana que sale y le damos a aceptar.

### A.3. Instalación en un terminal móvil

Se dispone del ejecutable de Android en la carpeta “./bin/Login.apk”. Este ejecutable se puede instalar en el terminal móvil utilizando un instalador de aplicaciones del estilo de “Apk Installer” disponibles en Play Store. También se puede ejecutar la aplicación desde eclipse en el propio terminal, o guardar mediante almacenamiento masivo el ejecutable “.apk” en el terminal como un archivo, e instalarlo posteriormente.

### A.4. Instalación del servidor

Para utilizar la aplicación es necesario tener un servidor corriendo con los archivos de la carpeta “AVANTI/PhpServer” en la raíz del mismo. El usuario puede utilizar los que se le ofrecen o gestionar su propio servidor si así lo desea.

Para instalar estos archivos el usuario debe instalar un servidor Apache con PHP y colocar estos archivos disponibles desde la red. Para hacer las pruebas en servidor local se aconseja utilizar “WampServer”<sup>19</sup> o “XamppServer”<sup>20</sup>.

A continuación debe cambiar los valores de las líneas 16, 17, 18 de la clase “HTTPServices” del paquete “com.avanti.http” con la URL de sus servidores y recompilar la aplicación.

Una vez hecho esto el usuario debe importar la base de datos en el servidor local para crear las tablas o instalar el servidor “Mysql”<sup>21</sup> y ejecutar el script “AVANTI/BackupBD/BD.sql” contra la base de datos para crear las tablas. Después, debe cambiar los archivos PHP del servidor Apache, anteriormente mencionados, para que hagan uso de su servidor, estableciendo la dirección, nombre de usuario y contraseña de la base de datos.

Por otro lado, la aplicación CalculaRuta, como ya se ha explicado, se encuentra ejecutándose permanentemente en el servidor, lo que es necesario subir dicha aplicación a un servidor que soporte Java y lanzar dicha aplicación.

### A.5. Puesta en Funcionamiento

Una vez se ha realizado la instalación de la aplicación en un terminal móvil y la instalación correcta del servidor, para que el sistema funcione correctamente es necesario realizar un mapeo previo del escenario en el que nos encontremos, en este caso la primera planta de la Facultad de Informática de la UCM, para almacenar en la base de datos toda la información referente a las direcciones MAC que va a permitir realizar posteriormente el posicionamiento del usuario.

---

<sup>19</sup> <http://www.wampserver.com/en/>

<sup>20</sup> <http://www.apachefriends.org/es/xampp-windows.html>

<sup>21</sup> <http://www.mysql.com/downloads/mysql/>

Para realizar dicho mapeo, se debe tener en cuenta las coordenadas establecidas en el sistema. Siendo X el largo del pasillo e Y el ancho del pasillo. Como coordenada Z se ha establecido el número 1, por tratarse de la primera planta.

---

# APÉNDICE B

## MANUAL DE USUARIO

---

A continuación se detalla un manual de usuario que recoge el funcionamiento principal y capturas de las actividades principales de la aplicación.

### **B.1. Pantalla de Inicio de la Aplicación**

Una vez instalada la aplicación en el terminal móvil, la iniciamos, abriéndose una pantalla inicial donde se mostrará un formulario de acceso. Si el usuario está registrado únicamente debe introducir su nombre de usuario y su contraseña y posteriormente pulsar el botón de *Entrar*. Si los datos no son correctos, se informará mediante un mensaje de error, pudiendo volver a introducir los datos.

En caso de no estar registrado, el usuario puede acceder al formulario de registro en la aplicación mediante el botón *Registrarse*.

Finalmente, si el usuario desea salir y cerrar la aplicación, puede abrir el menú mediante el botón correspondiente del teléfono y seleccionar la opción *Salir*.



Figura 18: Pantalla de Inicio de la Aplicación

## B.2. Pantalla de Registro de la Aplicación

En la pantalla de registro, habiendo accedido desde la pantalla principal, el usuario va a poder registrarse y de esta manera poder acceder al sistema. Para ello, debe introducir su nombre y apellidos, así como su nombre de usuario y su contraseña de acceso al sistema.

Si el usuario elige la opción de *Registrarse* y los datos son correctos, el usuario quedará registrado en la aplicación. Y para acceder a ella, únicamente debería rellenar el formulario principal de acceso. Si alguno de los datos no es correcto, se informará al usuario del error.

Por el contrario, si el usuario elige la opción de *Cancelar*, se volverá a la pantalla principal del formulario de acceso.

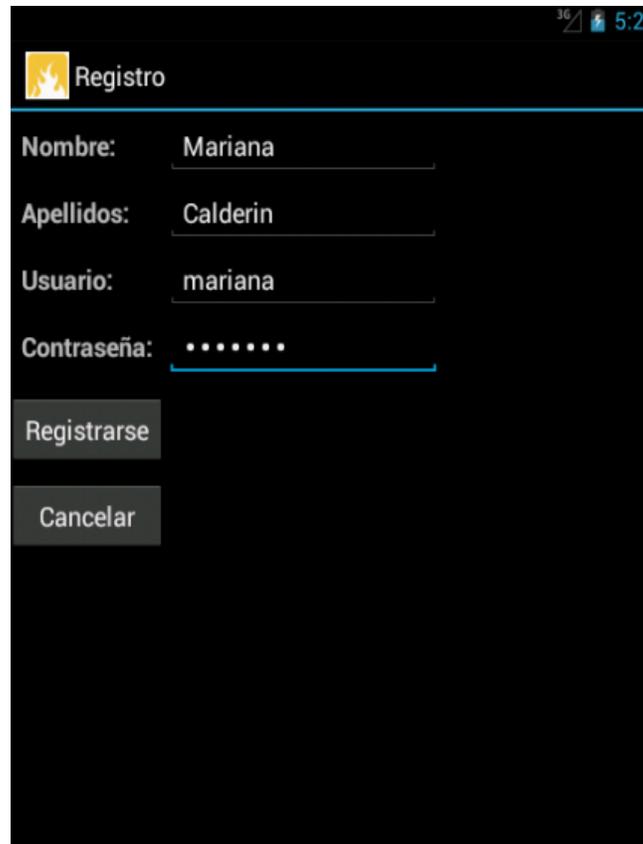


Figura 19: Pantalla de Registro de la Aplicación

### B.3. Menú principal de la Aplicación

Una vez hemos accedido al sistema, en el menú principal de la aplicación encontramos las siguientes opciones:

- Ver actividad Wifi
- Posicióname en el mapa
- Aplicación real
- Prueba del acelerómetro
- Indicar ruta hacia el destino
- Cerrar sesión

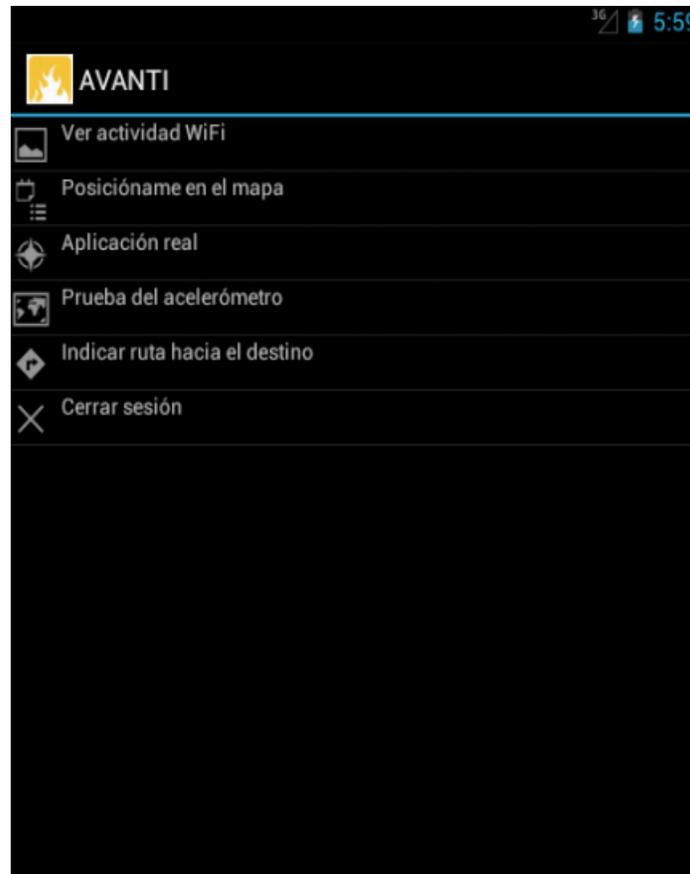


Figura 20: Pantalla del menú principal de la Aplicación

### B.3.1. Ver actividad Wifi

En esta pantalla, aparecerá una lista con las redes inalámbricas detectadas en la posición actual en la que se encuentra el usuario.

Si se selecciona cualquiera de estas redes, se abrirá una nueva ventana, donde aparecerán las características de dicha red:

- Nombre de la red (SSID)
- Dirección MAC (BSSID)
- Frecuencia
- Potencia

Si pulsamos el botón *Volver*, accedemos de nuevo a la pantalla dónde se muestra toda la actividad Wi-Fi.

Al pulsar el botón de menú en el terminal, aparecen tres opciones:

- La opción *Refrescar lista*, que realiza un nuevo escaneo de las redes.
- La opción *Insertar (Server)*, que abre un cuadro de diálogo para introducir las coordenada de la posición que queremos insertar en la base de datos para el posicionamiento. Si el usuario elige la opción *Aceptar* y los datos son correctos, se informará de que la operación se ha realizado con éxito, de lo contrario, se informará de que las coordenadas no son correctas. Si por el contrario, el usuario pulsa *Cancelar*, se cierra el cuadro de diálogo, volviendo a la pantalla con los resultados del escaneo.
- La opción *Salir*, que cierra esta pantalla, volviendo al menú principal de la Aplicación.

### **B.3.2. Posicióname en el mapa**

La ubicación del usuario en el mapa se muestra sobre una imagen en dos dimensiones del mapa de la primera planta de la Facultad de Informática. El usuario aparece representado mediante un punto rojo en el mapa.

### **B.3.3. Aplicación real**

En esta actividad, se muestra la vista de la cámara en la pantalla del dispositivo.

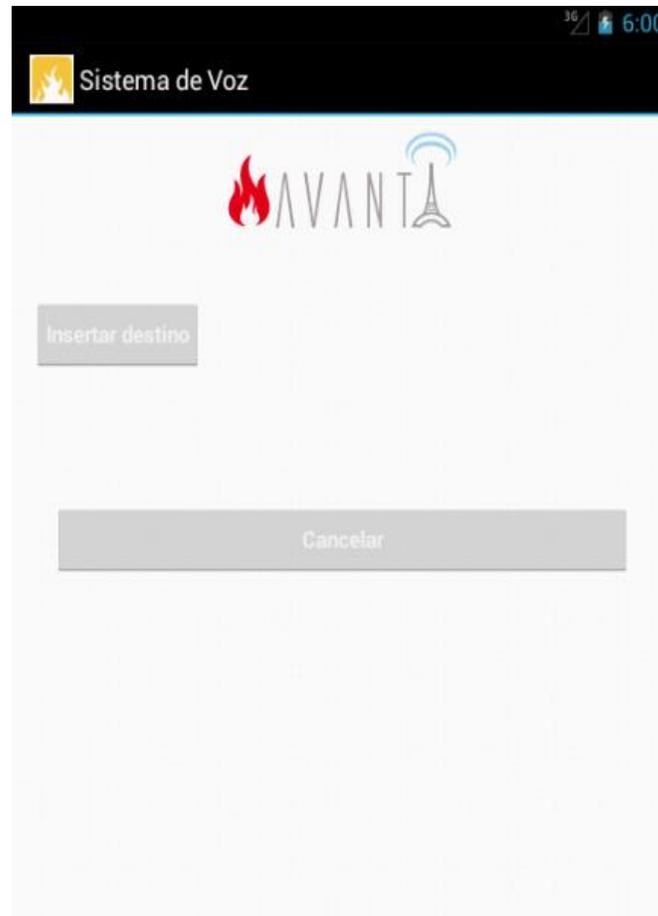
### **B.3.4. Prueba del acelerómetro**

En esta pantalla, se muestran los valores del acelerómetro para los ejes X, Y, Z, así como el giro sobre dichos ejes. También se indicará si el giro es válido o no. El giro será válido si el teléfono se encuentra de forma horizontal con la cámara apuntando hacia el frente, en ese caso, se mostrarán los valores normalizados del acelerómetro en cada uno de los tres ejes.

### **B.3.5. Indicar ruta hacia el destino**

Esta es la pantalla que contiene la principal funcionalidad de este proyecto. El usuario principalmente tiene dos opciones:

- Insertar destino
- Cancelar



*Figura 21: Pantalla Indicar Ruta hacia el destino*

Si el usuario selecciona la opción de *Cancelar*, se volverá al menú principal de la Aplicación.

Si por el contrario el usuario selecciona *Insertar destino*, se abre un cuadro de Reconocimiento de Voz, momento en el que el usuario deberá insertar el destino deseado. Una vez se ha reconocido el destino, aparece escrito en la pantalla junto al botón insertar destino.

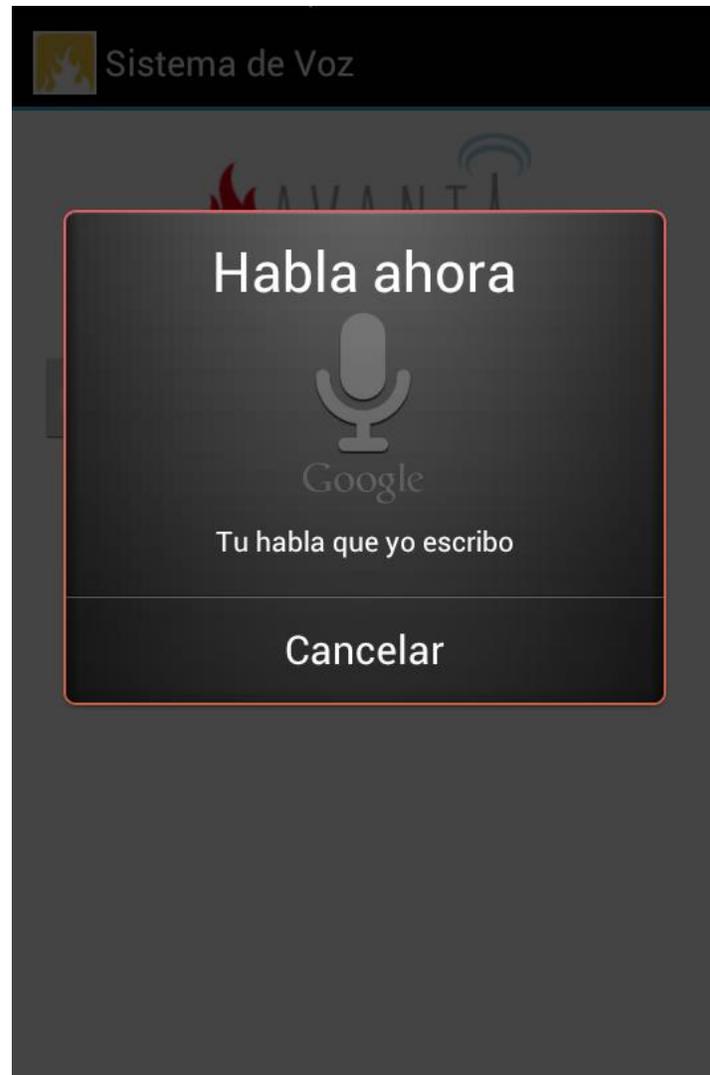


Figura 22: Pantalla del Reconocimiento de Voz

Si el destino ha sido insertado correctamente, el sistema comprueba que esté activado el Wi-Fi en el terminal. Es necesario tenerlo activado, pues necesitamos saber la posición actual en la que se encuentra el usuario. Para ello, si el Wi-Fi no está activado, se abrirá un nuevo cuadro de diálogo, dónde solicitará activar el Wi-Fi, el usuario puede *Aceptar*, caso en el que se activará, o puede *Cancelar*, lo que provocará volver al menú principal.

Una vez hemos activado el Wi-Fi, el sistema necesita obtener la orientación inicial en la que se encuentra el usuario para poder indicar correctamente el camino que tiene que seguir. Para ello, necesita que el móvil se encuentre en posición horizontal con la cámara mirando hacia el frente. Para lograr esto, mostrará un nuevo cuadro de diálogo donde avisará al usuario, este puede *Aceptar* ó *Cancelar* y volver al menú principal. Si el usuario acepta sin haber puesto el terminal en la posición requerida el cuadro de diálogo seguirá apareciendo por pantalla.

Realizados todos estos pasos, el sistema ha conseguido recopilar todos los datos necesarios para poder realizar el cálculo de la ruta. Generada la secuencia de instrucciones, el teléfono móvil las recibe, es cuando el terminal comienza la reproducción de voz de toda la secuencia de instrucciones generadas. Para mayor comodidad se han escrito también por pantalla la lista de instrucciones que el usuario debería seguir.



Figura 23: Pantalla con la Ruta Calculada

### B.3.6. Cerrar sesión

Finalmente, la última opción realiza la desconexión del usuario de la aplicación, regresando a la pantalla principal de la Aplicación.

---

# BIBLIOGRAFÍA

- [1] E. López Mañas, F. J. Moreno y J. Plá Herrero, AVANTI: Sistema de Asistencia a la Evacuación de Incendios, Madrid: Facultad de Informática, Universidad Complutense de Madrid, 2010.
- [2] R. S. Pressman, Ingeniería del Software “Un Enfoque Práctico”, Sexta Edición, McGraw-Hill, 2005.
- [3] J. Seitz, L. Patiño-Studencka, B. Schi, S. Haimerl, J. Gutiérrez Boronat, S. Meyer y J. Thi, “Sensor Data Fusion for Pedestrian Navigation Using WLAN and INS”..
- [4] P. Bahl y V. N. Padmanabhan, “RADAR: an in-building RF-based user location and tracking system”, in Proceedings of 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00), vol. 2, pp. 775–784, Tel Aviv, Israel, March, 2000.
- [5] Y. Chen y H. Kobayashi, “Signal strength based indoor geolocation”, in Proceedings of the IEEE International Conference on Communications (ICC '02), vol. 1, pp. 436–439, New York, NY, USA, April-May 2002.
- [6] S. Feldmann, K. Kyamakya y Z. Lue, An Indoor Bluetooth-Based Positioning System: Concept, Implementation and Experimental Evaluation, in: International Conference on Wireless Networks, 2003.
- [7] I. Cortázar y P. Sanz, “Localización en entornos Wi-Fi”, Comunicaciones de Telefónica I+D. Telefónica I+D, 2005.
- [8] E. Núñez Gómez, Modelado de Evacuación de Multitudes Mediante Agentes y Transcripción de Comportamientos, Madrid: Facultad de Informática, Universidad Complutense de Madrid, 2011.
- [9] Reiter, E., Dale, R. (2000): “Building Applied Natural Language Generation Systems”, CUP..

- [10] Cahill, L. et al. (2001): "The RAGS Reference Manual". Tech. Rep. ITRI-01-07, ITRI, University of Brighton..
- [11] Gamma, E. et al. (1995): "Design Patterns: Elements of Reusable Object-Oriented Software", Addison Wesley..
- [12] Johnson, R. E., Foote, B. (1988): "Designing Reusable Classes". J. of Object-Oriented Programming, vol. 1, 2..
- [13] P. Gervás, SurReal: a Surface Realization module. Natural Interaction based on Language Group Technical Report, Universidad Complutense de Madrid, Spain, 2006.
- [14] M. Elhadad, Technical Report CUCS-038-91. Columbia University, 1993.
- [15] M. Elhadad y J. Robin, Technical Report 96-03. Department of Computer Science, Ben Gurion University, 1996.
- [16] P. Gervás, TAP: a Text Arranging Pipeline. Natural Interaction based on Language Group Technical Report, Universidad Complutense de Madrid, Spain, 2007.
- [17] P. Gervás y R. Hervás, Evolutionary and case-based approaches to REG: NIL-UCMEvoTAP, NIL-UCM-ValuesCBR and NIL-UCMEvoCBR. In Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009), pages 187–188, Athens, Greece, March, Association for Computational Linguistics, 2009.
- [18] P. Gervás, UCM Submission to the Surface Realization Challenge. Surface Realization Challenge. Challenges 2011 Session at 13th European Workshop on Natural Language Generation (ENLG 2011), 2011.
- [19] F. Evennou y F. Marx, "Advanced Integration of WiFi and Inertial Navigation Systems for Indoor Mobile Positioning"., Hindawi Publishing Corporation, EURASIP Journal on Applied Signal Processing, Volume 2006, Article ID 86706, Pages 1–11..
- [20] H. Wang, H. Lenz, A. Szabo, J. Bamberger y U. D. Hanebeck, "WLAN-Based Pedestrian Tracking Using Particle Filters and Low-Cost MEMS Sensors", 4th Workshop on positioning, navigation and communication, Hannover, Germany, 2007 (WPNC'07).
- [21] I. Sommerville, "Ingeniería Del Software". 7ª edición, Addison-Wesley, 2005.
- [22] B. W. Boehm, Software Risk Management: "Principles and Practices", IEEE Software, January, 1991.
- [23] Software Quality Assurance Subcommittee of the Nuclear Weapons Complex Quality Managers under the United States Department of Energy, SQAS21.01.00-1999.