
**GUIDIN: Generador interactivo de
instrucciones de guía para usuarios con
necesidades especiales sobre plataformas
móviles**



Trabajo de fin de grado

**Jorge Almendros Fresnillo
Luis Hernández Donadeu**

**Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid**

Junio 2015

Documento maquetado con T_EX!S v.1.0.

GUIDIN: Generador interactivo de instrucciones de guía para usuarios con necesidades especiales sobre plataformas móviles

Memoria que presenta para el Trabajo Fin de Grado

**Jorge Almendros Fresnillo
Luis Hernández Donadeu**

Dirigida por:

**Gonzalo Méndez Pozo
Raquel Hervás Ballesteros**

**Departamento de Ingeniería del Software e Inteligencia
Artificial
Facultad de Informática
Universidad Complutense de Madrid**

Junio 2015

Copyright © Jorge Almedros Fresnillo
Luis Hernández Donadeu

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.

Jorge Almendros Fresnillo

Luis Hernández Donadeu

Agradecimientos

La elaboración de este proyecto a requerido de mucho esfuerzo y dedicación por parte de los dos integrantes del grupo. Pero al final ha sido posible sacarlo adelante, en parte gracias a las aportaciones, ánimos y empuje que nos han dado amigos y familiares, y por ello queremos aprovechar estas líneas para agradecerles el esfuerzo.

Por otro lado, también se merecen una mención en este apartado nuestros tutores, Gonzalo Méndez Pozo y Raquel Hervás Ballesteros, que nos han ayudado siempre que lo hemos necesitado, nos han guiado, soportado, aguantado retrasos (¡y qué retrasos!) y metido presión cuando ha hecho falta.

Gracias a todos.

Resumen

Dada la situación actual en la que nos encontramos en el mundo del posicionamiento wi-fi, se nos plantea el reto de realizar una aplicación que sea capaz de servir de guía a personas con discapacidad, con la complejidad añadida de guiarlas en el interior de edificios.

GuiDIn (Guía para personas con Discapacidad en Interiores) será el nombre que reciba nuestra aplicación. Consiste en posicionamiento en interiores basado en WPS. La aplicación es capaz de guiar a una persona, con discapacidad o no, dentro de la Facultad de Informática de la Universidad Complutense de Madrid. El dispositivo sabe en todo momento donde se encuentra el usuario y le indica el camino que debe seguir para llegar al lugar indicado por éste.

El posicionamiento se ha realizado mediante triangulación de intensidades de señal wi-fi, aunque se han valorado otras opciones, al final se concluyó que esta era la mejor opción. Se llevó a cabo un estudio sobre aplicaciones similares y no se encontró ninguna aplicación que estuviese orientada a personas con discapacidad, por lo que la idea de este proyecto es original e innovadora, siendo además el posicionamiento en interiores un tema también actual y sobre el que merece la pena investigar.

La plataforma sobre la que se ha desarrollado esta aplicación a sido Android. Se han tenido en cuenta otras posibilidades pero se descartaron por diversos motivos que se explican mas adelante. Entre estos motivos tenemos la facilidad de acceder a los sensores del dispositivo, además de la facilidad de acceso a las herramientas para su desarrollo.

Para la elaboración de este proyecto se ha llevado a cabo una gestión del proyecto que incluye gestión de requisitos y de riesgos. De igual modo se realizó una planificación del proyecto para poder cumplir con los tiempos establecidos para la realización el mismo.

El proyecto está dividido en dos grandes secciones, la parte del cliente que consiste en una aplicación Android que se ejecuta sobre un dispositivo móvil con este sistema operativo, y por otro lado la parte del servidor, un ordenador que se encarga de realizar todos los cálculos necesarios para la correcta ejecución de la aplicación que además libera de carga de trabajo al dispositivo.

La parte del cliente como se acaba de explicar consiste en una aplicación que se ejecuta sobre un dispositivo móvil que cuenta con las adaptaciones necesarias para que una persona con discapacidad visual o auditiva sea capaz de usarla.

El servidor se encarga de realizar la conexión de la aplicación con la base de datos con la que cuenta esta aplicación y además realiza los cálculos de las rutas, adaptando estas también a la discapacidad que le indique el cliente.

Palabras clave

- Posicionamiento en interiores
- Tecnología wi-fi
- Guía para personas con discapacidad
- Android

Abstract

Given the current situation in which the wireless positioning world is, we face the challenge of making an application that is capable to guide people with disabilities, with the complexity of guiding them inside buildings.

GuiDIn (Guía para personas con Discapacidad en Interiores) is the name of our application. It consists on indoor positioning based on WPS. The application is able to guide a person with or without disabilities, through the Facultad de Informática of Universidad Complutense de Madrid. The device always knows where the user is and shows the way forward to reach the place indicated by the user.

The positioning is performed using triangulation signal of several wi-fi intensities, other options were valued, but finally we concluded that this was the best option. A research was made on similar applications and any of the applications that we could find were oriented to people with disabilities. That is why the idea of this project is original and innovative, and also the indoor positioning is a current subject on which is worth investigating.

The platform on which this application has been developed is Android. Other possibilities were taken into account but they were discarded for various reasons explained in this text. Among these reasons we have the easiness to access the sensors of the device, plus the ease of access to the tools for their development.

For the development of this project has been carried out the project management including requirements management and risk management. Likewise, a project planning was made to satisfy the established times for performing the task.

The project is divided into two major sections, the client side consisting of an Android application running on a mobile device with the same operating system, and on the other side from the server, a computer that is responsible for making all calculations required for the proper execution of the application that also frees the device workload.

The client side as explained above consists of an application running on a mobile device that has the necessary modifications for a person with visual or hearing disability are able to use it.

The server handles the connection of the application with the database

that this application has and also performs the routes calculations, adapting them to the disability that the client sends.

Keywords

- Indoor positioning
- Wi-fi technology
- Guide to people with disabilities
- Android

Índice

Autorización	V
Agradecimientos	VII
Resumen	IX
Abstract	XIII
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
1.3. Estructura del documento	2
2. Introduction	5
2.1. Motivation	5
2.2. Objectives	5
2.3. Document structure	6
3. Estado de la cuestión	9
3.1. Sistemas de posicionamiento	9
3.1.1. Posicionamiento mediante triangulación	9
3.1.2. Posicionamiento GPS	9
3.1.3. Posicionamiento bluetooth	11
3.1.4. Posicionamiento wi-fi	11
3.2. Aplicaciones de guía en interiores	12
3.2.1. Google indoor maps	12
3.2.2. Infsoft	13
3.2.3. Redpin	14
3.3. Trabajos previos	14
3.3.1. Avanti: Sistema de asistencia a la evacuación de incendios 2009-2010	14
3.3.2. Sistema de guía por voz en interiores 2012-2013	15

3.3.3. Generador interactivo de instrucciones de guía sobre plataformas móviles 2013-2014	16
4. Tecnologías utilizadas	17
4.1. Android	17
4.2. Sistemas de voz	18
4.2.1. Reconocimiento de voz	19
4.2.2. Reproducción de voz	20
5. Gestión del proyecto	21
5.1. Análisis de requisitos	21
5.1.1. Encuestas con usuarios finales	21
5.1.2. Requisitos	22
5.2. Gestión de riesgos	23
5.2.1. Análisis de los riesgos	24
5.2.2. Estudio de los riesgos	25
5.3. Planificación del proyecto	28
6. Arquitectura de GUIDIN	31
6.1. Características del proyecto	31
6.2. Comunicación	32
7. Diseño del cliente	35
7.1. Introducción	35
7.2. Descripción de las clases utilizadas	36
7.3. Diagramas de casos de uso	41
7.4. Diagramas de secuencia	42
7.5. Manual de usuario	45
7.5.1. Pantalla de inicio de la aplicación	45
7.5.2. Pantalla de registro de la aplicación	45
7.5.3. Menú principal	45
7.5.4. Menú para Usuarios	46
7.5.5. Menú para Administradores	48
8. Diseño del servidor	51
8.1. Introducción	51
8.2. Descripción de la Base de Datos	52
8.3. Ficheros PHP	53
8.4. Aplicación Java	55
8.4.1. XML	55
8.4.2. Java	58

9. Conclusiones y trabajo futuro	61
9.1. Descripción de los resultados obtenidos	61
9.2. Conclusiones	62
9.3. Trabajo Futuro	62
10. Conclusions and future work	63
10.1. Description of results	63
10.2. Conclusions	63
10.3. future work	64
11. Trabajo individual	65
11.1. Jorge Almendros Fresnillo	65
11.2. Luis Hernández Donadeu	66
A. Manual de instalación	69
A.1. Requisitos	69
A.2. Instalación en Android Studio	69
A.3. Instalación en un dispositivo móvil	70
A.4. Instalación del servidor	70
B. Extensión de la aplicación	71
B.1. Conexión cliente servidor	71
B.2. Crear base de datos	72
B.3. Enlace cliente - base de datos	72
B.4. Medir intensidades wi-fi	73
B.5. Creación de los XML relativos a la estructura del edificio	73
B.6. Establecer destinos	73
B.7. Compilar y ejecutar la aplicación	74
B.8. Añadir discapacidades	74
C. Formularios pasados a personas con discapacidades	75
Bibliografía	79
Lista de acrónimos	81

Índice de figuras

3.1. Posicionamiento por triangulación.	10
3.2. Google indoor maps	13
3.3. Insoft1	14
3.4. Insoft2	14
3.5. Redpin	15
6.1. Estructura	32
6.2. Comunicación Cliente con BBDD	33
6.3. Comunicación Cliente con la aplicación Java	33
7.1. Paquete HTTP.	37
7.2. Paquete Sockets.	37
7.3. Paquete WPS.	38
7.4. Paquete rutas.	38
7.5. Clases principales del cliente.	40
7.6. Caso de uso usuario.	41
7.7. Caso de uso administrador.	42
7.8. Diagrama de secuencia Login.	43
7.9. Diagrama de secuencia Medir intensidades.	43
7.10. Diagrama de secuencia Posición en el mapa.	44
7.11. Diagrama de secuencia Indicar destino.	44
7.12. Inicio de la aplicación.	46
7.13. Pantalla de registro.	46
7.14. Menú de usuario	47
7.15. Pantalla de indicar destino	47
7.16. Pantalla de posición en el mapa.	48
7.17. Menú de administradores.	48
7.18. Pantalla de actividad wi-fi	49
7.19. Pantalla de detalle de red wi-fi	49
7.20. Pantalla de acelerómetros.	50
7.21. Pantalla de medir intensidades.	50

8.1. Comunicacion PHP con BBDD	51
8.2. Tablas de BBDD	52
8.3. Nueva posición	54
8.4. Dame posiciones	54
8.5. Login	55
8.6. Diagrama de clases Java	58
8.7. Secuencia Calcula Ruta	60
A.1. Instalación en el dispositivo móvil	70
B.1. Lanzamiento de la ejecución en consola	72
C.1. Formulario para personas con discapacidad visual.	76
C.2. Formulario para personas con discapacidad auditiva.	77

Índice de Tablas

5.1. SQAS-SEI	24
5.2. Criticidad de los riesgos encontrados	25

Capítulo 1

Introducción

1.1. Motivación

Hoy en día prácticamente todo el mundo posee un teléfono inteligente, o *smartphone*, y cada vez son más utilizados para tareas que nos facilitan la vida o nos entretienen. Entre estas tareas encontramos la de guiar a las personas, que nos da la posibilidad de llegar a multitud de sitios de forma rápida y precisa sin necesidad de estar buscando tediosamente como llegar a nuestro destino. Para ello se empezó a utilizar el GPS (*Global Positioning System*), para guiarnos hasta lugares tan distantes entre sí como podamos imaginar. Sin embargo, el problema ante el cual nos encontramos ahora es el de navegar dentro de edificios, donde el GPS no es capaz de guiarnos con precisión.

Actualmente la mayor parte de los dispositivos móviles que lleva la gente siempre encima posee tecnología inalámbrica wi-fi para conectarse a Internet. Sin embargo, esta tecnología no sólo sirve para esta función sino que también puede resultar bastante útil para posicionar un dispositivo a través de varias intensidades de señales wi-fi.

Aparte de lo ya mencionado, una de nuestras grandes motivaciones es la de poder hacer llegar esta tecnología a personas con discapacidad, facilitándoles la vida en todo lo que nos sea posible. Nuestra aplicación estará centrada en ayudar a personas con discapacidad visual y auditiva. Por lo tanto, nuestra problemática además del posicionamiento en interiores es la de conseguir que estas personas con las discapacidades mencionadas sean capaces de utilizar la aplicación de la mejor manera posible. Para ello se ha realizado una investigación relacionada con este tema y se han llevado a cabo encuestas para poder dar el mejor servicio posible.

1.2. Objetivos

El principal objetivo de este proyecto es mejorar lo ya conseguido en los proyectos anteriores de los que parte este y además conseguir que una persona con discapacidad visual o auditiva sea capaz de usar la aplicación sin ningún problema.

Como ya se ha mencionado anteriormente este proyecto surge como la evolución de otros proyectos realizados anteriormente en la Facultad de Informática de la Universidad Complutense de Madrid. Las aplicaciones anteriores son capaces de posicionar a un individuo dentro de la facultad siempre que se encuentre en la planta primera o la segunda y de guiarle hasta una serie muy limitada de destinos.

Nuestro primer objetivo es por lo tanto, conseguir localizar a un individuo en cualquier parte de la facultad. Para ello será necesario extender la aplicación ya existente, teniendo que rehacer gran parte de lo ya hecho y añadiendo nuevas funcionalidades.

El segundo gran objetivo de este proyecto es conseguir crear una aplicación accesible para personas con discapacidad visual o auditiva. Sufrir una de estas discapacidades no debe suponer un problema a la hora de usar la aplicación. Para ello se deben crear menús en los que se puedan elegir las opciones mediante voz y seleccionarlas de la misma forma. Además las rutas proporcionadas deben ajustarse a las necesidades de cada usuario.

Otra de las metas de este proyecto es mejorar el margen de error del posicionamiento conseguido en versiones anteriores de la aplicación. Para ello se modificarán los algoritmos destinados a este fin y se llevará a cabo un estudio acerca de los métodos de posicionamiento en interiores actuales.

Por otra parte, otro de nuestros propósitos con esta aplicación es conseguir que las instrucciones proporcionadas se den en un formato lo más natural posible, y de la forma más adecuada a cada usuario.

Por último se pretende aumentar considerablemente el número de destinos a los que se puede llegar, incluyendo despachos, aulas y cualquier otro tipo de estancia que podamos encontrar dentro del edificio propuesto.

1.3. Estructura del documento

Este documento se encuentra organizado en capítulos. En este Capítulo 1 en el que nos encontramos tenemos una pequeña introducción, otro apartado en el que se establecen los objetivos considerados en este proyecto y por último una breve descripción del contenido del documento. En el Capítulo 2 se repite el contenido del capítulo anterior pero en Inglés.

- El Capítulo 3 contiene el “Estado de la cuestión” en el que se habla de la situación actual del posicionamiento en interiores, de aplicaciones

similares ya existentes y de los trabajos previos de los que parte este trabajo.

- El Capítulo 4 explica las tecnologías utilizadas para la implementación del proyecto.
- En el Capítulo 5 se trata la gestión del proyecto, se hace un análisis de requisitos, se realiza la gestión de riesgos y se establece la planificación del proyecto.
- En el Capítulo 6 se explica la arquitectura que se ha seguido para el proyecto. Se detalla la estructura y la comunicación entre los módulos necesarios.
- El proyecto está dividido en dos grandes partes, y para cada una de ellas se ha dedicado un capítulo en este documento. Se corresponden con el Capítulo 7 y el Capítulo 8.
- El Capítulo 9 habla sobre los resultados, las conclusiones a las que se han llegado y nos da ideas de posibles trabajos futuros que pueden surgir a partir de GuiDIn.
- El Capítulo 10 cuenta lo mismo que el capítulo anterior pero en Inglés.
- Por último en el Capítulo 11 se explica de forma individual qué parte ha hecho cada uno de los miembros que ha realizado este proyecto.

Además se incluyen tres apéndices adicionales. En el primero (apéndice A) se proporciona un manual de instalación de la aplicación, del servidor y del proyecto Android. En el segundo (apéndice B) se explican los pasos a seguir para poder realizar una extensión de la aplicación. Por último, el apéndice final (apéndice C) consiste en una muestra de unos formularios creados para la realización de este proyecto.

Como final del documento se aporta una bibliografía y las referencias utilizadas para la realización tanto del documento como de la aplicación

Capítulo 2

Introduction

2.1. Motivation

Nowadays all the people have a smartphone and more often they are used for tasks that make easy our life or entertain us. Among these tasks one of them is guiding people, which gives us the ability to reach many sites quickly and accurately without the necessity of check how to get our destination. For that, the GPS was started to use to guide us to places as far apart as we can imagine. However, the issue with which we are now is to guide people inside buildings where the GPS is not able to accurately guide us.

Currently, most of the smartphones that people have, come with wi-fi wireless technology to connect to the Internet. However, this technology is not just for this feature but also can be quite useful for positioning a device through various signal intensities wi-fi.

Apart from the above, one of our greatest motivation is the possibility to bring this technology to people with disabilities making easy the life all that we can. Our application will focus on helping people with visual and hearing disabilities. Therefore, our problems besides the indoor positioning, is to get that these people with disabilities mentioned are able to use the application in the best way possible. For this we have conducted research about this disabilities and have conducted surveys to provide the best service possible.

2.2. Objectives

The main objective of this project is to improve the results in previous projects and also get that person with visual or hearing disabilities be able to use the application without any problem.

As mentioned above, this project is the evolution of other previous projects undertaken at the Facultad de Informática, Universidad Complutense de Madrid. The previous applications are capable of positioning an indivi-

dual within the building provided it is in the first or second floor and guide you to a very limited number of destinations.

Our first objective is to locate anyone anywhere in the faculty. This will require extending the existing application, having to redo much of it done and adding new features.

The second objective of this project is to create an accessible application for people with visual or hearing disability. Having one of these disabilities should not be a problem when using the application. To do this we must create menus where they can choose options through voice and select them in the same way. Besides the routes provided must meet the needs of each user.

Other of the goals in this project is to improve the margin of error of positioning achieved in previous versions of the application. To do this we will change the positioning algorithms and will carry out a study on the current interiors positioning methods.

Moreover, one of our goals of this application is to get the instructions that are given in the most natural possible format, and in the most appropriate way to each user.

As last objective we aim to significantly increase the number of destinations that can be reached, including offices, classrooms and any other kind of place that we can find in the proposed building.

2.3. Document structure

This document is organized into chapters. In this Chapter 2 we have a short introduction, other section in which the objectives are established and finally a brief description of the document structure.

- Chapter 3 contains the “ State of the art ” in which we talk about the current situation of indoor positioning of existing similar applications and previous jobs than part this work.
- Chapter 4 explains the technologies used for the project implementation.
- Chapter 5 is about project management, requirements analysis, risk management and the project planning is established.
- Chapter 6 explains the architecture that has been followed for the project. Is detailed the structure and the necessary communication between modules.
- The project is divided into two parts, and each has devoted a chapter in this document. They correspond to the Chapter 7 and Chapter 8.

-
- Chapter 9 discusses the results, the conclusions that have been reached and gives us ideas for possible future work that may arise from GuiDIn.
 - Chapter 10 tells the same than the last Chapter but in English.
 - Chapter 11 explains individually how much has each of its members has made in this project.

Additionally, three additional appendices are included. In the first (Appendix A) a manual installation of the application server and the Android project is provided. In the second (Appendix B) explains the steps to follow to make an extension of the application. Finally, the end appendix (Appendix C) is a sample of some forms created for this project.

As a final document a bibliography and references used to make both the document and the application is provided.

Capítulo 3

Estado de la cuestión

Este capítulo trata sobre los diferentes sistemas de localización que existen actualmente y se hace una valoración acerca de por qué se han utilizado para desarrollar la aplicación o por qué no. El mayor problema que nos hemos encontrado ha sido la falta de precisión, así como la falta de medios que es necesario desplegar para algunos tipos de posicionamiento. También se hace un estudio de algunas aplicaciones similares ya existentes en el mercado, además de una pequeña descripción de los proyectos anteriores de los que parte este mismo proyecto.

3.1. Sistemas de posicionamiento

3.1.1. Posicionamiento mediante triangulación

La triangulación (Wikipedia, c) es una técnica de posicionamiento que necesita al menos tres señales. El origen de estas señales es conocido y lo único que hace falta es conocer la distancia a cada de ellas. Con estos datos se puede trazar una esfera alrededor de las señales con radio la distancia al dispositivo receptor de la señal, de esta forma las esferas generadas se cortarán unas a otras produciendo un único punto donde coincidan todos los cortes siendo ese el punto donde se encontrará el dispositivo que se quiera localizar. Si se desea conocer la altitud es necesario que como mínimo existan cuatro emisores de señales. Este sistema es aplicable a cualquier tipo de posicionamiento, ya sea GPS, bluetooth, wi-fi, etc.

La Figura 3.1 muestra una aproximación de cómo funciona este tipo de posicionamiento

3.1.2. Posicionamiento GPS

El sistema de posicionamiento por GPS (*Global Positioning System*) es el más extendido y conocido mundialmente. El GPS (Wikipedia, b) es un ob-

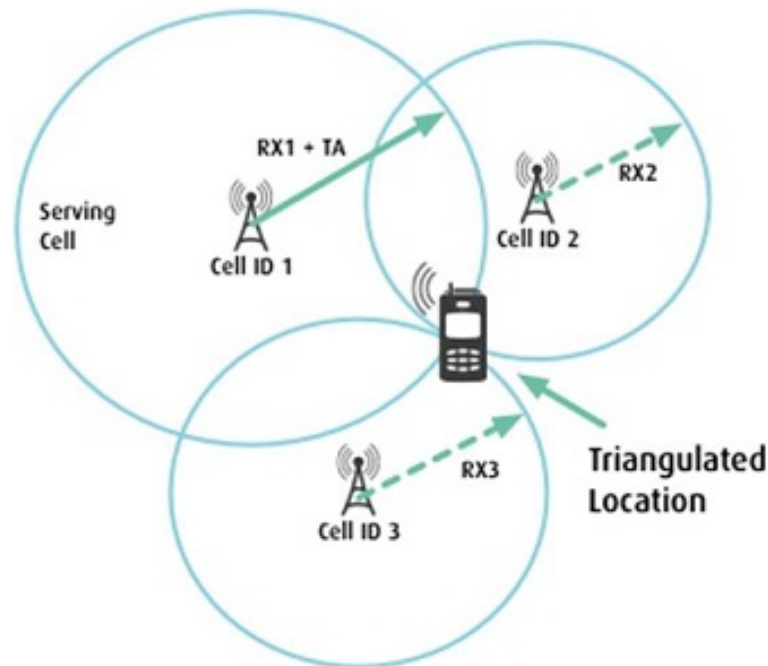


Figura 3.1: Posicionamiento por triangulación.

jeto que permite a una persona determinar en todo el mundo la posición de un objeto, una persona o un vehículo con una precisión hasta de centímetros (si se utiliza GPS diferencial), aunque lo habitual son unos pocos metros de precisión. El sistema fue desarrollado, instalado y empleado por el Departamento de Defensa de los Estados Unidos. El sistema GPS está constituido por 24 satélites y utiliza la triangulación para determinar en todo el globo una posición con una precisión de más o menos metros.

El GPS funciona mediante una red de 24 satélites en órbita sobre el planeta tierra, a 20.200 km de altura, con trayectorias sincronizadas para cubrir toda la superficie de la Tierra. Cuando se desea determinar la posición, el receptor que se utiliza para ello localiza automáticamente como mínimo cuatro satélites de la red, de los que recibe unas señales indicando la identificación y la hora del reloj de cada uno de ellos. Con base en estas señales, el aparato sincroniza el reloj del GPS y calcula el tiempo que tardan en llegar las señales al equipo, y de tal modo mide la distancia al satélite mediante método de triangulación inversa, la cual se basa en determinar la distancia de cada satélite respecto al punto de medición. Conocidas las distancias, se determina fácilmente la propia posición relativa respecto a los satélites. También se consigue una exactitud extrema en el reloj del GPS, similar a la de los relojes atómicos que llevan a bordo cada uno de los satélites.

La antigua Unión Soviética construyó un sistema similar llamado GLO-NASS, ahora gestionado por la Federación Rusa. Actualmente la Unión Eu-

ropea está desarrollando su propio sistema de posicionamiento por satélite, denominado Galileo. A su vez, la República Popular China está implementando su propio sistema de navegación, el denominado Beidou.

Sin embargo, el gran problema ante el que se encuentra este tipo de posicionamiento son los obstáculos. El posicionamiento GPS funciona perfectamente cuando hay conexión directa con los satélites, pero en el momento que hay algún obstáculo de por medio, es muy probable que se pierda la conexión. Un claro ejemplo de esto es cuando un coche que se va guiando por satélite entra en un túnel, ya que en ese momento la conexión se pierde. La solución que intentan implementar estos aparatos es que intuyen donde debería estar el coche basándose en la velocidad y el camino que sigue la carretera, siendo un sistema poco fiable y no utilizable para guiar a personas.

3.1.3. Posicionamiento bluetooth

La conexión bluetooth es una red inalámbrica de área personal (WAPN), que permite la conexión y transferencia de datos entre varios dispositivos. La parte positiva de esta conexión la encontramos en el consumo, ya que ofrece consumos muy bajos tanto para los emisores como para los receptores. Por otro lado tenemos el alcance de estas redes. Incluso en sus últimas versiones (4.0) no supera los 10 metros por lo que sería necesario disponer de un número de balizas demasiado grande para cubrir toda la superficie que ocupa el edificio en el que se intenta hacer la navegación (Retana, 2010). De la misma forma la velocidad de transmisión también es un problema en este tipo de redes, ya que no se acercan a las velocidades que pueden ofrecer las conexiones wi-fi, por ejemplo.

3.1.4. Posicionamiento wi-fi

Las redes wi-fi son redes inalámbricas de área local (Wireless Local Area Network o WLAN). Existen diversos tipos de wi-fi, basado cada uno de ellos en un estándar IEEE 802.11 aprobado. Los estándares IEEE 802.11b, IEEE 802.11g e IEEE 802.11n disfrutaron de una aceptación internacional debido a que la banda de 2.4 GHz está disponible casi universalmente, con una velocidad de hasta 11 Mbit/s, 54 Mbit/s y 300 Mbit/s, respectivamente.

En la actualidad ya se maneja también el estándar IEEE 802.11ac, conocido como wi-fi 5, que opera en la banda de 5 GHz y que disfruta de una operatividad con canales relativamente limpios. La banda de 5 GHz ha sido recientemente habilitada y, además, no existen otras tecnologías (bluetooth, microondas, etc.) que la estén utilizando, por lo tanto existen muy pocas interferencias. Su alcance es algo menor que el de los estándares que trabajan a 2.4 GHz (aproximadamente un 10 por ciento), debido a que la frecuencia es mayor (a mayor frecuencia, menor alcance).

La idea de localización a través de señales wi-fi (Wikipedia, d) (Chen

and Kobayashi, 2002) consiste en que un dispositivo capta la señal wi-fi de un dispositivo de red. Una vez conocida su dirección MAC, el dispositivo se conecta a una base de datos en la que hay almacenadas diferentes coordenadas cartesianas asociadas a diferentes señales MAC (en caso de haberlas) y la intensidad que la señal wi-fi del dispositivo de red alcanza en ese punto. De este modo, se puede interpolar entre la intensidad captada por el dispositivo y la guardada en la base de datos para conocer la posición en la que se encuentra el dispositivo de red.

Este tipo de posicionamiento permite localizar un dispositivo con un margen de error de un metro como máximo, lo cual es considerado un margen bastante pequeño y aceptable para la realización de nuestro proyecto. Hay que tener en cuenta que se pretende guiar a personas con discapacidad visual y la precisión es un factor muy importante, ya que habrá que pasar por zonas con cierta estrechez tales como puertas o ascensores.

Estos hechos nos han llevado a tomar la decisión de que el posicionamiento por wi-fi haya sido el elegido para desarrollar nuestra aplicación. Este modo de posicionamiento nos aporta precisión, velocidad de transmisión, relativa facilidad de implementación, y suficiente alcance de las señales wi-fi.

3.2. Aplicaciones de guía en interiores

A continuación se explicarán como funcionan algunas aplicaciones similares, que también intentan realizar guía en interiores mediante intensidades de señal wi-fi.

3.2.1. Google indoor maps

Google indoor maps (Google, 2014) es un proyecto de Google con el que se pretende ampliar la funcionalidad de la aplicación Google maps. Esta aplicación permite visualizar mapas de prácticamente todo el mundo, incluso con imágenes reales. Sin embargo hasta hace poco tiempo no se ha incluido la posibilidad de visualizar el interior de las estructuras de los edificios, en concreto edificios públicos como aeropuertos, centros comerciales, hospitales, etc. de los cuales podemos observar incluso las diferentes plantas de las que esté compuesto el edificio. Como se puede observar en la Figura 3.2 la interfaz que ofrece esta aplicación es muy similar a la aplicación de exteriores. Para localizar el dispositivo dentro del edificio utilizan intensidades de señales wi-fi.

Esta aplicación está todavía en desarrollo, aunque ya está disponible para su descarga y uso. La incorporación de mapas se hace mayormente por los propios usuarios que se tienen que encargar de localizar donde se encuentra el

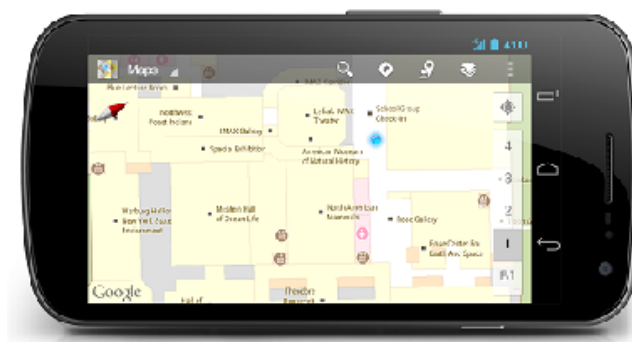


Figura 3.2: Google indoor maps

lugar donde se encuentra el edificio, subir el mapa de cada planta que quieran incluir y por último alinearlos para que encaje con los mapas de Google.

Una de las ventajas de esta aplicación es que al ser de Google, una de las empresas más grandes del mundo, la visibilidad y el alcance de cualquier aplicación que utilice esta tecnología puede ser muy grande. Sin embargo, por lo que hemos podido investigar el posicionamiento no es muy preciso, utilizan el mismo sistema de posicionamiento que para los mapas en exteriores, ayudándose un poco de las intensidades de wi-fi, pero estas son solo recolectadas gracias a la información que envían sus teléfonos sin que los usuarios se den cuenta, y por lo tanto la precisión no es muy buena.

3.2.2. Infsoft

Esta empresa alemana (Infsoft, 2015) se dedica tanto al posicionamiento en interiores como en exteriores. Para el posicionamiento en interiores, que es el que nos atañe, utilizan todo tipo de sensores disponibles en el dispositivo móvil, como GSM, 3G/4G (LTE), wi-fi, campos magnéticos, presión del aire, barómetros, acelerómetros, giroscopios, bluetooth y GPS. Con todo esto son capaces de dar una precisión de menos de un metro. Proporcionan además información de la ruta por texto además de mostrarla en un mapa. También han implementado un sistema de realidad aumentada que muestra las indicaciones sobre las imágenes que muestra la cámara. Las Figuras 3.3 y 3.4 muestran esta aplicación en funcionamiento.

Además, proporcionan herramientas para subir y editar mapas, y también para calibrar las posiciones. Trabajan sobre todo tipo de plataformas móviles, ya sea Android, IOS, Windows Phone o Blackberry por lo que pueden llegar a un gran público.



Figura 3.3: Insoft1



Figura 3.4: Insoft2

3.2.3. Redpin

Redpin (Redpin, 2008) es un sistema *open source* que fue desarrollado con la meta de proporcionar precisión a nivel de habitación. El sistema es capaz de identificar el que habitación se encuentra un dispositivo pero no de indicar el punto exacto.

Su sistema se basa en “fingerprints” o huellas que asignan a cada zona. De cada una de ellas recolectan cierta información como la intensidad wi-fi, con lo que luego son capaces de indicar en que zona se encuentra el dispositivo. La aplicación posee clientes para Android e IOS, pero su gran ventaja es que desde su página oficial se puede descargar todo el código para poder utilizarlo. La figura 3.5 muestra una de las pantallas de esta aplicación.

3.3. Trabajos previos

Como ya se ha mencionado anteriormente este proyecto surge como una evolución de otros proyectos ya desarrollados en la Facultad de Informática. Lo que hace diferente a GuiDIn de los trabajos previos es la adaptación a las personas con discapacidad.

Este proyecto, al igual que los anteriores, se encuentra englobado en el proyecto MILES (TIN2009-14659-C03) del departamento de Ingeniería del Software e Inteligencia Artificial de la Facultad de Informática de la Universidad Complutense de Madrid.

Gran parte del trabajo previo ha sido de mucha utilidad para el desarrollo de nuestra aplicación. A continuación se detallan los proyectos llevados a cabo en años anteriores de los cuales parte GuiDIn.

3.3.1. Avanti: Sistema de asistencia a la evacuación de incendios 2009-2010

Avanti fue el proyecto original del cual han partido todos los posteriores, incluido este mismo. Fue un proyecto de Sistemas Informáticos llevado a cabo



Figura 3.5: Redpin

por Enrique Lopez Mañas, Francisco Javier Moreno y Javier Plá Herrero (E. López Mañas, F. J. Moreno y J. Plá Herrero, 2010).

Dicho proyecto se planteó como un sistema capaz de guiar a las personas durante un simulacro de incendio. Desde este proyecto se utilizó el posicionamiento mediante wi-fi para realizar la guía en interiores, porque como ya se ha demostrado es la opción más viable. Además de esto se emplearon los acelerómetros del dispositivo para estimar mejor la posición del usuario, y poder posicionar al usuario incluso en zonas donde la señal wi-fi apenas llega.

Un gran atractivo de esta aplicación era el uso que se hacía de la realidad aumentada, para que a través de la cámara del dispositivo se pudiesen mostrar unos fuegos simulados sobre la propia Facultad de Informática

3.3.2. Sistema de guía por voz en interiores 2012-2013

Trabajo de Fin de Grado realizado por Mariana Martín- Calderín de la Villa (Mariana Martín-Calderín de la Villa, 2013). Este proyecto se centró en la guía en interiores. Se utilizó el mismo tipo de posicionamiento por wi-fi, pero se mejoraron las ayudas al posicionamiento a través del acelerómetro y la brújula del dispositivo y además se adaptó a las nuevas versiones de Android.

También se incluyeron sistemas de reconocimiento y reproducción de voz utilizados para pedir el destino y para reproducir las instrucciones generadas

por el calculo de la ruta.

3.3.3. Generador interactivo de instrucciones de guía sobre plataformas móviles 2013-2014

Proyecto de fin de carrera de Ingeniería Informática realizado por Víctor Gutiérrez Rodríguez, Juan Diego Lozano Martín y Víctor Manuel Pose Murga (Víctor Gutiérrez Rodríguez, Juan Diego Lozano Martín y Víctor Manuel Pose Murga, 2014). Este proyecto consistió básicamente en una mejora sobre el proyecto anterior. Se mejoró el posicionamiento ya que se implementó la posibilidad de realizar la guía a través de varias plantas del edificio, además de que se facilitó la posibilidad de llevar la aplicación a otros edificios.

Se mejoró también la generación de instrucciones, creando unas instrucciones más simples, intuitivas y eficaces.

Conclusiones

Los sistemas de posicionamiento analizados tienen sus ventajas y sus desventajas como se ha explicado en este capítulo, y teniendo esto en cuenta se ha elegido el sistema de posicionamiento wi-fi. Sin duda es el que mejor se adapta a las características del proyecto ya que no requiere ningún tipo de instalación de material para llevarlo a cabo, como por ejemplo requiere el posicionamiento por bluetooth. El posicionamiento GPS se descartó desde el principio ya que en interiores es totalmente inservible este tipo de posicionamiento. Sin embargo, sí se usa la triangulación, ya que lo que se hace con las intensidades de señal wi-fi es triangularlas para conseguir localizar el dispositivo.

Respecto a los proyectos anteriores se puede decir que han sido muy útiles sobre todo para realizar el posicionamiento en el cliente así como para tener una buena base de los algoritmos que calculan la ruta. Nuestro trabajo ha consistido en adaptar estas funcionalidades a personas con discapacidad, teniendo que realizar modificaciones en las rutas generadas teniendo en cuenta la discapacidad del usuario. De la misma forma la aplicación cliente requiere las adaptaciones necesarias para que un discapacitado visual o auditivo sea capaz de usarla sin problema, lo que conlleva un uso exhaustivo de las funciones de reproducción y reconocimiento de voz además de proporcionar las indicaciones de una forma u otra también en función de la discapacidad.

Capítulo 4

Tecnologías utilizadas

Este capítulo describe el sistema operativo Android (Google, 2015a), utilizado para el desarrollo de la aplicación, y explica por qué se ha usado como plataforma. Por otro lado también se abordará el tema de los sistemas de voz utilizados, ya sea para pasar texto a voz o viceversa.

4.1. Android

Este sistema operativo, basado en el kernel de Linux, fue diseñado desde un principio para dispositivos táctiles, tales como móviles y tabletas, aunque más recientemente se ha empezado a utilizar en relojes, televisiones y coches, para las que se han desarrollado las versiones específicas Android Wear, Android TV y Android auto respectivamente.

Inicialmente el proyecto fue desarrollado por la empresa Android Inc. la cual recibía ayudas de Google, y finalmente en 2005 se llevó a cabo la compra por parte de Google, actual propietaria del sistema operativo. La primera versión fue presentada en el 2007 pero no fue hasta finales del 2008 cuando se lanzó el primer teléfono al mercado, el HTC Dream (Google, 2015b).

Este sistema operativo posee una gran comunidad de desarrolladores creando aplicaciones para extender la funcionalidad de los dispositivos. Recientemente se ha superado el millón de aplicaciones, de las cuales dos tercios son gratuitas.

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (surface manager), un framework OpenCore, una base de datos relacional SQLite, una Interfaz de programación de API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C Bionic. El sistema operativo está compuesto por 12 millones de líneas de código,

incluyendo 3 millones de líneas de XML, 2,8 millones de líneas de lenguaje C, 2,1 millones de líneas de Java y 1,75 millones de líneas de C++.

Google liberó la mayoría del código de Android bajo la licencia Apache, una licencia libre y de código abierto. Esto facilita que diferentes fabricantes como Samsung, HTC, o Sony puedan usar el sistema operativo y modificarlo produciendo así una gran diversidad.

El acceso a todo tipo de sensores resulta mucho más sencillo en Android que en otros sistemas operativos para móviles como IOS de Apple o Windows Phone de Microsoft.

La publicación de aplicaciones en el Google Play resulta también bastante más fácil y económica que en el resto de plataformas, y hay una gran facilidad de llegar a un público bastante extenso.

Recientemente Google ha publicado un entorno de desarrollo integrado (IDE) que facilita la creación de aplicaciones, totalmente gratuito y que recibe actualizaciones constantemente. Android Studio que es el nombre que recibe este entorno y ha sido la plataforma que hemos usado para el desarrollo de nuestra aplicación y algunas de sus características son:

- Renderización en tiempo real.
- Consola de desarrollador: consejos de optimización, ayuda para la traducción, estadísticas de uso.
- Soporte para construcción basada en Gradle.
- Refactorización específica de Android y arreglos rápidos.
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones, y otros problemas.
- Plantillas para crear diseños comunes de Android y otros componentes.
- Soporte para programar aplicaciones para Android Wear.

Android es un sistema operativo puntero, que cuenta con la última tecnología tanto en software como en hardware, y sin duda es la mejor opción para llevar a cabo nuestra aplicación.

4.2. Sistemas de voz

Los sistemas de voz engloban dos posibilidades, reconocimiento del habla y reproducción de texto con voz. Este tema ha avanzado mucho en los últimos años, aunque es cierto que todavía no es perfecto y es fácil que se produzcan errores, pero hoy en día es una tecnología aceptable. Actualmente los sistemas de voz no están muy extendidos y se usan para tareas muy determinadas, pero es cierto que facilitan el uso de muchas aplicaciones a los usuarios.

Algunas de las situaciones donde es habitual ver el uso de esta tecnología es en la traducción, en los manos libres en el coche o para realizar búsquedas en la web.

Los dispositivos móviles recientemente están incluyendo estos sistemas ya que hay una dura competición entre las tres grandes compañías de desarrolladores de sistemas operativos para dispositivos inteligentes. Google ha añadido a su sistema operativo el llamado Google Now que aparte de dar información relacionada con la ubicación permite que en cualquier momento puedas decirle al dispositivo “Ok Google” seguido de una instrucción como por ejemplo “Qué tiempo hace en Madrid” o “Busca algo en Internet”. En Apple también han trabajado bastante en este aspecto y han desarrollado “Siri”, una aplicación bastante similar a Google Now y también a Cortana, siendo esta desarrollada por Microsoft e incluida en sus Windows Phone y también en su sistema operativo Windows a partir de la versión 10.

Para nuestra aplicación los sistemas de reconocimiento y reproducción de voz son de vital importancia ya que una persona con discapacidad visual tiene que utilizar la aplicación únicamente a través de las instrucciones que vaya leyendo el dispositivo y diciendo que es lo siguiente que quiere hacer. Para ello GuiDIn es capaz de ir leyendo al usuario todas las opciones disponibles en la aplicación y explicarle como llevar a cabo las tareas, ya sea registrarse, elegir una opción del menú o indicar un destino.

4.2.1. Reconocimiento de voz

El reconocimiento automático del habla (RAH) o reconocimiento automático de voz (Tamada, 2014) es una disciplina de la Inteligencia Artificial que tiene como objetivo permitir la comunicación hablada entre seres humanos y computadoras. El problema que se plantea en un sistema de este tipo es el de hacer cooperar un conjunto de informaciones que provienen de diversas fuentes de conocimiento (acústica, fonética, fonológica, léxica, sintáctica, semántica y pragmática), en presencia de ambigüedades, incertidumbres y errores inevitables para llegar a obtener una interpretación aceptable del mensaje acústico recibido.

Un sistema de reconocimiento de voz es una herramienta computacional capaz de procesar la señal de voz emitida por el ser humano y reconocer la información contenida en ésta, convirtiéndola en texto o emitiendo órdenes que actúan sobre un proceso. En su desarrollo intervienen diversas disciplinas, tales como la fisiología, la acústica, la lingüística, el procesamiento de señales, la inteligencia artificial y la ciencia de la computación. Estos sistemas existen desde hace bastante tiempo,

4.2.2. Reproducción de voz

Los sistemas de reproducción de voz, o TTS (Google, 2015c) por sus siglas en inglés (*Text To Speech*), permiten la transformación del texto a una voz artificial. Requisitos:

- Debe producir una voz sintética (artificial) que resulte natural y sea inteligible.
- La síntesis del habla ha de ser completamente automática, sin que se tenga que introducir ningún tipo de reajuste manual en ninguna parte del proceso.
- El texto introducido en el sistema ha de ser un texto arbitrario cualquiera, no puede estar amañado en ningún sentido.

Consta de tres fases:

- En la primera fase se realiza una representación lingüística simbólica, para ello se siguen tres procesos consecutivos:
 - Normalización del texto. Se convierte la totalidad del texto a una forma textual convencional. Esto afecta principalmente a las cifras, abreviaturas, etc. A la normalización del texto también se la denomina pre-procesado o tokenización.
 - Conversión fonética. Una vez normalizado el texto se asignan transcripciones fonéticas a cada palabra. El proceso de convertir las transcripciones fonéticas en palabras se denomina «conversión texto-fonema» (TTP en sus siglas en inglés de *text-to-phoneme*) o «conversión grafema-fonema» (GTP en sus siglas en inglés de *grapheme-to-phoneme*).
 - División prosódica. Se divide el texto en unidades prosódicas, tales como unidades sintagmáticas, proposiciones y frases.
- En la segunda fase, la que forma el sintetizador propiamente dicho, toma como entrada la representación lingüística simbólica y la transforma en voz sintética.

Capítulo 5

Gestión del proyecto

5.1. Análisis de requisitos

En este apartado haremos un estudio de los requisitos del proyecto, analizando los requisitos funcionales y no funcionales de las dos partes que lo componen.

Un requisito funcional define una función del sistema de software o sus componentes. Una función es descrita como un conjunto de entradas, comportamientos y salidas. Los requerimientos funcionales pueden ser: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que se supone un sistema debe cumplir.

Un requisito no funcional o atributo de calidad es un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requisitos funcionales

Para hallar los requisitos funcionales de aplicación se realizó un estudio sobre la población objetivo de la misma.

5.1.1. Encuestas con usuarios finales

Se han realizado dos tipos de encuestas, cada una orientada al sector en el que se iba a centrar el proyecto: personas con discapacidad auditiva y personas con discapacidad visual. Ambas encuestas comenzaban con una pequeña descripción de la aplicación además de una explicación sobre la finalidad de esta misma.

La encuesta realizaba preguntas acerca de la información que iban a necesitar durante la guía, el formato en el que esta información iba a ser proporcionada, o la forma que preferían de usar la aplicación, ya sea por voz, teclas o teclado. Adicionalmente se solicitaban sugerencias y posible información útil para llevar a cabo la aplicación.

Las encuestas pretendían ser lo más simples posibles y lo suficientemen-

te abiertas para dar libertad a la hora de elegir opciones, y poder recoger información muy valiosa de cara a la realización de la aplicación.

5.1.1.1. Encuesta discapacidad auditiva

Respecto a la forma en que la guía iba a ser mostrada, hubo una gran variedad de opiniones. No hubo un resultado claro ya que hubo un 37,5 % de la gente que prefería el camino en formato de texto, un 25 % que prefería visual y otro 37,5 % que prefería una mezcla de ambas. Por lo tanto se optó por la última opción para que fuese mas versátil y unificase todas las opiniones recibidas.

En la segunda parte de la encuesta se preguntó si sería útil que se fuese mostrando información de las zonas por la que se pasaba. Ante esta pregunta los resultados fueron muy claros y un 100 % opinó que sí.

La última parte de la encuesta se centraba en la entrada de información por parte del usuario, y de los entrevistados un 62,5 % prefería por teclado mientras que el resto prefería por voz. En este último caso se optó por hacer caso de la mayoría.

5.1.1.2. Encuesta discapacidad visual

Respecto a la forma en que los usuarios con esta discapacidad iban a recibir las instrucciones se les planteó si las preferían en lenguaje natural, por pitidos o una mezcla de ambas. Un 85,7% prefirió el lenguaje natural sobre el de pitidos, mientras que el 14,3 % prefirió una combinación de pitidos y lenguaje natural

Para la selección de opciones un 57,1 % de los encuestados prefirió usar las teclas de volumen para desplazarse por los menús y un 42,9 % prefirió que una voz les leyese las opciones disponibles y ellos seleccionasen la opción también por voz.

En este último caso, debido a que no había un consenso muy unificado, se optó por una solución intermedia, una voz leería las opciones disponibles, pero se usarían las teclas de volumen para desplazarse por los menús.

5.1.2. Requisitos

Requisitos funcionales

- Capacidad de la aplicación de acceder a capas de bajo nivel del dispositivo, para recolectar de éstas la información necesaria para el posicionamiento y los cálculos de rutas.
- Tiene que tener capacidad suficiente para procesar la entrada de información por parte del usuario, ya sea de forma escrita o hablada.

- Debe ser capaz de realizar correctamente el reconocimiento de voz.
- Debe ser capaz de reproducir la ruta dada por el servidor.
- Debe ser capaz de posicionar al usuario en el mapa con la máxima precisión posible.
- Debe proporcionar conexión en tiempo real con el terminal móvil.
- Tiene que ser capaz de encontrar la trayectoria mínima que debe seguir el usuario, teniendo en cuenta el origen y el destino facilitados por la aplicación, así como, si lo hubiese, el grado de discapacidad del usuario.
- La funcionalidad principal de este proyecto es la generación de las rutas y las trayectorias que el usuario debe seguir para alcanzar su destino, por lo que es un requisito vital para el éxito del proyecto.
- Debe ser capaz de generar una serie de instrucciones en el formato adecuado a la situación del usuario a partir de de la ruta calculada: se convierte en un requisito indispensable para el éxito del proyecto, pues es necesario que la ruta se adecue correctamente a cada usuario.
- Debe hacer sencillo, una posible ampliación del proyecto. Es muy probable que en un futuro se quiera ampliar este proyecto a otras discapacidades, por lo tanto es necesario hacer la aplicación lo mas modulable posible.

Requisitos no funcionales: Software

- Versión Android entre la 2.3 hasta la 4.2

Requisitos no funcionales: Hardware

- Se trabajará con un Samsung Galaxy S4 y un Samsung Galaxy Note 3. Se requiere un terminal que disponga de las siguiente características como mínimo: 1Ghz de procesador, 512MB RAM y 5Mb de espacio libre

5.2. Gestión de riesgos

Durante el análisis de riesgos informáticos tomaremos como referencias los activos informáticos, las vulnerabilidades y amenazas a los cuales están expuestos y la probabilidad de que estos ocurran y el impacto que tendrían en caso de que ocurriese. De tal forma podremos adaptar los controles adecuadamente a cada caso.

Los riesgos serán analizados de forma pro activa, de tal forma que serán identificados antes de que alguno ocurra, de esta forma se podrán prevenir y dar soluciones si se llegan a producir.

Una vez no se haya podido evitar un riesgo, lo identificaremos, así como sus causas y se asignarán recursos para evitar que suponga un problema para el correcto desarrollo del proyecto.

Se creará una lista para cada uno de los riesgos, recogiendo el nivel que presenta cada uno, teniendo en cuenta la relación entre la probabilidad y la severidad de que ocurran. Este nivel queda reflejado en la técnica SQAS-SEI (Department of Energy Quality Managers, Software Quality Assurance Subcommittee, 1999) (Tabla 5.1) dónde se establecen los siguientes valores:

Probabilidad/ severidad	Frecuente	Probable	Ocasional	Remoto	Improbable
Catastrófico	Intolerable	Intolerable	Intolerable	Alto	Medio
Crítico	Intolerable	Intolerable	Alto	Medio	Bajo
Serio	Alto	Alto	Medio	Bajo	Tolerable
Menor	Medio	Medio	Bajo	Tolerable	Tolerable
Insignificante	Medio	Bajo	Tolerable	Tolerable	Tolerable

Tabla 5.1: SQAS-SEI

5.2.1. Análisis de los riesgos

Resumimos los riesgos encontrados en la Tabla 5.2, de acuerdo a estos parámetros:

- Nombre
- Probabilidad
- Severidad
- Nivel de Criticidad

Quedan organizados en función de su nivel de criticidad. Los riesgos más graves serán aquellos de nivel de criticidad más alto:

Nombre	Probabilidad	Severidad	N. Criticidad
Abandono de un integrante del equipo	Remoto	Crítico	Medio
Falta de tiempo de dedicación debido al trabajo	Frecuente	Menor	Medio
Desconocimiento de las tecnologías	Ocasional	Insignificante	Tolerable
Estimaciones poco realistas	Probable	Serio	Alto
Problemas con el servidor	Remoto	Serio	Bajo
Problemas con la red wifi de la facultad	Probable	Serio	Alto
Complicaciones con proyectos anteriores	Remoto	Menor	Tolerable
Fallo en la adaptación de la guía a la discapacidad	Probable	Serio	Alto
Posibilidad de bloqueo de API's utilizadas	Improbable	Crítico	Bajo

Tabla 5.2: Criticidad de los riesgos encontrados

5.2.2. Estudio de los riesgos

- Abandono de un integrante del equipo

Descripción: Uno o varios miembros del equipo abandonan el proyecto.

Probabilidad: Remoto.

Severidad: Crítica.

Nivel de Criticidad: Medio.

Posibles causas: Problemas en otros ámbitos.

Indicadores: Falta de asistencia las reuniones No realización del trabajo asignado.

Plan de mitigación: Redistribución de las tareas asignadas a cada miembro. Eliminación de alguna de las funcionalidades a cubrir en el sistema.

- Falta de tiempo de dedicación por culpa del trabajo

Descripción: Debido a que ambos miembros trabajamos podría ser que en algún momento no tuviésemos el tiempo necesario para dedicárselo al proyecto.

Probabilidad: Frecuente.

Severidad: Menor.

Nivel de Criticidad: Medio.

Posibles causas: Entrega de proyectos. Fechas de cursos.

Indicadores: No realización del trabajo asignado.

Plan de mitigación: Redistribución de las tareas asignadas a cada miembro. Intentar recuperar el tiempo en el siguiente intervalo.

- Desconocimiento de las tecnologías

Descripción: Las tecnologías utilizadas en las versiones anteriores podrían ser desconocidas para los nuevos desarrolladores.

Probabilidad: Ocasional.

Severidad: Insignificante.

Nivel de Criticidad: Tolerable.

Posibles causas: Uso de librerías o funciones desconocidas para los miembros.

Indicadores: Ralentización del trabajo propuesto.

Plan de mitigación: Como es algo que será muy normal que pase, se tomará esto en cuenta cuando se haga la estimación en los tiempos de entrega.

- Estimaciones poco realistas

Descripción: Ponemos fechas a los objetivos planteados que no pueden cumplirse.

Probabilidad: Probable.

Severidad: Serio.

Nivel de Criticidad: Alto.

Posibles causas: Problemas en las entregas de los deadlines.

Indicadores: No se realizan las entregas a tiempo.

Plan de mitigación: Redistribución de las tareas asignadas a cada miembro. Bajar las expectativas hasta volver al tiempo base.

- Problemas con el servidor

Descripción: El servidor se encuentra inaccesible.

Probabilidad: Remoto.

Severidad: Serio.

Nivel de Criticidad: Bajo.

Posibles causas: El servidor está apagado.

Indicadores: No se puede conectar al servidor.

Plan de mitigación: Reiniciar el servidor y ver por qué se ha apagado.

- Problemas con la red wifi de la facultad

Descripción: Los datos recogidos de la wi-fi pueden ser incorrectos o haber cambiado desde que se recogieron.

Probabilidad: Probable.

Severidad: Serio.

Nivel de Criticidad: Alto.

Posibles causas: Reajuste de las redes wi-fi. Cambio en los routers de la facultad.

Indicadores: Posicionamiento erróneo.

Plan de mitigación: Actualizar la información de la wifi de la facultad.

- Complicaciones con proyectos anteriores

Descripción: Algún modulo puede llegar a presentar incompatibilidades con las nuevas versiones.

Probabilidad: Remoto.

Severidad: Menor.

Nivel de Criticidad: Tolerable.

Posibles causas: Problemas en el trabajo heredado .

Indicadores: Fallo en el posicionamiento. Fallo en la ruta generada cuando no se tienen discapacidades.

Plan de mitigación: Investigar y corregir el fallo en la versión anterior.

- Fallo en la adaptación de la guía a la discapacidad

Descripción: No se puede adaptar la ruta alguna discapacidad concreta.

Probabilidad: Probable.

Severidad: Serio.

Nivel de Criticidad: Alto.

Posibles causas: Discapacidad no contemplada o fallo al generar la ruta por que no se puede llegar al destino con una determinada discapacidad.

Indicadores: La ruta no es válida.

Plan de mitigación: Re adaptación del código al generar las rutas a las nuevas discapacidades. Informar correctamente de que el destino existe pero no es alcanzable por la discapacidad del usuario.

- Posibilidad de bloqueo de API's/librerías utilizadas

Descripción: puede suceder que algún elemento de las API's/librerías utilizadas sean bloqueadas por parte del desarrollador de las mismas.

Probabilidad: Improbable.

Severidad: Crítica.

Nivel de Criticidad: Bajo.

Posibles causas: Cambio de políticas en las API's/librerías utilizadas.

Indicadores: Errores de funcionamiento de la aplicación.

Plan de mitigación: Cambió a alguna de las otras tecnologías evaluadas que sean viables.

5.3. Planificación del proyecto

El proyecto estaba definido en base a los objetivos que indicaba el plan del proyecto. Los objetivos que había que cumplir era la actualización de la aplicación de guía en interiores desarrollada en años anteriores para personas con discapacidad. Como no se indicó para que discapacidades concretas había que desarrollar el proyecto decidimos centrarnos en dos de ellas. Las elegidas fueron la discapacidad auditiva y la discapacidad visual.

Para llevar el control del proyecto se establecieron reuniones con los directores del proyecto. Estas reuniones se producían cada dos semanas, y en ellas hablábamos de lo realizado hasta la fecha, de los problemas surgidos y del siguiente grupo de objetivos que teníamos que alcanzar.

Como ya se ha mencionado anteriormente, este proyecto es una evolución de otro existente. Por lo tanto la primera tarea lógica que apareció fue el análisis y estudio del estado de la aplicación, para detectar donde debíamos centrarnos. El código estaba bien comentado, aunque algunas veces esos comentarios no eran totalmente correctos, pero nos sirvieron para hacernos una idea de cómo funcionaba la aplicación.

Uno de los mayores problemas de este proyecto es la necesidad de encontrarse en el edificio de la facultad para realizar las pruebas, sobre todo las de posicionamiento. Por lo tanto había veces que el trabajo realizado en casa no se podía probar correctamente hasta que íbamos a la facultad.

La primera tarea de desarrollo que nos planteamos fue migrar a la aplicación Android a un proyecto nuevo, y de esta forma eliminar el código obsoleto que había en ella. Después de esto hubo dos líneas de trabajo en paralelo, la primera consistía en adaptar la parte de generar rutas, la cual se encuentra en el lado del servidor, para que tuviese en cuenta los obstáculos y las discapacidades existentes, y la otra línea fue la adaptación de la aplicación para personas con discapacidad. Para ello hicimos una serie de cuestionarios a diferentes personas, con diferentes discapacidades, de tal forma que

desarrollamos la aplicación en base a estos datos. Durante estos procesos hubo que actualizar los modelos usados para que se ajustasen a las nuevas funcionalidades y la implementación de las nuevas librerías.

Estas dos tareas junto con la tarea de hacer que se comunicasen correctamente ambas partes con los nuevos modelos fue la parte que más tiempo nos acaparó.

El último paso que hubo que hacer fue ajustar la generación de las instrucciones correctamente a cada discapacidad, mejorando el sistema cada vez que lo usábamos hasta que obtuvimos los resultados que deseábamos.

Capítulo 6

Arquitectura de GUIDIN

6.1. Características del proyecto

En esta sección trataremos las características del proyecto en su conjunto. El proyecto está dividido en dos módulos principales, conectados entre sí. Estos son un programa cliente y un programa servidor (Matsudaira., 2012). La división entre cliente y servidor es debida al gran coste computacional (Arora, 2009) que supone el calculo de ruta, y dado que el terminal móvil ya realiza muchas funciones se optó el realizar estas tareas en un servidor, descargando de esta forma la carga del terminal. Aprovechando la necesidad de tener un servidor para el calculo de ruta se decidió usar también el servidor para almacenar la Base de Datos de la aplicación. De tal forma que en el servidor se generan las instrucciones para la ruta y el terminal móvil es un cliente Android que proporciona al usuario el resto de servicios.

En la aplicación Android se establecerá el destino al que se quiere llegar y ésta enviará al servidor dicha información. Dado que la aplicación está dirigida a personas con discapacidades la selección de destino se puede hacer tanto por voz como por teclado, haciendo que la aplicación sea más accesible. A su vez la aplicación reproducirá las instrucciones recibidas del servidor y reproducirá éstas según las necesidades del usuario, el cual habrá indicado como prefiere recibir las instrucciones.

Para mostrar la ruta hasta el destino nos basamos en los datos que hemos recopilado al preguntarle a personas con diferentes discapacidades, decidiendo que para personas con deficiencia auditiva mostraremos la ruta de una forma y para personas con deficiencia visual de otra.

El algoritmo usado para el posicionamiento bajo redes wi-fi es el algoritmo de los k-vecinos más cercanos, o k-nn por sus siglas en inglés, el cual se ha reutilizado partiendo de los proyectos anteriores.

El servidor es el encargado de recoger la información que le envía la aplicación Android y a través de diferentes archivos calcular la ruta óptima al destino. De tal modo que con esta serie de archivos será el servidor el encar-

gado de encontrar la ruta óptima para cada usuario, teniendo en cuenta el origen, el destino y el grado de discapacidad. La ruta generada será preparada en una serie de instrucciones que serán enviadas al móvil para que éste las interprete y se las muestre al usuario de forma legible.

Tanto el cliente como el servidor deberán estar en contacto de forma periódica, tanto para ver que el usuario no se ha desviado de la ruta o por el contrario para ver si lo ha hecho y recalcular la ruta que el usuario debería seguir.

Cliente y servidor se comunican mediante un sistema de red, por el cual intercambian la información necesaria. Los datos del posicionamiento wi-fi son cargados en el cliente, y el módulo WPS se encarga de estimar nuestra posición, generando las coordenadas correspondientes.

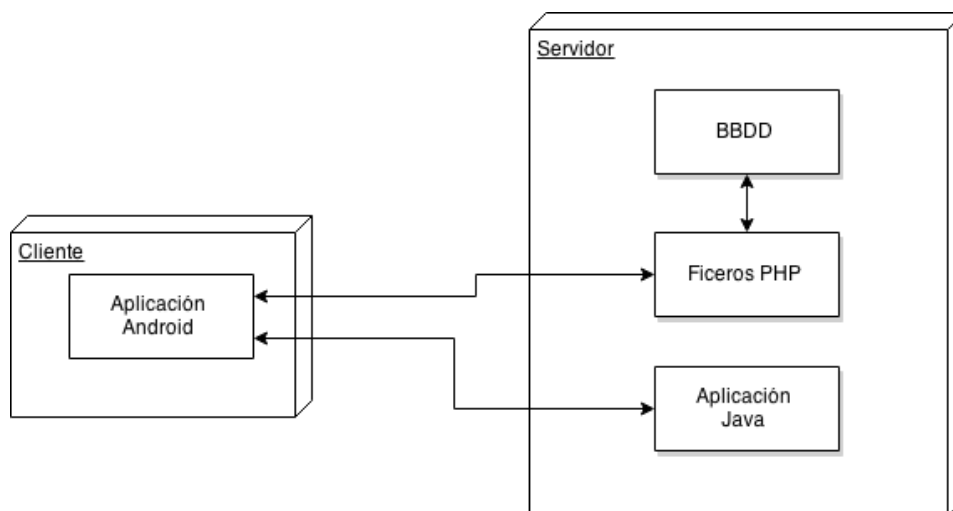


Figura 6.1: Estructura

Como se refleja en la Figura 6.1 y como ya se ha comentado anteriormente esta sería la estructura de la aplicación.

6.2. Comunicación

El cliente y el servidor tienen dos formas de comunicarse: la primera está preparada para la comunicación con la Base de Datos, mientras que la segunda está preparada para la comunicación con la aplicación Java.

Como ya se ha mencionado anteriormente, y como se puede observar en la Figura 6.2, la comunicación con la Base de Datos se realiza a través de un fichero PHP, al cual se le envían los datos por método GET. Éste los recoge y los compara con los que hay en Base de Datos. Tras comprobarlo devuelve al cliente si los resultados son positivos o negativos.

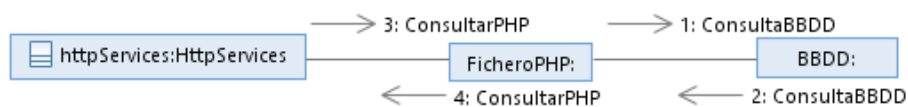


Figura 6.2: Comunicación Cliente con BBDD

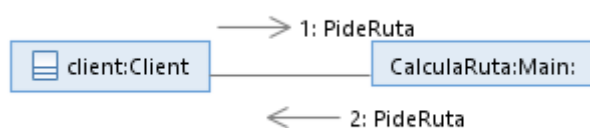


Figura 6.3: Comunicación Cliente con la aplicación Java

Para la comunicación con la aplicación Java se usa un socket que conecta ambas aplicaciones (Figura 6.3), al establecer la conexión con el socket se conecta con la aplicación Java y le envía los datos para generar la ruta.

La aplicación generará la ruta según los datos recibidos y tras generarla usará el socket para devolver la ruta en formato de lenguaje natural.

Por último la aplicación cerrará la conexión con el socket dando así por finalizada la petición por el cliente y se volvería a configurar en modo de espera del siguiente socket.

Capítulo 7

Diseño del cliente

7.1. Introducción

La parte del cliente de este proyecto consiste en una aplicación para móviles con sistema operativo Android, siempre que la versión sea igual o superior a la 4.0, incluyendo así la versión más reciente 5.0 que recibe el nombre de "Lollipop". Esto coincide con la API 21 de Android.

La funcionalidad principal de esta aplicación es la de guiar a personas con discapacidad en el interior de edificios. Se han tenido en cuenta dos discapacidades principalmente, la discapacidad visual y la auditiva.

Para la discapacidad auditiva no hace falta que la aplicación requiera de funcionalidad específica, ya que esta discapacidad no limita a la hora de moverse por los menús o de escribir en el teclado, sin embargo hay que tenerla en cuenta a la hora de mostrar las indicaciones ya que tienen que ser dadas mediante texto.

La discapacidad visual requiere de mucha más personalización para poder usar la aplicación completamente. En todo momento se informa al usuario mediante voz de en qué pantalla se encuentra, de las opciones de las que dispone y de lo que tiene que decir o que teclas físicas pulsar para elegir la opción deseada. Así será capaz de elegir opciones mediante las teclas de volumen, o de indicar el destino mediante voz. Las indicaciones que recibirá serán únicamente mediante voz reproducida por el dispositivo.

Las rutas que reciben los usuarios están adaptadas a las discapacidades que poseen. Al realizar el registro en la aplicación hay que seleccionar en una lista una serie de elementos, que hemos llamado superables, que son ubicaciones tales como, escaleras, rampas o puertas que indican por donde puede pasar el usuario para posteriormente realizar la guía en función a esta información.

La otra gran funcionalidad que ofrece la aplicación cliente es la de medición de intensidades wi-fi. Mediante una interfaz muy sencilla se puede llevar a cabo la medición en cualquier edificio que disponga de los requisitos

necesarios. A continuación en este capítulo se detalla como llevar a cabo la medición (sección 7.5.5.3). Esta función se encuentra disponible solo si el usuario está registrado en la base de datos como administrador. Para que un usuario sea considerado como administrador se debe realizar un cambio en el servidor, por lo tanto desde el cliente no es posible darse de alta como administrador.

Como ya se ha mencionado anteriormente esta aplicación extiende del proyecto Avanti, realizado el año pasado, y del que se diferencia mayormente en la adaptación para discapacitados, la cual requiere de modificaciones tanto en el cliente para hacer la aplicación accesible, como en el servidor para realizar un calculo de la ruta adecuado, así como la modificación de la representación del edificio mediante XML que ha sido modificada para ofrecer información útil al usuario durante la guía.

Para llevar a cabo las modificaciones relativas a la accesibilidad en el cliente se han creado unos cuestionarios, en el que se solicita información sobre como debería ser la aplicación y como se deberían dar las indicaciones de la forma mas adecuada para cada discapacidad. Para ello se hizo llegar dos cuestionarios, uno relativo a la discapacidad visual (Apéndice C.1) y otro a la discapacidad auditiva (Apéndice C.2) a diversas personas que sufren estas discapacidades.

Respecto a la aplicación cliente Avanti, también se ha añadido la ya mencionada diferenciación entre usuarios y administradores de la aplicación. Básicamente la diferencia es que el administrador posee mas opciones en el menú relativas a la medición de intensidades wi-fi, haciendo así un menú mucho mas simple para los usuarios, sin perder las funcionalidades de medición.

7.2. Descripción de las clases utilizadas

A continuación se describen los paquetes y clases utilizados para la implementación del cliente.

- Paquete *http*: utilizado para conectar la aplicación con la base de datos mediante llamadas a archivos php alojados en el servidor. Gracias a estas funciones se realiza el alta y logueo de usuarios o la inserción o recuperación de las coordenadas relativas a un punto en la base de datos. La figura 7.1 muestra el diagrama de clases de este paquete.
- Paquete *sockets*: realiza la conexión con la aplicación java que se encuentra ejecutándose en el servidor. La clase Client se encarga de mandar la información relativa a las coordenadas actuales del usuario y su discapacidad y de recoger la respuesta con la información de la ruta a seguir. El diagrama de clases de este paquete se muestra en la Figura 7.2.

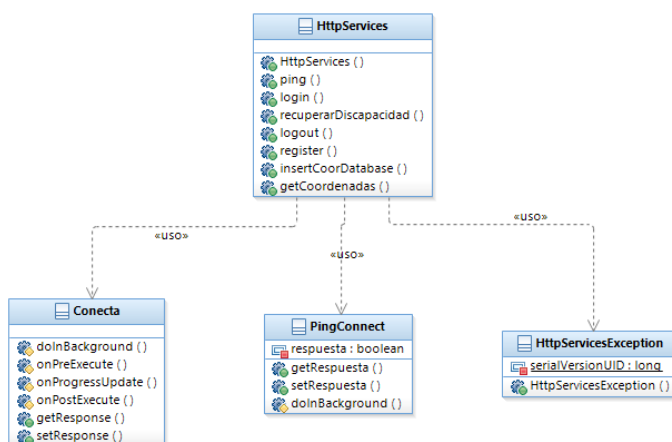


Figura 7.1: Paquete HTTP.

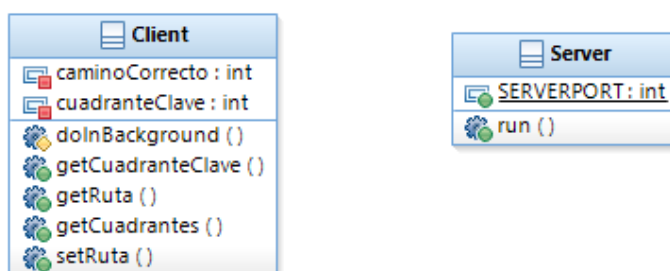


Figura 7.2: Paquete Sockets.

- Paquete *wps*: este paquete se encarga de realizar el escaneo de señales wi-fi (Clase WPS) y de realizar los cálculos necesarios para obtener una posición concreta (Clase WPSDatabase). Además contiene dos Thread, uno relativo a la ubicación y otro a los datos de las intensidades. Las relaciones de las clases de este paquete se muestran en el diagrama de clases de la Figura 7.3.
- Paquete *rutas*: contiene las clases necesarias para la representación del edificio, tales como aristas, cuadrantes y lista de cuadrantes. Su diagrama de clases aparece reflejado en la Figura 7.4.
- Paquete posición: contiene la representación y funcionalidad del mapa que se muestra en el dispositivo.

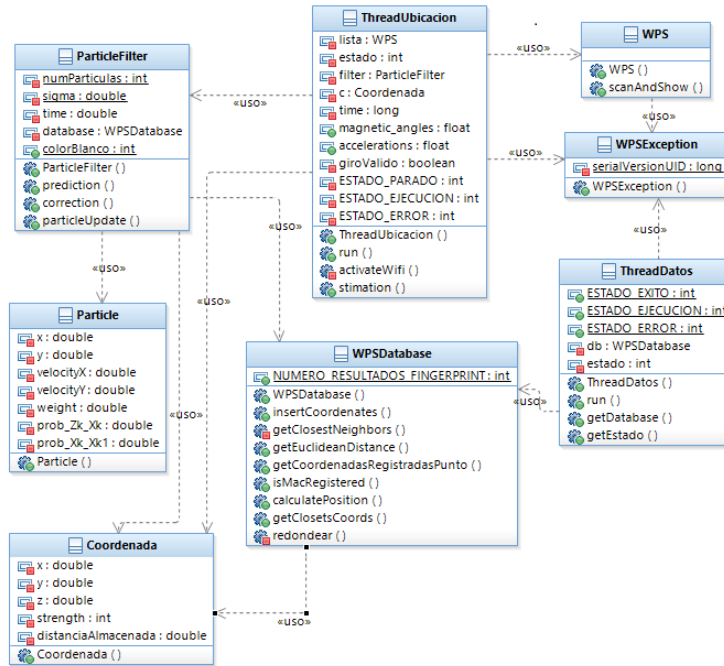


Figura 7.3: Paquete WPS.

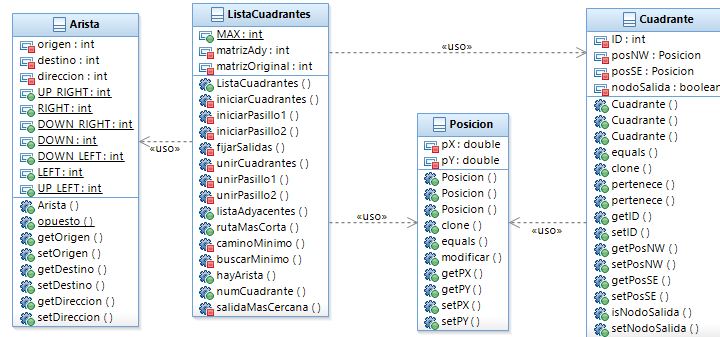


Figura 7.4: Paquete rutas.

- Clase *Acelerómetro*: muestra los sensores del dispositivo relativos al acelerómetro. Para cada eje x, y, y z muestra la acción sobre cada uno de ellos además del giro sufrido.
- Clase *Login*: pantalla principal de la aplicación, permite al usuario entrar al menú o acceder al registro. Nada más iniciar reproduce una voz que explica que hacer en el caso de tener discapacidad visual. Para ello hace uso de las librerías “TextToSpeech” y “SpeechToText” de

Android, al igual que todas las clases que utilizan la reproducción y el reconocimiento de voz.

- Clase *Registro*: permite rellenar los datos necesarios para dar de alta un usuario, se tiene en cuenta la discapacidad que presenta el usuario y se le pide que seleccione los elementos que puede superar. Todos estos datos son enviados directamente a la base de datos.
- Clase *Menú*: esta clase consiste en un `ListAdapter` que contiene las opciones disponibles. Existen dos tipos de usuarios como ya se ha mencionado antes. En la versión para usuarios normales se muestran tres opciones, indicar destino, mostrar posición en el mapa y cerrar sesión. Para los administradores además de estas opciones se añaden las opciones de consultar la actividad wi-fi, ver los acelerómetros y la más importante medir intensidades. Para los usuarios con discapacidades se les da la posibilidad de elegir una opción utilizando las teclas de volumen.
- Clase *Wifis*: clase encargada de obtener toda la información relativa a las redes wi-fi que es capaz de localizar el dispositivo. Utiliza el paquete WPS detallado anteriormente para realizar sus funciones. Muestra la lista de redes que alcanza a localizar el dispositivo.
- Clase *Medir*: presenta una interfaz muy sencilla y ofrece la posibilidad de ir almacenando las intensidades de señal wi-fi de cada punto de forma muy sencilla. a través de unos botones se indica la coordenada x, y y z de la que se quiere almacenar la información y se dispone de un botón que actualiza los datos relativos a las intensidades y los manda a la base de datos. Adicionalmente muestra la última posición medida como ayuda para la medición. Al igual que la clase wifis también hace uso del paquete WPS.
- Clase *Indicar destino*: es la principal clase de la aplicación y es la encargada de realizar todo el proceso de guía, desde solicitar el destino, ya sea por texto o voz hasta dar las indicaciones para llegar al destino solicitado. Para ello hace uso de la mayoría de paquetes mencionados antes. A continuación se describe el proceso que sigue esta clase para cumplir con su función:

Una vez obtenido el destino comienza la ejecución de un bucle que se ejecuta cada 5 segundos (lanzado por un `Timer`), donde se obtiene el cuadrante en el que está posicionado el dispositivo, mediante el método de los k-vecinos. Después esta posición se comparará con los cuadrantes obtenidos del servidor para comprobar si el usuario está en la ruta correcta. Si no fuese así y el dispositivo hubiese salido de la ruta se vuelve a calcular la ruta utilizando la nueva posición actual y el mismo

destino. Una vez hecha la consulta se volverá a iniciar el bucle con los nuevos cuadrantes. Por el contrario si el usuario efectivamente está en la ruta se muestra por pantalla o por voz en función de la discapacidad la instrucción de la siguiente posición a la que el usuario deberá ir. Al llegar a la posición indicada se proporcionan las siguientes instrucciones necesarias para llegar al siguiente punto.

La Figura 7.5 muestra el diagrama de las clases Acelerómetro, Login, Registro, Menu, Wifis, Medir e Indicar destino.

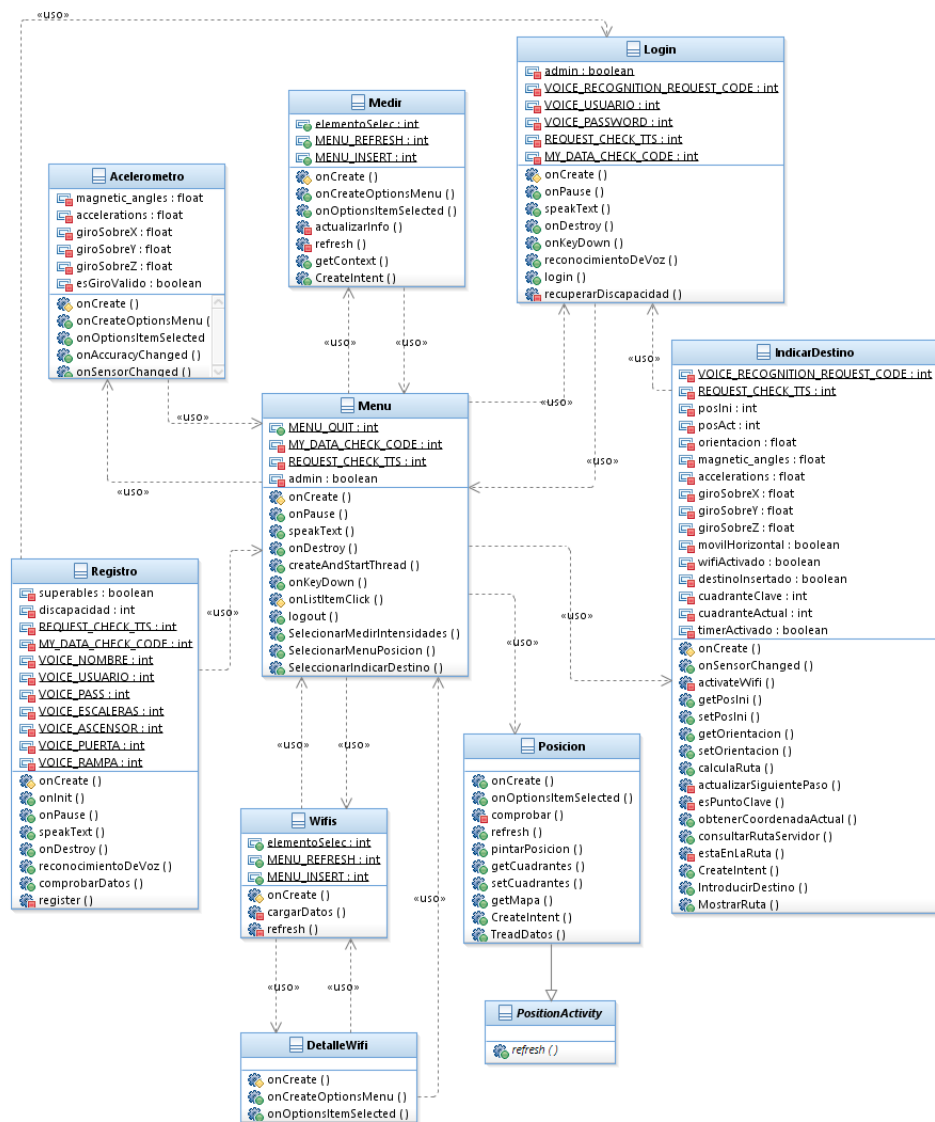


Figura 7.5: Clases principales del cliente.

Las interfaces gráficas de la aplicación están representadas a través de archivos XML lo cuales se encuentran dentro del paquete *res* y dentro de este se encuentran en la carpeta *layout*, ruta genérica de Android para almacenar las interfaces gráficas.

7.3. Diagramas de casos de uso

En esta sección se muestran los diagramas de caso de uso para el caso de un administrador y de un usuario normal. La Figura 7.6 muestra el diagrama para el caso del usuario, y la Figura 7.7 muestran el caso del administrador. El caso de uso del usuario muestra las posibles funciones que puede llevar a cabo, siendo estas mas limitadas que las opciones para el caso de uso del administrador, el cual puede realizar las mismas funciones que el usuario además de otras funciones como medir intensidades, ver los acelerómetros y ver las redes wi-fi. El usuario normal solo cuenta con las funciones de Indicar destino, Mostrar la posición en el mapa, y Cerrar sesión.

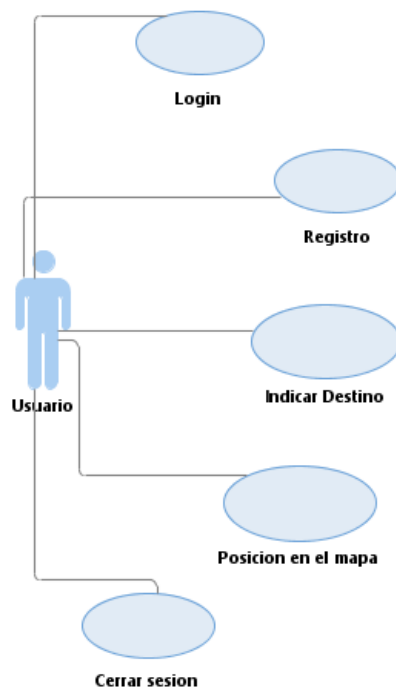


Figura 7.6: Caso de uso usuario.

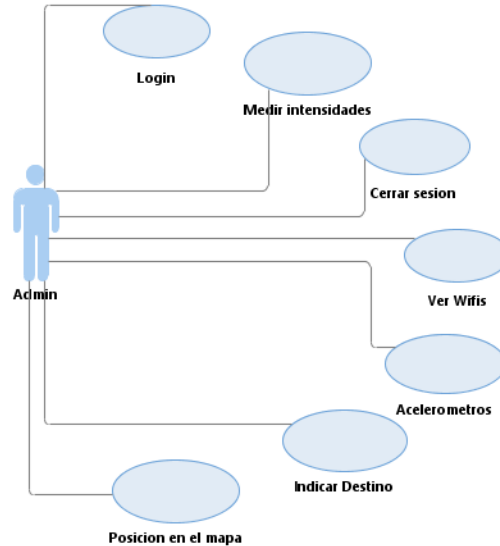


Figura 7.7: Caso de uso administrador.

7.4. Diagramas de secuencia

En esta sección se muestran mediante diagramas de secuencia el flujo de ejecución de las principales funcionalidades de la aplicación.

La Figura 7.8 muestra el diagrama de la función Login. En ella se aprecia que el usuario dispone de la opción de seleccionar la opción de registrarse, interactuando con la clase *httpServices* para realizar la conexión con la base de datos y con la clase *Registro* para rellenar los datos. En el caso de ya estar registrado solo se comprueba si el usuario introducido es correcto mediante una conexión con la base de datos utilizando la clase ya mencionada.

La Figura 7.9 muestra el diagrama de la función Medir intensidades. Para llevar a cabo esta funcionalidad se hace uso de la clase *medir* que proporciona la interfaz necesaria para que el usuario introduzca las coordenadas de la posición que se quiere registrar. La clase *WPS* es la encargada de recoger la información de las redes wi-fi detectadas por el dispositivo. Por último se envía la información a la base de datos haciendo uso una vez mas de la clase *httpServices* para ello.

La Figura 7.10 muestra el diagrama de la función Posición en el mapa. La clase *posición* se encarga de mostrar el mapa de la planta en la que se encuentre el usuario y la información del lugar exacto donde se encuentra, para ello utiliza las clases *ThreadDatos* y *ThreadUbicación*, encargadas de conseguir esta información. La posición mostrada se recalcula constantemente al realizar cualquier movimiento sobre la pantalla.

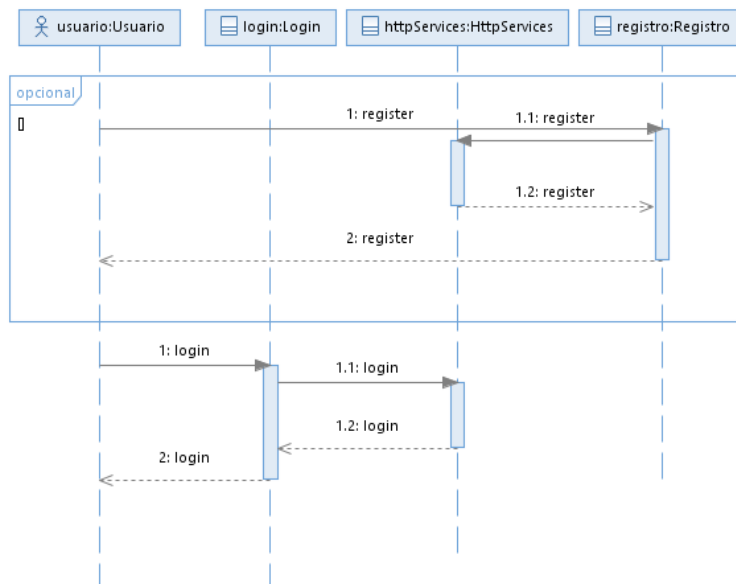


Figura 7.8: Diagrama de secuencia Login.

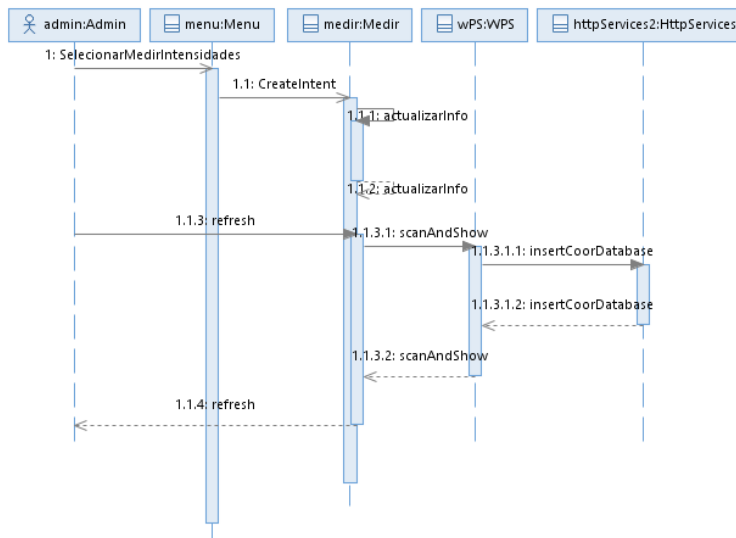


Figura 7.9: Diagrama de secuencia Medir intensidades.

En último lugar, La Figura 7.11 muestra el diagrama de la función Indicar destino. La clase *Indicar destino* es la encargada de solicitar el destino y una vez introducido solicitar todos los datos necesarios para indicar la ruta. Primeramente se necesita la localización inicial del dispositivo y para ello se usa la clase *WPS* de forma similar que para la función posición en el mapa.

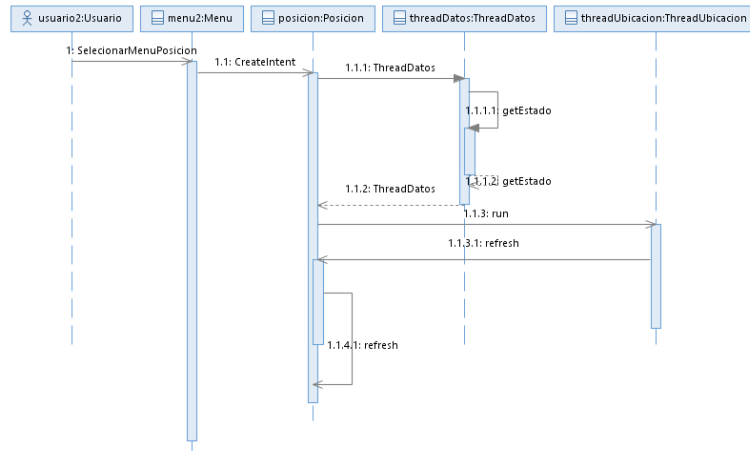


Figura 7.10: Diagrama de secuencia Posición en el mapa.

Una vez calculada la posición se envían los datos al servidor mediante la clase *Client* que se encarga también de almacenar la ruta devuelta por el servidor. Una vez obtenida la ruta la clase indicar destino va mostrando la ruta al usuario según la va necesitando. Esta clase funciona con un *Timer*, que ejecuta el método *calcularRuta* cada segundo para comprobar la posición del usuario y seguir mostrando una ruta correcta.

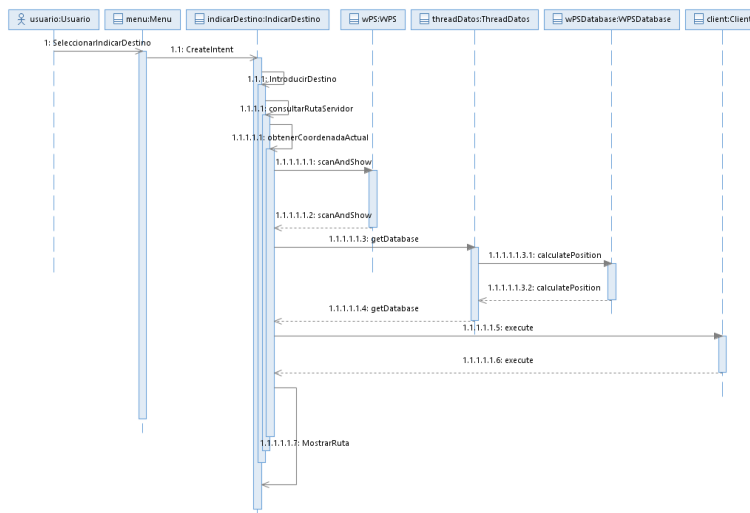


Figura 7.11: Diagrama de secuencia Indicar destino.

7.5. Manual de usuario

En esta sección se explica como usar la aplicación, tanto desde el punto de vista de los usuarios normales, como el de los administradores. Se detallará como usar cada pantalla tanto si el usuario sufre discapacidad visual o auditiva como si no padece ninguna, ya que la aplicación es útil también para personas sin discapacidad.

7.5.1. Pantalla de inicio de la aplicación

Esta pantalla, que podemos apreciar en la figura 7.12 es la de inicio de la aplicación que nos da la posibilidad de entrar al menú rellenando los campos usuario y contraseña si el usuario se ha registrado previamente, o si por el contrario el usuario quiere darse de alta para poder usar la aplicación se pulsará el botón Registrarse.

Respecto a la adaptación para personas con discapacidades, nada más iniciar la aplicación se reproducirá una voz que preguntará por voz al usuario si tiene discapacidad visual y le indicará que ha de decir y cuando para poder interactuar con esta pantalla, ya sea por voz o a través de las teclas físicas del dispositivo.

7.5.2. Pantalla de registro de la aplicación

Esta pantalla (Figura 7.13) permite a un usuario darse de alta en nuestra base de datos. Se solicita la discapacidad y en caso de que la haya, se pide que seleccione en una lista los obstáculos que puede superar, ya sean escaleras, ascensores, puertas o rampas.

Si se ha llegado a esta pantalla través de un comando de voz se asume que el usuario es invidente y se empiezan a solicitar los datos por voz, como si de una conversación se tratase. Una vez rellenos todos los datos, se pulsará el botón Aceptar y se volverá a la pantalla de inicio para realizar el inicio de sesión y poder empezar a usar la aplicación.

7.5.3. Menú principal

Existen dos menús en la aplicación, una para usuarios y otro para administradores. Los usuarios solo dispondrán de opciones relacionadas con el posicionamiento y la sesión. En concreto tendrán la opción de indicar un destino para generar las instrucciones hasta el punto indicado, mostrar la posición en el mapa del edificio, o cerrar sesión.

Los administradores contarán además de con las opciones descritas, con otras opciones necesarias para el desarrollo de la aplicación y para el mapeo

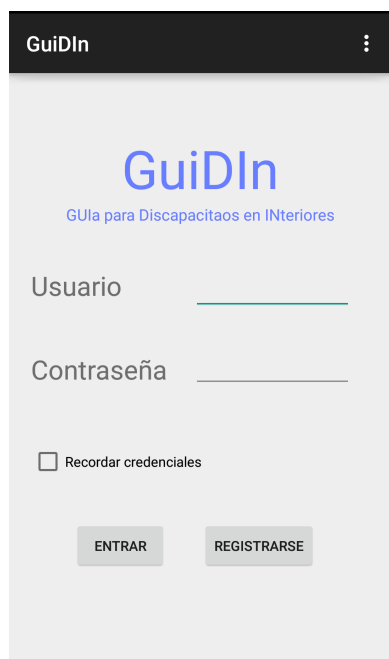


Figura 7.12: Inicio de la aplicación.



Figura 7.13: Pantalla de registro.

del edificio, como son: ver los acelerómetros o medir intensidades de las señales wi-fi. Todas estas opciones se detallan en los siguientes puntos.

7.5.4. Menú para Usuarios

Esta pantalla consiste básicamente en un menú como se puede apreciar en la Figura 7.14 que muestra las opciones principales de GuiDIn: Indicar destino, Posición en el mapa, y Cerrar sesión. Las opciones son leídas por la aplicación si el usuario es discapacitado visual, y tendrá la opción de seleccionarlas mediante las teclas de volumen y la tecla de botón atrás. El botón Subir volumen permite seleccionar Indicar destino, el de Volumen abajo permite seleccionar Posición en el mapa, y por último la tecla de botón atrás permite volver a la página de inicio cerrando la sesión.

7.5.4.1. Indicar destino

Ésta es la primera opción del menú de usuarios la cual permite generar las instrucciones necesarias para llegar al destino indicado. El destino se solicita por voz o teclado en función de la discapacidad del usuario, y las instrucciones se muestran por pantalla o las lee la aplicación también en función de la discapacidad del usuario. Posee una interfaz muy sencilla como se puede ver en la Figura 7.15

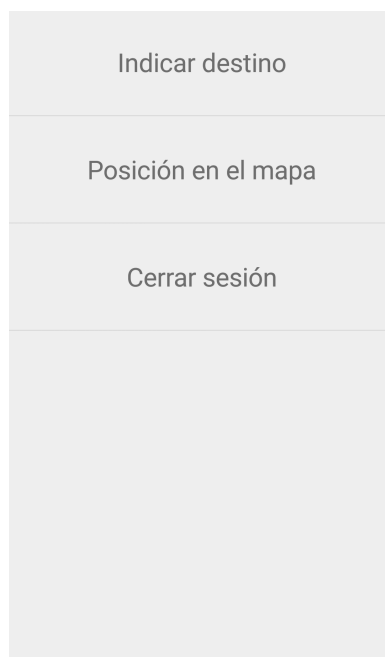


Figura 7.14: Menú de usuario

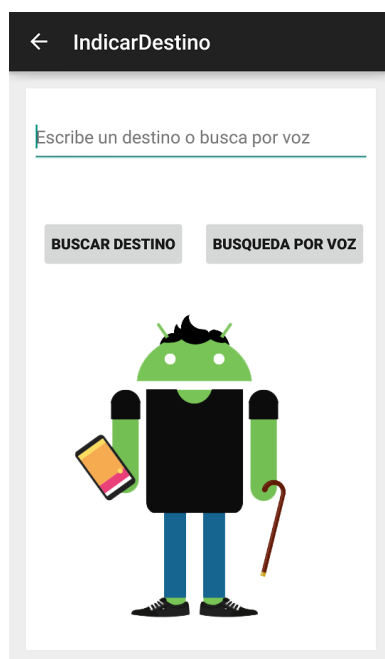


Figura 7.15: Pantalla de indicar destino

7.5.4.2. Posición en el mapa

La función de esta pantalla (Figura 7.16) es mostrar el mapa de la planta donde se encuentra el usuario e indicarle donde se encuentra, para que le sirva para orientarse y como ayuda para la guía. Si el usuario sufre discapacidad visual se le indicará por voz donde se encuentra.

7.5.4.3. Cerrar sesión

Esta opción cierra la sesión del usuario y nos devuelve a la pantalla de inicio de la aplicación.

7.5.5. Menú para Administradores

Como se dijo al inicio de esta sección, los administradores contarán con más opciones en el menú de la aplicación. En la Figura 7.17 se pueden observar todas las opciones. A continuación se detallan cada una de ellas.

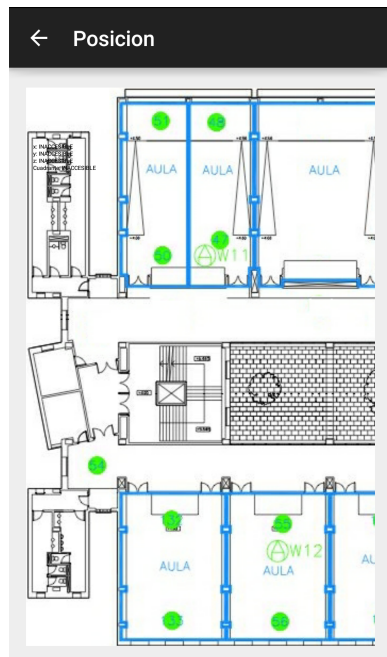


Figura 7.16: Pantalla de posición en el mapa.

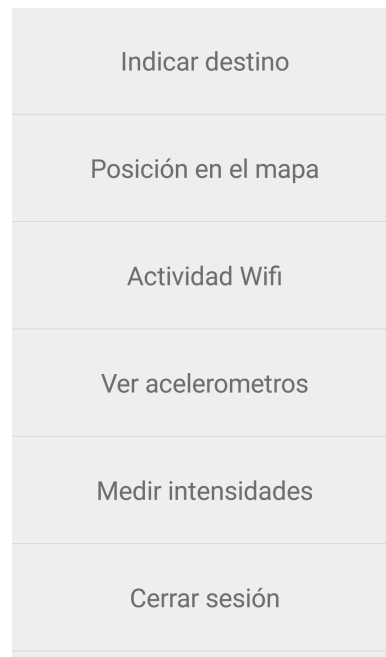


Figura 7.17: Menú de administradores.

7.5.5.1. Actividad wi-fi

Esta opción nos lleva a una primera pantalla (Figura 7.18) en la que se muestra la lista de repetidores wi-fi que está recibiendo el dispositivo en el momento que se selecciona esta opción. Si se pulsa en una de ellas la aplicación nos lleva a otra pantalla en la que veremos la red seleccionada en detalle (Figura 7.19). En concreto se muestra el SSID, el BSSID, la frecuencia y la potencia de la red wi-fi seleccionada.

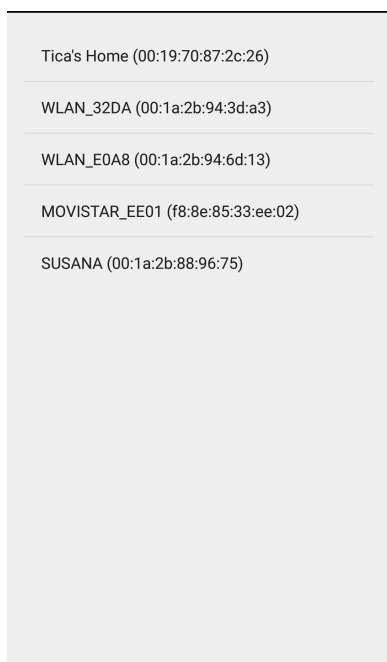


Figura 7.18: Pantalla de actividad wi-fi

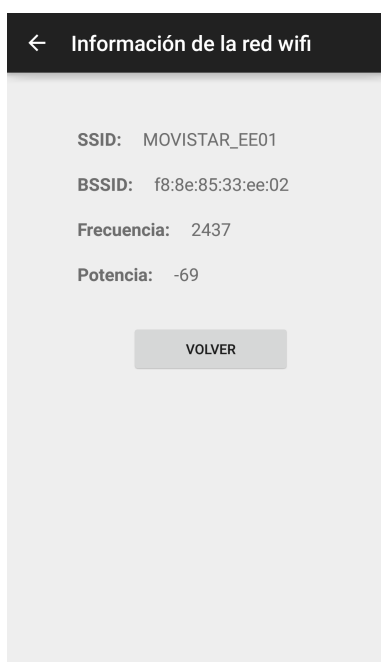


Figura 7.19: Pantalla de detalle de red wi-fi

7.5.5.2. Ver acelerómetros

Esta pantalla (Figura 7.20) muestra la lista de acelerómetros con los que cuenta el dispositivo y muestra el valor de cada uno de ellos.

7.5.5.3. Medir intensidades

Esta función es de gran utilidad para los administradores ya que permite almacenar toda la información recibida por el dispositivo de las redes wi-fi de las que recibe señal. El procedimiento que hay que seguir es indicar el punto en el que se está del edificio a través de los botones “+” y “-” correspondientes con los ejes, x, y, z como se puede apreciar en la Figura 7.21 y posteriormente

pulsar el botón Refrescar y Aceptar. Esta acción almacenará todos los datos necesarios en la base de datos.

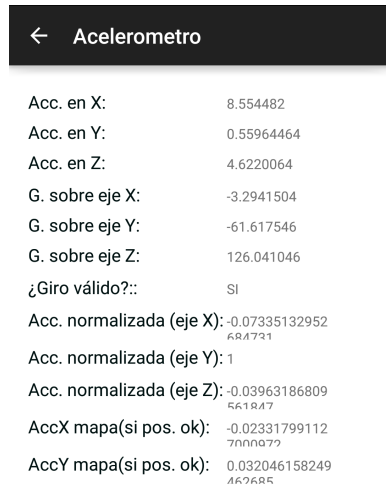


Figura 7.20: Pantalla de acelerómetros.

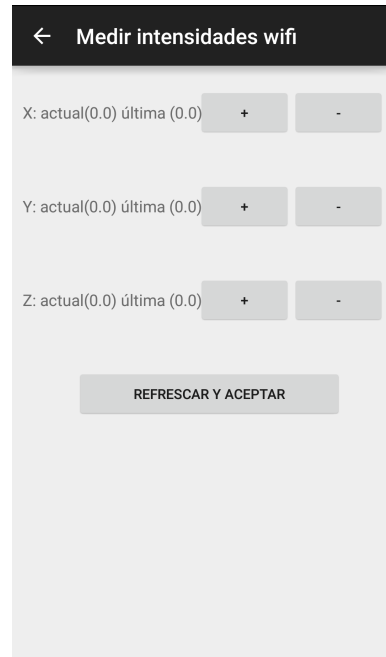


Figura 7.21: Pantalla de medir intensidades.

Capítulo 8

Diseño del servidor

8.1. Introducción

El servidor en este proyecto se divide en tres partes, la Base de Datos (BBDD), la aplicación en Java para el cálculo de rutas y el servidor PHP para la conexión entre la aplicación y la Base de Datos.

La BBDD se encuentra montada sobre MySQL y se utiliza para el almacenamiento de los usuarios y de las intensidades wi-fis de los edificios.

Por su lado la parte de PHP es la encargada de comunicar el cliente con la BBDD, siendo estos ficheros los que realizan las consultas a la BBDD con la lógica pedida por parte del cliente (Figura 8.1).

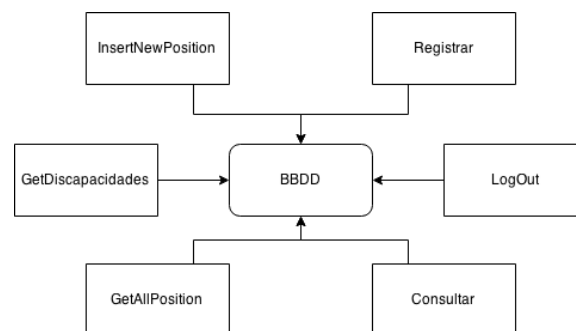


Figura 8.1: Comunicacion PHP con BBDD

La parte que completa el servidor es la aplicación Java que se ejecuta en el servidor. Esta parte es la más compleja de las tres que están alojadas en el servidor ya que es la encargada de generar la ruta desde la posición actual del usuario hasta el destino seleccionado, usando para ello un algoritmo de búsqueda en anchura en un árbol (Cormen et al., 2001). El cliente le envía a la aplicación la posición inicial y la final mediante un socket, además de la discapacidad del usuario y los obstáculos que éste puede llegar a superar. Con la información recibida por medio del socket y la información que tiene

almacenada en formato XML genera la ruta óptima para cada usuario.

Como ya se ha mencionado anteriormente esta aplicación es una evolución de un anterior proyecto, pero en esta versión se la quiso hacer accesible a personas con discapacidades.

Para ello tuvimos que adaptar la BBDD para que almacenase tanto la discapacidad de cada usuario como la lista de objetos que cada usuario es capaz de superar. Además hubo que modificar los ficheros PHP para que añadiesen dicha información en el registro y consulta de los usuarios.

Por su parte la aplicación Java tuvo que ser modificada para que al generar los caminos tuviese en cuenta los obstáculos que podían ser superados y a la hora de generar las indicaciones para la ruta hubo que tener en cuenta la discapacidad, ya que la ruta no se generaría igual para una persona que pueda leer la pantalla a otra que no.

8.2. Descripción de la Base de Datos

Como ya se ha dicho la BBDD esta montada sobre MySQL (Oracle, 2015) y reutiliza lo creado hasta ahora por las anteriores versiones de la aplicación, añadiendo lo necesario para las discapacidades.

La Base de Datos cuenta con tres tablas (Figura 8.2):

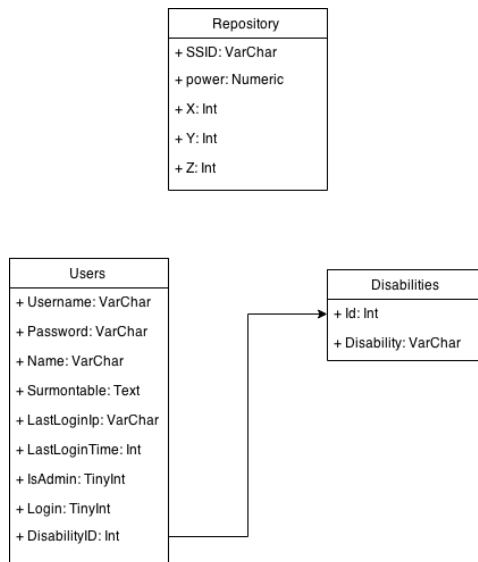


Figura 8.2: Tablas de BBDD

- Repository: Es la tabla que contiene toda la información de las wi-fis de los edificios, indicando para cada coordenada la potencia de cada red wi-fi que se mide en ella.

- **Disabilities:** Esta tabla es nueva para este proyecto y contiene cada discapacidad soportada y un ID que la identifica:
 - 0: Sin discapacidad.
 - 1: Discapacidad Auditiva.
 - 2: Discapacidad Visual.
- **Users:** Esta tabla contiene la información de cada usuario, se adaptó para que cada usuario guardase su discapacidad y su lista de obstáculos superables, para cada usuario se guarda la siguiente información:
 - Username: único para cada usuario.
 - Password: contraseña del usuario.
 - Name: nombre real del usuario.
 - Surmountable: lista de superables por el usuario.
 - LastloginIp: ultima Ip desde la que hizo login.
 - LastloginTime: Ultima hora en la que hizo login.
 - IsAdmin: indica si el usuario es Admin o no.
 - Login: indica si esta logeado o no.
 - Disabilityid: ID de la discapacidad del usuario.

8.3. Ficheros PHP

Los ficheros PHP se encuentran alojados en un servidor Apache, de forma que son accesibles mediante una conexión Rest (Richardson, 2007).

La mayor parte de los ficheros ya existían, a excepción de `getDiscapacidad.php` que es totalmente nuevo, pero al igual que la BBDD la mayoría de estos ficheros han tenido que ser modificados para los nuevos requisitos.

Los ficheros PHP disponibles son:

- `Registrar.php`: añade un nuevo usuario a la Base de Datos. Este fichero ha sido modificado para que guarde de cada usuario su discapacidad y los objetos superables por cada uno.
- `Logout.php`: Desloguea a un usuario de la aplicación. Este archivo no ha sido modificado.
- `InsertNewPosition.php`: Añade una nueva posición con sus intensidades a la Base de Datos. Este archivo no ha sido modificado.
- `GetDiscapacidad.php`: Dado un usuario devuelve su discapacidad y la lista de objetos que es capaz de superar el usuario.

- GetAllPositions.php: Devuelve todas las posiciones almacenadas de la Base de Datos. Este archivo no ha sido modificado.
- Cosultar.php: Consulta a la Base de Datos si existe un usuario con una contraseña concreta y se usa para el login. Si devuelve un acierto es que es correcto y el cliente procede al login.

8.3.0.4. Diagramas de secuencia

Para la comunicación con la Base de Datos se sigue la estructura que se refleja en la Figura 8.3

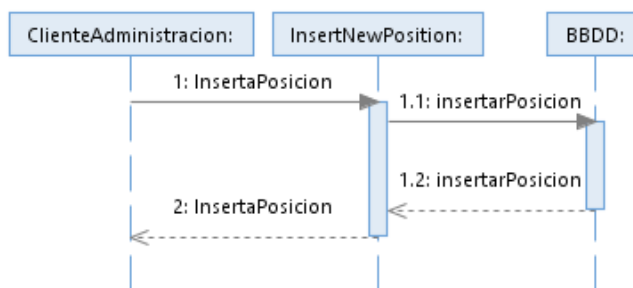


Figura 8.3: Nueva posición

Este método solo puede ejecutarlo un cliente que se haya registrado en la aplicación como administrador.

Otro de los métodos mas utilizados en la comunicacion con la Base de Datos es el método que devuelve todas las posiciones almacenadas (Figura 8.4). Este método, al contrario que el anterior, no necesita que el cliente sea un administrador.

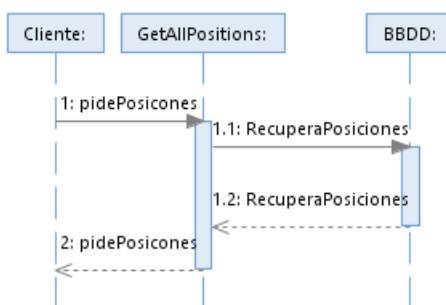


Figura 8.4: Dame posiciones

En estos dos diagramas se ha realizado el proceso de login con antelación. Para llevarlo a cabo también se ponen en contacto el Cliente y el Servidor (Figura 8.5).

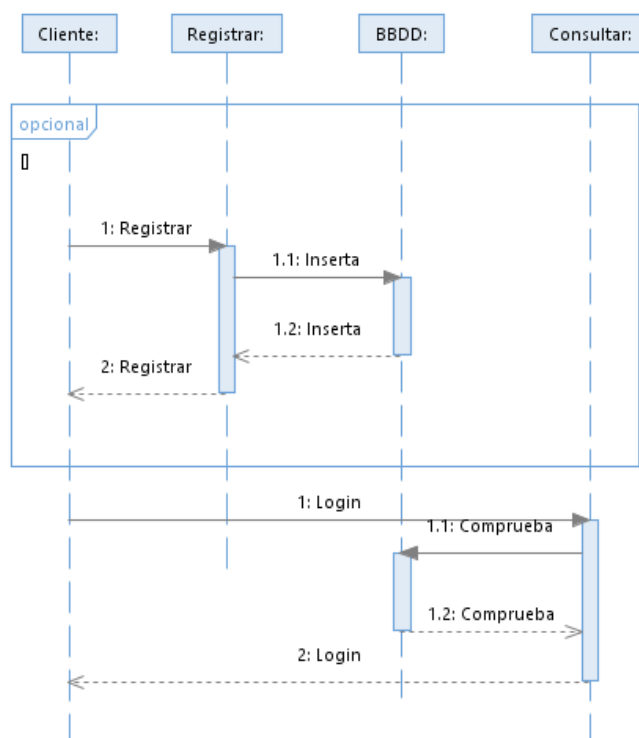


Figura 8.5: Login

8.4. Aplicación Java

La principal función de la aplicación Java es encontrar la ruta óptima entre la posición inicial y la posición destino. Para ello se usa una búsqueda en anchura sobre un grafo que contiene la información del edificio.

La información del edificio está almacenada en una serie de archivos XML. En estos ficheros la facultad está dividida por plantas, cada planta por pasillos y cada pasillo por estancias.

8.4.1. XML

El estado del edificio se encuentra en dos tipos de XMLs, el primero indica todos los edificios y sus respectivas plantas, los segundos indican para cada planta la información correspondiente.

Ejemplo XML de un edificio:

```

<edificio>
  <planta nombre='nombrePlanta'>
    <archivo>nombreFichero</archivo>
  </planta>
</edificio>
  
```

En el XML de los edificios se tiene que indicar todas las plantas del mismo y hay que poner el nombre de la planta el nombre del fichero XML que contendrá la información de la planta, pero sin la extensión XML.

Ejemplo XML de una planta:

```
<?xml version='1.0' encoding='UTF-8'?>
<planta>
  <Z>1</Z>
  <estancia id="pasillo1">
    <tipo>PASILLO</tipo>
    <cuadrantes>
      <cuadrante idc="0">
        <SE>
          <X>3</X>
          <Y>13</Y>
        </SE>
        <NW>
          <X>0</X>
          <Y>16</Y>
        </NW>
        <conectado>
          <norte>-1</norte>
          <sur>-1</sur>
          <este>1</este>
          <oeste>-1</oeste>
        </conectado>
        <objetos>
          <objeto id="objeto1">
            <tipo>mesa</tipo>
            <total>>false</total>
            <posicion>
              <X>1</X>
              <Y>1</Y>
            </posicion>
          </objeto>
          <objeto id="objeto2">
            <tipo>mesa</tipo>
            <total>>false</total>
            <posicion>
              <X>1</X>
              <Y>2</Y>
            </posicion>
          </objeto>
          .
          .
          .
        </objetos>
      </cuadrante>
    </cuadrantes>
  </estancia>
</planta>
```

```

    <cuadrante idc="1">
      <SE>
        <X>6</X>
        <Y>13</Y>
      </SE>
      <NW>
        <X>3</X>
        <Y>16</Y>
      </NW>
      <conectado>
        <norte>-1</norte>
        <sur>-1</sur>
        <este>2</este>
        <oeste>0</oeste>
      </conectado>
      <objetos>
        <objeto id="objeto1">
          <tipo>fuente</tipo>
          <total>>false</total>
          <posicion>
            <X>1</X>
            <Y>3</Y>
          </posicion>
        </objeto>
      </objetos>
    </cuadrante>
    .
    .
    .
  </cuadrantes>
</estancia>
.
.
.
</planta>

```

En este elemento tenemos una planta por fichero, la planta esta dividida por estancias, y a su vez, las estancias están divididas por cuadrantes. Las estancias se identifican por ids al igual que los cuadrantes.

La lista de elementos que contiene este fichero, que corresponde a una planta determinada, es:

- Z: Altura de la planta, única por fichero también
- Estancia: División principal del mapa, cada planta puede tener n estancias.
 - Cuadrante: Cada estancia tendrá un numero n de cuadrantes

- SE: Posición X e Y que indica la esquina Sur-Este del cuadrante
- NW: Posición X e Y que indica la esquina Nor-Oeste del cuadrante
- Conectado: Indica el id del cuadrante con el que esta conectado y en que dirección.
- Objetos: Lista de objetos que contiene el cuadrante.
 - ◇ Objeto: Objeto concreto identificado por id
 - ◇ Tipo: Tipo de objeto que estamos tratando
 - ◇ Total: Si es un obstáculo total o no.
 - ◇ Posición: Posición del objeto en el cuadrante

8.4.2. Java

La aplicación Java tiene la estructura que se muestra en la Figura 8.6

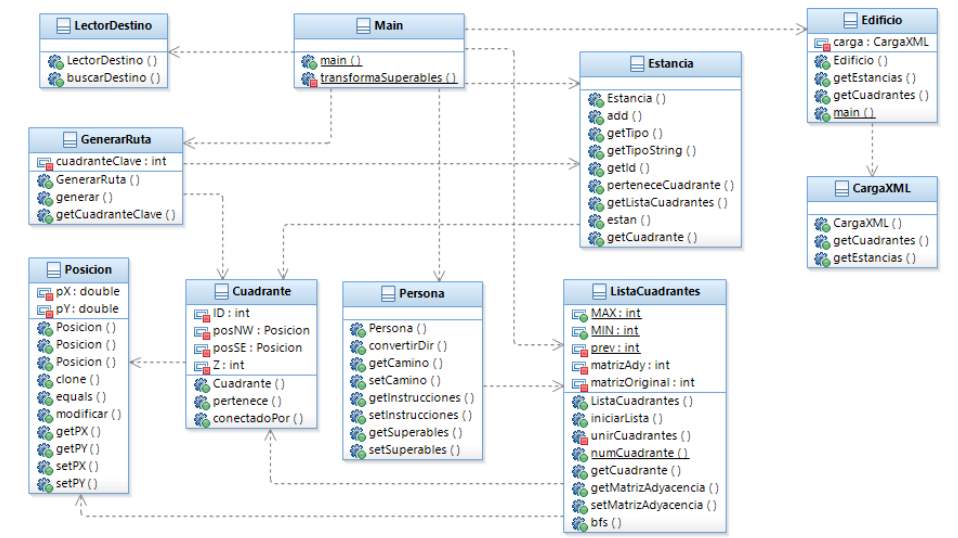


Figura 8.6: Diagrama de clases Java

- Main: Clase encargada de ejecutar la aplicación.
- Cuadrantes: Clase que engloba toda la información de un cuadrante.
- Estancias: Clase que engloba toda la información de una estancia.
- GenrarRuta: Clase con los métodos para generar la ruta óptima.
- LectorDestino: Transforma el destino escrito a un cuadrante concreto.

- ListaCuadrantes: Contiene una lista de cuadrantes.
- Objeto: Clase que engloba la información relativa a un objeto.
- Posicion: Clase que simboliza una posición en el mapa.
- TipoObjeto: Enumerado con el tipo de objetos que tenemos.
- Persona: Clase que engloba la información de una persona.
- CargaXML: Clase encargada de leer los XML y transformarlo a las clases correctas.
- Edificio: Clase que engloba un edificio entero.

8.4.2.1. Diagramas de secuencia

En la Figura 8.7 se representa la secuencia que sigue el proceso encargado de generar las ruta.

La conexión entre el servidor y la aplicación cliente se realiza mediante un socket (Wikipedia, a) en la clase Main, la cual espera hasta que una aplicación se conecta y va esperando hasta que recibe toda la información necesaria.

Una vez que recibe toda la información forma un elemento de tipo Persona, el cual contiene la posición inicial, el destino y la información del edificio. Con toda esta información la clase Persona genera la ruta. Para hallar el camino óptimo usa una búsqueda por anchura generando un árbol de búsqueda con la información del edificio, siendo el padre el inicio y el objetivo el destino.

El método de búsqueda es parte de lo que estaba implementado del año anterior. En esta versión hemos añadido el tipo de obstáculos y comprobamos antes de pasar de un cuadrante a otro si puede pasar por ese obstáculo. Si no es así se toma como que no hay camino entre esos dos cuadrantes y seguirá buscando.

Una vez obtenido la ruta óptima lo transformamos a un camino legible a través de la clase GeneraRuta.

Por último se utiliza el socket creado al principio para enviar por este medio la ruta de forma legible al dispositivo.

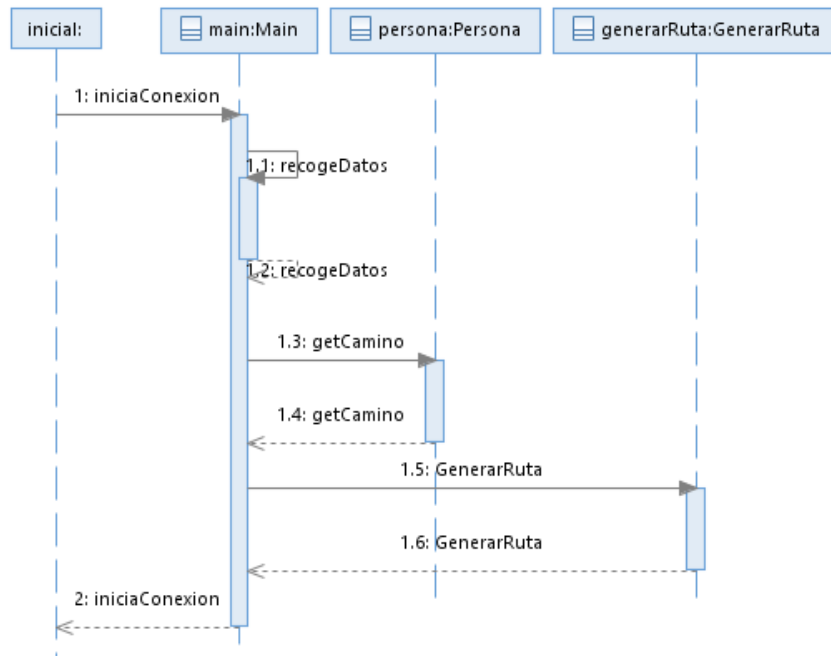


Figura 8.7: Secuencia Calcula Ruta

Capítulo 9

Conclusiones y trabajo futuro

9.1. Descripción de los resultados obtenidos

El primer objetivo que pretendíamos cumplir era conseguir localizar a un individuo dentro del edificio de la Facultad de Informática. Este objetivo a supuesto un reto debido a diversos problemas relacionados con la red wi-fi de la Facultad de informática. Sin embargo, finalmente este objetivo ha sido cumplido y se ha podido posicionar un dispositivo que se encuentre dentro del edificio mencionado.

Otro de los objetivos primarios fue el de conseguir adaptar la aplicación a personas con discapacidad visual o auditiva. Para ello se han hecho modificaciones tanto en el dispositivo móvil como en el servidor, todas ellas destinadas a lograr este objetivo. Todo ello ha sido añadido al proyecto satisfactoriamente y el resultado ha sido una aplicación completamente accesible para personas con alguna de estas discapacidades mencionadas.

Una de nuestras pretensiones era la de mejorar el posicionamiento. Esta mejora no ha sido posible llevarla a cabo. Sin embargo esto no supone un impedimento a la hora de usar la aplicación ya que el margen actual es suficientemente preciso.

Por último se propusieron dos objetivos secundarios, no por ello menos importantes, que servían como mejora de la aplicación. Dichos objetivos son, por un lado el incremento de destinos a los que se puede llegar con la aplicación y por otro la mejora de las instrucciones guía proporcionadas. Este objetivo es importante si tenemos en cuenta el tipo de usuarios a los que va dirigida la aplicación. El lenguaje utilizado para proporcionar las instrucciones es más “humano” y conexo, además cada instrucción es dada en el momento en el que se completa la instrucción anterior.

9.2. Conclusiones

La mayor parte de resultados obtenidos han sido positivos ya que se han alcanzado la mayor parte de los objetivos que se habían propuesto.

La accesibilidad para personas con discapacidad visual o auditiva ha sido realizada satisfactoriamente.

Por otro lado, las instrucciones son ahora proporcionadas en lenguaje mas natural. Siempre queda cierto margen de mejora en este aspecto, pero respecto a la versión anterior se puede apreciar dicha mejora.

En siguiente lugar, y relativo a la propuesta de incremento de destinos, podemos concluir que este objetivo ha sido claramente cumplido ya que para la versión anterior sólo se contemplaban un par de destinos y actualmente se contemplan la mayor parte de destinos posibles dentro de las zonas mapeadas del edificio.

En último lugar, el posicionamiento de un dispositivo ha sido conseguido con la misma precisión que se consiguió en versiones anteriores. Por lo tanto, se ha conseguido posicionar pero no ha sido posible mejorar el margen de precisión.

9.3. Trabajo Futuro

Tal y como está diseñada la aplicación GuiDIn, es muy fácil reutilizar la aplicación para cualquier edificio público. Por lo tanto, sería una opción viable extender la aplicación a todas las facultades de la Universidad Complutense, o incluso a algún museo o centro público, lugar donde sería de gran utilidad una aplicación de estas características. El sistema de mapeo implementado para representar los edificios es sencillo y fácil de replicar.

Otra posible ampliación sería añadir más discapacidades ya que actualmente solo se contempla la discapacidad visual y la auditiva, siendo fácilmente implementable añadir cualquier discapacidad.

Una gran mejora sería añadir una interfaz de guía similar a un GPS para exteriores, algún tipo de flecha o indicador que funcione sobre el mapa o utilizando realidad aumentada sobre el propio edificio.

Esta aplicación podría llegar a ser realmente útil si se llegase a ampliar la funcionalidad de guía también en exteriores, utilizando por supuesto el GPS del dispositivo. De esta forma la guía sería completa y podría guiarse a una persona, con discapacidad o, no desde cualquier punto hasta el interior de un edificio.

Por ultimo, para poder llegar a un número de personas mayor, se podría añadir la opción del multilinguaje, y así llevar la aplicación a lugares de habla no hispana.

Capítulo 10

Conclusions and future work

10.1. Description of results

The first goal we wanted to accomplish was to locate an individual within the building of the Facultad de Informática. This goal has been a challenge because of various problems related to the wi-fi network at the Facultad de Informática. Eventually, however, this objective has been achieved and has been able to position a device that is within the listed building.

Another of the primary objectives was to get to adapt the application to persons with visual or hearing disability. This has required to made changes to both, the mobile and server, all aimed at achieving this goal. This has been successfully added to the project and the result is a fully accessible for people with disabilities mentioned in this application.

One of our aims was to improve the position. This improvement has not been possible to carry it out. However this is not an impediment to using the application because the current margin is sufficiently precise.

Finally, two secondary objectives that served as improved application implementation were proposed. These objectives are, on the one hand the increase in destinations that can be reached with the application and on the other improving the guidance instructions provided. This goal is important if we take into account the type of users that the application is directed. The language used to provide instructions is “ human ” and connected, each instruction is also given at the time that the previous instruction is completed.

10.2. Conclusions

Most results have been positive since we have achieved most of the goals proposed.

Accessibility for people with visual or hearing disability has been performed successfully.

Moreover, the instructions are now given in more natural language. There is always some room for improvement in this respect, but the improvement respect to the previous versions, can appreciated.

In the next place, and on the proposed increase of destinations, we can conclude that this objective has been clearly fulfilled regarding the previous version, in which only have a few destinations are watched and currently it have the most possible destinations contemplated within the zones mapped the building.

Finally, the positioning of a device has been achieved with the same precision as was achieved in previous versions. Therefore, it has managed to position but has not been possible to improve the margin of accuracy.

10.3. future work

As the GuiDIn application is designed, it is very easy to reuse the application to any public building. Therefore, would be a viable option to extend the application to all the faculties of the Universidad Complutense, or even to a museum or public institution, where it would be very useful. The mapping system implemented to represent the buildings is simple and easy to replicate.

Another possible extension would be to add more disabilities, currently only the visually impaired and hearing are contemplated, being easily implementable to add any other disability.

A big improvement would be to add a user interface similar to a GPS guide for outdoor, some kind of arrow or indicator that works on the map or using augmented reality on the building itself.

This application could become really useful if you were to extend the functionality of guide also outdoors, using of course the GPS device, the guide would be complete and would be guided to a person, disabled or not from any point to the interior of a building.

Finally, in order to reach a larger number of people, you could add the option of multilingual, and thus lead to the application of non-Spanish speaking places.

Capítulo 11

Trabajo individual

Al ser un proyecto pensado para dos personas hemos intentado repartir el trabajo de la forma más justa posible, trabajando más un miembro del grupo cuando el otro estaba más ocupado y viceversa. Una parte de este proyecto ha sido hecha de forma conjunta, pero también hay partes que se han hecho individualmente. A continuación se explica que ha hecho cada componente de la pareja.

11.1. Jorge Almendros Fresnillo

Nada más empezar quisimos entender en qué consistieron los proyectos anteriores, ya que como se ha dicho antes ésta es la cuarta versión de un proyecto que lleva desarrollándose durante los últimos cuatro años en la Facultad de Informática. Por lo tanto nuestra primera tarea consistió en leer individualmente las memorias de los trabajos previos y posteriormente observar y entender el código generado en la versión anterior principalmente para ver qué podríamos reutilizar para esta versión, tanto en la parte del cliente como en el servidor.

Hecho esto, el siguiente paso fue migrar la parte que se ha reutilizado, adaptarla a la versión de Android utilizada y aplicarle nuestras mejoras. Jorge se encargó de gran parte del proceso de migración, debido a que se ha reutilizado una parte importante del código relativa al posicionamiento y a la medición de intensidades.

Para hacer la adaptación a personas con discapacidad se plantearon varias opciones, pero al no estar seguros si lo que nosotros pensábamos que era lo correcto era la mejor opción decidimos crear unas encuestas para que las realizaran posibles usuarios de la aplicación que sufriesen alguna de las dos discapacidades contempladas. Se crearon dos formularios, uno para cada discapacidad en los que se hacían preguntas acerca del uso de la aplicación y sobre como debía proporcionarse la guía. Los resultados fueron tenidos en cuenta a la hora de desarrollar la aplicación. Jorge fue el encargado de crear

el formulario dedicado a las personas con discapacidad auditiva.

Como se ha explicado anteriormente este proyecto tiene dos grandes partes, la parte del cliente y la parte del servidor. Jorge se centró más en la parte del cliente mientras que Luis se ha dedicado más a la parte del servidor. Esto no implica que cada uno se limite a su parte exclusivamente sino que se ha dedicado más a ella, pero siempre aportando ideas y colaborando en la otra parte.

La parte del cliente ha tenido que recibir todas las modificaciones necesarias para adaptar la aplicación a personas con discapacidad. Especialmente supuso un reto la adaptación para personas con discapacidad visual porque deben ser capaces de utilizar la aplicación a través de comandos de voz y pulsaciones de las teclas físicas del dispositivo. Jorge llevó a cabo gran parte de esta adaptación. Es cierto que en la aplicación anterior se usaba un poco la parte de reconocimiento de voz, pero solo para solicitar el destino y no de forma secuencial como se utiliza ahora, solicitando un dato tras otro y combinándose con el uso de las teclas físicas y la reproducción de voz por parte del dispositivo.

Además de la adaptación para personas con discapacidad también ha sido necesario modificar los datos que se envían al servidor. Ahora se mandan más datos que en la versión anterior ya que para generar la ruta adaptada a la discapacidad del usuario hay que mandar los datos relativos a los superables, que son los lugares por donde puede pasar un usuario, además de la discapacidad. Dicha modificación también fue realizada por Jorge.

Una parte importante del proyecto, que también conlleva un tiempo considerable, es la realización de esta memoria. A esta tarea nos dedicamos ambos componentes por igual.

Por otra parte también se han tenido que realizar muchas pruebas que debido a las características del proyecto la mayor parte de las veces se tenían que realizar en el edificio elegido. Esta parte del desarrollo del proyecto ha sido siempre realizada en conjunto, ya que debía haber concordancia entre las dos partes.

Por último, la guía para personas con discapacidad requería de un desarrollo concreto y de adaptación en función de la discapacidad. Las rutas generadas son diferentes para cada usuario concreto, y son generadas en el servidor. Sin embargo en la parte del cliente también hacían falta modificaciones para ofrecer estas instrucciones por voz o por texto en función de la discapacidad. Esta parte también ha requerido trabajo por parte de los dos miembros del grupo.

11.2. Luis Hernández Donadeu

Al ser una evolución de proyectos anteriores el primer paso consistió en ver el estado en el que estaba el proyecto. Este estudio se realizó sobre la

documentación facilitada por los tutores y se llevó de forma personal por ambos integrantes del equipo, poniendo en común en la siguiente reunión sus puntos de vista y conclusiones.

En estas conclusiones se vio que el proyecto estaba dividido, por razones de diseño, en dos partes bien diferenciadas, el cliente y el servidor, siendo el último en el cual Luis enfocó el trabajo realizado en el proyecto, colaborando con el cliente en la fases de comunicación entre ambos.

Tras el estudio de la documentación se pasó al estudio de la parte concreta de cada uno de los miembros. Luis estudió la organización de los fichero PHP, la base de datos y la aplicación Java, viendo como funcionaba todo y viendo qué alternativas posibles había. Tras poner todos los conocimientos en conjunto se decidió qué pasos debían hacerse para esta fase de la evolución del programa. Debido a que ninguno de los integrantes del equipo había trabajado antes con personas con discapacidades se tomó la decisión de crear un formulario consultivo para cada una de las discapacidades abordadas. Luis realizó el formulario para las personas con discapacidades visuales y cuando estuvieron listos estos formularios se los facilitamos a diversas organizaciones enfocadas a este tipo de personas.

Con los resultados obtenidos fijamos el rumbo seguido. Para ello se modificó la base de datos actual para aceptase las discapacidades con las que trabajaríamos y los obstáculos presentes en la facultad que podrían suponer un problema. Para este objetivo hubo que modificar los ficheros PHP para que transmitieran dichos cambios entre la base de datos y el cliente.

La parte más laboriosa del servidor vino en el proceso de generar las rutas con las discapacidades. Para ello contamos con un aplicación Java que se comunica mediante un socket con el cliente, y recibe de éste la discapacidad y el destino al que se quiere dirigir la persona que lo está usando. Se tuvieron en cuenta los obstáculos para ver si la ruta generada contenía algunos de estos obstáculos y si era así se procedía a la recalculación del camino sin tener en cuenta este punto. Por su parte la discapacidad se usa a la hora de enviar al cliente las instrucciones con la ruta. Las instrucciones son generadas para cada tipo de discapacidad siguiendo los resultado obtenidos en los formularios.

Para probar el cliente era necesario estar en la facultad. Por ello para probar la integración de ambas partes nos reunimos en la facultad. En estas pruebas se controlaba que todos los elementos funcionasen bien y que la comunicación entre cliente y servidor fuese correcta.

Otra parte fundamental fue el desarrollo de esta memoria, en la cual se trabajó de forma conjunta en diferentes apartados.

Apéndice A

Manual de instalación

Este apéndice explica cómo instalar la aplicación en un dispositivo móvil además de como instalar el proyecto en el Eclipse o en Android Studio para realizar modificaciones al mismo.

A.1. Requisitos

Sólo existen dos requisitos para poder utilizar GuiDIn. En primer lugar se requiere de un dispositivo móvil con sistema operativo Android 2.3 o superior que disponga de conexión a internet mediante wi-fi, y en segundo lugar se necesita que esté establecida una conexión a internet, ya que no solo hace falta que esté encendido el wi-fi para recoger la información de las redes wi-fi, si no que la conexión es necesaria para el inicio de sesión y el envío de la información al servidor.

A.2. Instalación en Android Studio

La instalación de la aplicación cliente se tiene que realizar en un entorno de desarrollo o IDE de las siglas en inglés Integrated development environment, como son Eclipse o el que hemos utilizado nosotros Android studio. Es necesario tener instalado el SDK (Software Development Kit) de Android que soporte la versión 5.0 del sistema operativo Android, versión máxima para la que ha sido desarrollada la aplicación.

Se proporcionan los archivos necesarios para poder importar el proyecto en cualquiera de los entornos de desarrollo mencionados. Los pasos a seguir para la importación son:

- Pulsar en File y seleccionar “New”
- Seleccionar la opción “Import project”
- Elegir la ruta donde se encuentra el proyecto

- Pulsar el botón OK

Hecho esto el proyecto estará importado.

A.3. Instalación en un dispositivo móvil

Si lo único que se desea es utilizar y probar la aplicación, la instalación en un dispositivo móvil es tremendamente sencilla. Una vez ya importado el proyecto en Eclipse o Android Studio como se ha explicado en el apartado anterior, se seleccionaría la opción exportar aplicación a un archivo apk. Luego solo habría que trasladar este archivo al dispositivo e instalarlo. Para hacerlo solo hay que pulsar sobre este archivo y a continuación se mostrara una pantalla como se muestra en la Figura A.1

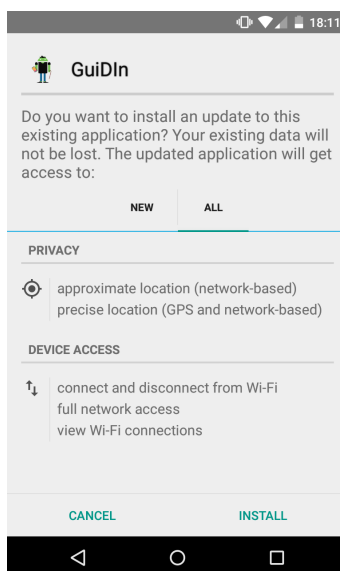


Figura A.1: Instalación en el dispositivo móvil

A.4. Instalación del servidor

El servidor no requiere de ningún tipo de modificación para probar la aplicación, todo lo necesario para el funcionamiento de la aplicación ya está instalado y se encuentra ejecutándose en el servidor de la facultad de informática. Si se quiere hacer alguna modificación en el funcionamiento del servidor, se explica en el apéndice B como se debe hacer.

Apéndice B

Extensión de la aplicación

La extensión de GuiDIn es una tarea relativamente sencilla. Su estructura no es compleja y es fácilmente modificable para adaptarse a otros edificios y otras discapacidades. A continuación se detallan los pasos a seguir.

B.1. Conexión cliente servidor

Es necesario que la aplicación se ejecute contra un servidor en el que este corriendo la aplicación que lleva a cabo el cálculo de las rutas.

Lo primero que se debe hacer es modificar esta aplicación ligeramente para poder realizar la conexión entre el cliente y el servidor como se explica en el capítulo 6.

Se importará en Eclipse el proyecto CalcularRuta, y en la clase Main, dentro del paquete principal hay que modificar la variable PORT de tipo int, se le asignará el puerto que se use para la conexión.

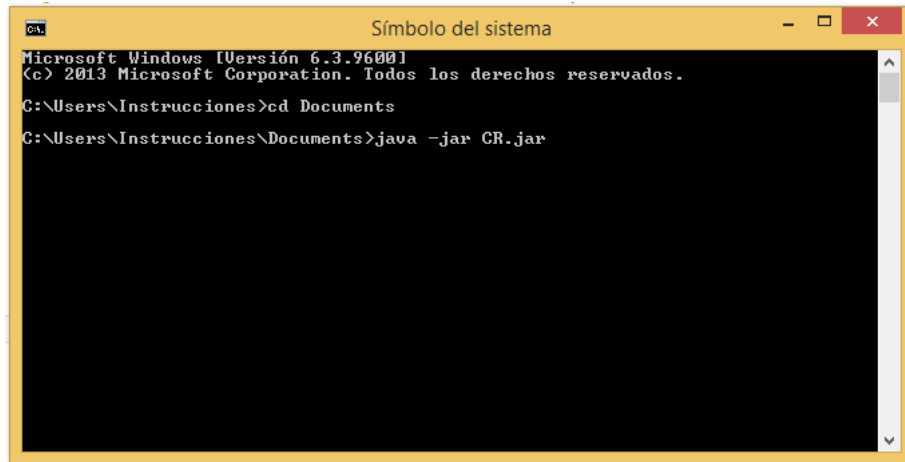
Una vez establecido, exportaremos el proyecto a un fichero .jar ejecutable, que tendremos que alojar en un directorio de nuestro servidor.

Por último, para que la aplicación se mantenga ejecutándose y esperando a posibles clientes se deben de seguir los siguientes pasos:

- Instalar la máquina virtual de JAVA[X] en nuestro servidor
- Abrir una consola de comandos (cmd) en Windows y situarnos en la carpeta donde se encuentre el jar generado.
- Escribir: `java -jar NombreAplicacion.jar` y pulsar ENTER

El resultado de estas instrucciones sería el que se muestra en la Figura B.1. De esta forma ya tendremos nuestra aplicación servidor escuchando, lista para recibir clientes.

A continuación tendremos que modificar nuestra aplicación cliente para establecer tanto la dirección en la que se encuentra alojada en nuestro servi-



```
Microsoft Windows [Versión 6.3.9600]
(c) 2013 Microsoft Corporation. Todos los derechos reservados.
C:\Users\Instrucciones>cd Documents
C:\Users\Instrucciones\Documents>java -jar CR.jar
```

Figura B.1: Lanzamiento de la ejecución en consola

dor, como el puerto de escucha. Para ello, importamos el proyecto en Android Studio, o Eclipse y nos vamos al paquete sockets y en la clase Client.java modificaremos las siguientes variables:

- `String adresaServer = 'dirección servidor';`
- `int PORT = XX;` Ha de ser el mismo que el establecido en la aplicación servidor

B.2. Crear base de datos

Como se explica en el capítulo referente a la Base de Datos (8.2), es necesario disponer de una base de datos, tanto para almacenar los usuarios como los datos de las mediciones de intensidad wi-fi. Esta base de datos debe mantener la estructura detallada en el dicho capítulo.

Por otra parte también será necesario alojar en el servidor los archivos php necesarios para la conexión entre la aplicación y la base de datos.

B.3. Enlace cliente - base de datos

El cliente debe modificar los archivos que contienen la información de la conexión con la base de datos, en concreto hay que modificar la clase `HttpServices` que se encuentra en el paquete `http`. Se modificarán las variables:

- `String Host:` se le asignará la dirección IP del hosting elegido.
- `String Port:` contendrá el puerto del hosting

- String urlServer: contendrá algo similar a 'http://direccionWEB';

Por otra parte también será necesario modificar todos los archivos php que deben estar alojados en el servidor. Se modificará la variable conexión y quedará de la siguiente forma:

```
$conexion = mysql_connect('dir IP','usuario BBDD','contraseña BBDD');
```

B.4. Medir intensidades wi-fi

Una vez creada la base de datos ya es posible utilizar la aplicación cliente para llevar a cabo la medición de intensidades wi-fi. El uso de esta funcionalidad se detalla en el Capítulo 7.5.5.3.

Será necesario definir un eje de coordenadas x,y,z sobre el edificio que se quiera mapear, siempre siendo el eje z el indicador de planta en la que se está midiendo la intensidad. Cuanto mayor sea el número de intensidades medidas mayor será la precisión obtenida por la aplicación

B.5. Creación de los XML relativos a la estructura del edificio

El edificio está representado mediante unos cuantos archivos XML. En concreto debe existir uno que defina el edificio (edificio.xml) y adicionalmente un archivo por cada planta que se quiera representar.

El XML del edificio tiene la función de índice, basta con añadir la planta que queramos incluir.

Según esta representación existirán los archivos nombreFichero.xml que contendrán las descripción de cada planta. La estructura de las plantas se encuentra en la sección de XML anteriormente mencionada.

Estos ficheros deberán estar alojados en el servidor, dentro de una carpeta llamada "xml", en el mismo directorio en el que se encuentre la aplicación .jar.

Para ver un ejemplo del formato de los XML así como información de todos los campos disponibles se puede ver la sección propia de los ficheros en el capítulo del servidor 8.4.1.

B.6. Establecer destinos

Todos los posibles destinos dentro de un edificio son almacenados en un fichero JSON. La estructura es la siguiente:

```
[  
{ 'lugar'      : 'sala de juntas',  
  'cuadrante' : '14', },
```

```
{ 'lugar'      : 'aula 13',  
  'cuadrante' : '15', }  
]
```

Lugar representa la estancia a la que nos referimos. Cuadrante hace referencia al identificador del cuadrante en el que se encuentra el destino. El fichero ha de llamarse `destinos.json` y estará alojado en el servidor, en el mismo directorio que la aplicación `.jar`.

B.7. Compilar y ejecutar la aplicación

Por último, con la aplicación servidor ya compilada y corriendo en un `.jar`, se compilará la aplicación cliente con el Eclipse o el Android Studio y se generará el archivo `.apk` el cual será enviado al dispositivo Android donde se instalará y ejecutará.

B.8. Añadir discapacidades

Añadir discapacidades es relativamente sencillo. En la parte del cliente habría que añadir la posibilidad de que el usuario al registrarse pueda seleccionarla.

Además si las formas ya existentes en las que se muestra la información de la ruta no se adaptan a la nueva discapacidad habría que añadir esta nueva funcionalidad. Se modificaría la clase `indicarDestino.java` en la que se tendría en cuenta la discapacidad para generar la información de la manera más adecuada a la discapacidad.

En el servidor no es necesario hacer ninguna modificación, la ruta se genera igualmente en función de los superables que haya seleccionado el usuario y se adaptará perfectamente a la discapacidad.

Apéndice C

Formularios pasados a personas con discapacidades

A continuación se muestra el formulario relleno por personas con discapacidad visual en la Figura C.1 y el formulario relleno por personas con discapacidad auditiva en la Figura C.2.

Estos formularios han servido para obtener información relativa a cómo dar las instrucciones de guía, para poder adaptarlas lo máximo posible a cada discapacidad además de para conseguir información muy valiosa acerca de como puede utilizar un discapacitado visual o auditivo una aplicación móvil.

¿Cómo le gustaría ser guiado un discapacitado visual?

GuiDIn es una aplicación pensada para ayudar a personas discapacitadas a llegar a un destino concreto dentro de un edificio. La aplicación será capaz de posicionarle en todo momento y de guiarle hasta el destino introducido de una forma u otra en función de la discapacidad que sufra. Con este cuestionario se pretende facilitar lo máximo posible el uso de la aplicación, y que esta se ajuste lo mejor posible a cualquier discapacidad.

*Obligatorio

Respecto a la guía, prefiere: *

- Instrucciones en lenguaje natural que le indiquen el camino a seguir, (ej: gire a la derecha)
- Sistema de pitidos en un oído u otro dependiendo de la dirección a tomar
- Combinación de las 2 opciones anteriores
- Otras opciones

Si ha elegido otra indíquelas:

¿Cómo preferiría desplazarse por los menús de la aplicación? *

- Teclas de volumen y que una voz vaya leyendo las opciones
- Lectura por voz de las opciones disponibles en la pantalla actual en la que se encuentre de la aplicación y selección también por voz
- Otras opciones

Si ha elegido otra indíquelas:

Sugerencias, opciones, o cualquier cosa que debería tener la aplicación:

Figura C.1: Formulario para personas con discapacidad visual.

¿Cómo le gustaría ser guiado un discapacitado auditivo?

GuiDIn es una aplicación pensada para ayudar a personas discapacitadas a llegar a un destino concreto dentro de un edificio. La aplicación será capaz de posicionarle en todo momento y de guiarle hasta el destino introducido de una forma u otra en función de la discapacidad que sufra. Con este cuestionario se pretende facilitar lo máximo posible el uso de la aplicación, y que esta se ajuste lo mejor posible a cualquier discapacidad

*Obligatorio

Respecto a la guía, prefiere: *

- Que se le indique por texto los pasos a seguir para llegar al destino
- Visualizar su posición en un mapa junto con una ruta dibujada que explique cómo llegar
- Mezcla de las 2 opciones anteriores
- Otras opciones

Si ha elegido otras opciones, indíquelas

En la explicación de cómo llegar al destino, ¿le es útil que se le indique información sobre los lugares por los que va pasando para asegurarse de que va por el camino correcto? (ej: a su derecha verá una ventana) *

- Sí
- No

Como preferiría elegir el destino *

- Por voz
- Por teclado
- Viendo el mapa del edificio y señalando el punto exacto
- Cuantas más opciones mejor
- Otras opciones

Si ha elegido otras opciones, indíquelas

Sugerencias, opciones, o cualquier cosa que debería tener la aplicación

Figura C.2: Formulario para personas con discapacidad auditiva.

Bibliografía

- Arora, Sanjeev; Barak, B. (2009). *Computational Complexity: A Modern Approach*. Cambridge Press.
- Chen, Y. and Kobayashi, H. (2002). *Signal strength based indoor geolocation*. Proceedings of the IEEE International Conference on Communications.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2001). *Introduction to Algorithms*. MIT Press and McGraw-Hill.
- Department of Energy Quality Managers, Software Quality Assurance Subcommittee (1999). Software risk management a practical guide. *Office of the Chief Information Officee, In Press*.
- E. López Mañas, F. J. Moreno y J. Plá Herrero (2010). *Avanti: Sistema de asistencia a la evacuación de incendios*.
- Google (2014). *Google maps indoor*.
- Google (2015a). *Android*.
- Google (2015b). *Historia de android*.
- Google (2015c). *Texttospeech*.
- Infsoft (2015). *Infsoft*.
- Mariana Martín-Calderín de la Villa (2013). *Sistema de guía por voz en interiores*.
- Matsudaira., K. (2012). *Scalable web architecture and distributed systems*.
- Oracle (2015). *Mysql documentation*.
- Redpin (2008). *Redpin*.
- Retana, J. A. C. (2010). Sistema de posicionamiento en interiores mediante balizas bluetooth. *Universidad Pontificia Comillas Escuela técnica superior de ingeniería*.

Richardson, L. S. R. (2007). RESTful web service. O'Reilly Media.

Tamada, R. (2014). Android speech to text tutorial.

Víctor Gutiérrez Rodríguez, Juan Diego Lozano Martín y Víctor Manuel Pose Murga (2014). Generador interactivo de instrucciones de guía sobre plataformas móviles.

Wikipedia. Network socket.

Wikipedia. Sistema de posicionamiento global.

Wikipedia. Triangulación.

Wikipedia. Wi-Fi positioning system.

acr!