
Título en español (definido en Cascaras\cover.tex)
Title in English (defined in Cascaras\cover.tex)



Trabajo de Fin de Grado
Doble grado en Ingeniería Informática y Matemáticas
Curso 2019–2020

Autor
Nombre Apellido1 Apellido2

Director
Director 1
Director 2

Colaborador
Colaborador 1
Colaborador 2

Facultad de Informática
Universidad Complutense de Madrid

Título en español (definido en
Cascaras\cover.tex)
Title in English (defined in
Cascaras\cover.tex)

Trabajo de Fin de Máster en **Ingeniería Informática**
Departamento de **XXXXXXXXXXXXXX**

Autor
Nombre Apellido1 Apellido2

Director
Director 1
Director 2
Colaborador
Colaborador 1
Colaborador 2

Dirigida por el Doctor
Director 1
Director 2

Facultad de Informática
Universidad Complutense de Madrid

23 de marzo de 2020

Dedicatoria

*A Pedro Pablo y Marco Antonio, por crear TeXiS
e iluminar nuestro camino*

Agradecimientos

A Guillermo, por el tiempo empleado en hacer estas plantillas. A Adrián, Enrique y Nacho, por sus comentarios para mejorar lo que hicimos. Y a Narciso, a quien no le ha hecho falta el Anillo Único para coordinarnos a todos.

Resumen

Título en español (definido en Cascaras\cover.tex)

Un resumen en castellano de media página, incluyendo el título en castellano. A continuación, se escribirá una lista de no más de 10 palabras clave.

Palabras clave

Máximo 10 palabras clave separadas por comas

Abstract

Title in English (defined in Cascaras\cover.tex)

An abstract in English, half a page long, including the title in English. Below, a list with no more than 10 keywords.

Keywords

10 keywords max., separated by commas.

Índice

| | |
|--|-----------|
| 1. Introducción | 1 |
| 1.1. ¿Por qué la navegación por interiores? | 2 |
| 1.2. Objetivos | 3 |
| 1.3. Plan de trabajo | 3 |
| 1.4. Explicaciones adicionales sobre el uso de esta plantilla | 3 |
| 2. Estado de la Cuestión | 5 |
| 2.1. Aplicaciones de guía | 5 |
| 2.1.1. Google Maps | 5 |
| 2.1.2. BlindSquare | 7 |
| 2.1.3. Nearby Explorer | 9 |
| 2.1.4. Lazarillo | 10 |
| 2.1.5. Wayfindr | 10 |
| 2.1.6. Conclusiones | 11 |
| 2.2. Sistemas de posicionamiento | 12 |
| 2.2.1. GPS | 12 |
| 2.2.2. Wi-Fi | 13 |
| 2.2.3. Balizas Bluetooth | 13 |
| 3. Descripción del Trabajo | 15 |
| 3.1. Reunión en el Centro de Tiflotecnología e Innovación de la ONCE | 15 |
| 3.1.1. Introducción | 15 |
| 3.1.2. Entrevista | 16 |
| 3.1.3. Conclusiones | 18 |
| 3.2. Estudio de la precisión de los beacons | 18 |
| 3.2.1. Aplicación <i>miniapp</i> | 19 |
| 3.2.2. Aplicación <i>cuadrantes v1</i> | 19 |
| 3.2.3. Resultados | 19 |
| 3.2.4. Interferencias en la señal bluetooth | 21 |
| 3.2.5. Mediciones en puntos clave de la facultad | 22 |
| 3.3. Mapeo del edificio | 25 |
| 4. Diseño e implementación | 29 |
| 4.1. Servidor | 29 |

| | |
|--|-----------|
| 4.1.1. Archivos XML | 29 |
| 4.1.2. Código java | 29 |
| 4.1.3. Cálculo de la ruta óptima | 31 |
| 4.1.4. Detalles técnicos del posicionamiento | 32 |
| 4.2. Cliente | 32 |
| 5. holi | 33 |
| 6. Conclusiones y Trabajo Futuro | 35 |
| 7. Introduction | 37 |
| 8. Conclusions and Future Work | 39 |
| A. Título del Apéndice A | 41 |
| B. Título del Apéndice B | 43 |

Índice de figuras

| | |
|---|----|
| 2.1. Plano de un edificio proporcionado por Google Maps. | 6 |
| 2.2. Vista del interior del Madison Square Garden. | 7 |
| 2.3. Ejemplo de navegación y búsqueda en Google Maps Indoors. | 7 |
| 2.4. Método de triangulación GPS. | 12 |
| 3.1. Interfaz de la aplicación miniapp. | 20 |
| 3.2. Interfaz de la aplicación cuadrantes v1. | 21 |
| 3.3. Gráfico con las distancias medidas al beacon CPne. | 22 |
| 3.4. Gráfico con las distancias medidas al beacon eAaw. | 22 |
| 3.5. Gráfico con las distancias medidas al beacon 8v2c. | 23 |
| 3.6. Gráfico de los beacons 8v2c y CPne superpuestos. | 23 |
| 3.7. Mapa de la planta baja de la Facultad de Informática con la ubicación de los beacons (rojo) y los puntos de medición (verde). | 24 |
| 3.8. Mapa de la primera planta de la Facultad de Informática con la ubicación de los beacons (rojo) y los puntos de medición (verde). | 25 |
| 3.9. Mapa de la planta baja de la Facultad de Informática con la ubicación definitiva de los beacons (rojo) en el hall. | 26 |
| 3.10. Primera versión del mapeo de la primera planta de la Facultad de Informática. | 27 |
| 3.11. Versión final del mapeo de la primera planta de la Facultad de Informática. | 27 |
| 3.12. Mapeo de la planta baja de la Facultad de Informática. | 28 |
| 4.1. Diagrama de las clases principales del servidor. | 31 |

Índice de tablas

Introducción

“Frase célebre dicha por alguien inteligente”

— Autor

En la actualidad, los *smartphones* se han convertido en los protagonistas indiscutibles de nuestro día a día. El informe anual de *Ditrendia* (?) recoge que el 68 % de la población mundial (5100 millones) cuenta con un *smartphone*, mientras que este porcentaje asciende al 96 % cuando hablamos de la población española. Es decir, aproximadamente 32,6 millones de españoles navegan por Internet a diario con su teléfono móvil.

Por otro lado, resulta prácticamente imposible imaginar un *smartphone* que no tenga instalado una aplicación de navegación, ya que este tipo de aplicaciones se han convertido en herramientas esenciales y habituales, puesto que no solo se limitan a dar la ruta óptima entre dos puntos, sino que también aportan distintas alternativas para ir a pie, con transporte público, información sobre el mismo como horarios, cambios temporales, etc. Se estima que el 75 % de los usuarios españoles utilizan aplicaciones de navegación mensualmente, siendo la tercera actividad más practicada después de la mensajería instantánea y la visualización de videos online (?).

No cabe duda de lo útil que resulta poder consultar la ruta entre dos puntos pero, ¿estas aplicaciones son igual de apropiadas para todos los usuarios? ¿Se tienen en cuenta las necesidades de aquellos que padecen discapacidad visual? En España, 70.775 personas sufren *ceguera legal* según la ONCE (?). Este término engloba dos tipos marcados y diferenciados, lo que se conoce como ceguera (ausencia de visión o solo percepción de luz) y la deficiencia visual (mantenimiento de un resto de visión funcional para la vida cotidiana). En ambos casos, las personas que las padecen afrontan numerosos desafíos en su vida diaria, la mayor parte de ellos derivados de la total falta de información. Un vistazo a nuestro alrededor es suficiente para darnos cuenta de cuán visuales son la mayor parte de los mensajes útiles que usamos en nuestro entorno: desde leer la etiqueta de un producto en el supermercado, hasta saber si nos encontramos en la parada de autobús correcta. De ahí que los ojos sean considerados los principales órganos sensoriales y que su pérdida conlleve una reducción considerable de independencia, ya que el acceso a la información significa autonomía. En ocasiones, esto viene acompañado de un segundo problema con el que muchos están acostumbrados a lidiar, el exceso de protección. A menudo familiares, amigos o incluso desconocidos asumen que un invidente necesita ayuda sin preguntar o sin esperar a ser llamados. Este frecuente comportamiento genera impotencia en el individuo en lugar de independencia y le quita espacio para aprender a realizar una tarea por sí mismo.

En resumen, la falta de accesibilidad es el eje central del que nacen numerosos problemas que afectan a la vida de las personas que presentan ceguera legal. Otro ejemplo más es

el caso del ocio y la tecnología. No abundan los libros adaptados. De hecho, según la *World Blind Union* “más del 90 % del material publicado no es accesible para invidentes o personas con deficiencia visual”(?). E igual ocurre con Internet. El grueso de las páginas web y aplicaciones no consideran las necesidades especiales de estos potenciales usuarios, dejándoles completamente al margen. Por ello, la respuesta a las dos preguntas lanzadas al comienzo de esta sección es no, actualmente son pocas las aplicaciones que tienen en cuenta a las personas que sufren discapacidad visual y, en particular, son pocas las aplicaciones de navegación que están adaptadas. Es por esto, que en nuestro trabajo de fin de grado hemos querido abordar este problema, estudiando, para ello, tecnologías accesibles que nos permitan desarrollar una aplicación de navegación que facilite una guía adaptada para estos usuarios.

1.1. ¿Por qué la navegación por interiores?

Como es natural, todos nos vemos obligados a desplazarnos en nuestro día a día. Normalmente suelen ser lugares conocidos a los que llegamos de una manera más o menos automática, sin tener que pensar mucho ya que conocemos y memorizamos todo lo que hay en dichos recorridos. Sin embargo, de manera puntual modificamos dichas rutinas, ya sea por problemas temporales que inhabilitan la ruta en cuestión, como por la necesidad de desplazarnos a un lugar al que no habíamos ido antes. Paralelamente, hay un conjunto de edificios que visitamos con cierta frecuencia y que por ende nos resultan familiares y donde nos ubicamos perfectamente, pero en ocasiones nos surge la necesidad de ir a otros por primera vez, véase un hospital, un museo o un centro comercial.

A menudo estas situaciones despiertan desorientación, incomodidad y rechazo en las personas que las viven ya que se encuentran frente a una situación de descontrol e incertidumbre debido a la falta de conocimiento. A nadie le gusta sentirse perdido, pero cuando te falta uno de los cinco sentidos y uno de los más esenciales, la vista, esto se vuelve mucho más duro, ya que, hay un gran vacío informativo. Basta pensar en cuántas personas se te cruzan por la calle, cuántos obstáculos sorteas a diario tanto en interiores como en exteriores, cuántas veces cruzas la carretera para alcanzar tu destino, cuántas veces te apoyas leyendo el nombre de una calle o un cartel en un edificio, cuántas veces bajas/subes unas escaleras o esperas al ascensor, o miras el número del autobús que está por llegar... Ahora imagina hacerlo sin ayuda de la vista. Todo esto son ejemplos de situaciones muy cotidianas que para las personas videntes no suponen ningún esfuerzo mientras que para las personas con discapacidad visual suponen un gran reto.

En los últimos años se ha estudiado mucho el sector de la navegación por exteriores, actualmente son varias las apps que mediante el GPS proporcionan una guía de origen a destino. Este hecho, acompañado de la creciente sensibilización con el problema de la ceguera y del *boom* de las tecnologías accesibles ha favorecido que cada vez más desarrolladores se interesen por la accesibilidad y la promuevan en este tipo de aplicaciones. Sin embargo, en la navegación por interiores aún vemos un claro vacío ya que es un terreno menos explorado en general y, consecuentemente, menos adaptado. Por ello, hemos decidido centrar nuestras investigaciones en este sector, buscando paliar el malestar al que estos usuarios se enfrentan en su día a día. Para la consecución de este fin, desarrollaremos una aplicación accesible que sirva de guía a invidentes por espacios interiores, más concretamente por la Facultad de Informática de la Universidad Complutense de Madrid.

1.2. Objetivos

Descripción de los objetivos del trabajo.

1.3. Plan de trabajo

Aquí se describe el plan de trabajo a seguir para la consecución de los objetivos descritos en el apartado anterior.

1.4. Explicaciones adicionales sobre el uso de esta plantilla

Si quieres cambiar el **estilo del título** de los capítulos, edita `TeXiS\TeXiS_pream.tex` y comenta la línea `\usepackage[Lenny]{fncychap}` para dejar el estilo básico de L^AT_EX.

Si no te gusta que no haya **espacios entre párrafos** y quieres dejar un pequeño espacio en blanco, no metas saltos de línea (`\\`) al final de los párrafos. En su lugar, busca el comando `\setlength{\parskip}{0.2ex}` en `TeXiS\TeXiS_pream.tex` y aumenta el valor de `0,2ex` a, por ejemplo, `1ex`.

TFMTeXiS se ha elaborado a partir de la plantilla de TeXiS¹, creada por Marco Antonio y Pedro Pablo Gómez Martín para escribir su tesis doctoral. Para explicaciones más extensas y detalladas sobre cómo usar esta plantilla, recomendamos la lectura del documento `TeXiS-Manual-1.0.pdf` que acompaña a esta plantilla.

¹<http://gaia.fdi.ucm.es/research/texis/>

Capítulo 2

Estado de la Cuestión

2.1. Aplicaciones de guía

En los últimos años ha aumentado la sensibilización tecnológica en áreas de inclusión a usuarios con discapacidad visual. De modo que las tecnologías accesibles tienen, cada vez más, un papel central en el desarrollo de aplicaciones, logrando recortar las limitaciones que antes las separaban de las personas que sufren algún tipo mayor de dificultad visual y dirigiéndose a un público más amplio.

Al igual que las personas videntes, las personas con ceguera son usuarios de aplicaciones de muy variada índole, por ello encontramos apps ya adaptadas en categorías como: redes sociales, entretenimiento, lectura, identificación de colores y objetos, etc.

En esta sección, haremos un pequeño estudio sobre las aplicaciones accesibles existentes en el campo de la navegación, bien sea por interiores o exteriores, y su funcionamiento.

2.1.1. Google Maps

El pasado 10 de Octubre de 2019, en el “World Sight Day”, Google dió a conocer la última actualización de la famosa aplicación *Google Maps*¹. Esta incluiría una nueva característica desarrollada desde cero por y para personas con discapacidad visual que convertiría a la misma en una app accesible.

El proyecto consiste en la implementación de una nueva funcionalidad que facilita la posibilidad de recibir instrucciones de voz más detalladas y nuevos tipos de anuncios verbales muy útiles para las rutas de a pie para personas con visibilidad reducida. Algunas de las nuevas instrucciones incluidas son: informar de manera proactiva que estás en la ruta correcta, la distancia hasta el próximo giro, la dirección en la que estás caminando, avisos para cruzar con precaución si te aproximas a una gran intersección, notificaciones en caso de ser redirigido por causa de haber abandonado accidentalmente la ruta correcta, etc. De esta manera, la aplicación pretende brindar de independencia a las personas que padecen ceguera tratando de que se sientan cómodas y seguras a la hora de explorar lugares nuevos y desconocidos. La guía de voz detallada para la navegación está actualmente en desarrollo, estando ya disponible en inglés en los Estados Unidos y en japonés en Japón. Su soporte para otros idiomas y países sigue en camino.

En cuanto a la navegación por interiores, *Google Maps*² con su actualización 6.0 incor-

¹<https://blog.google/products/maps/better-maps-for-people-with-vision-impairments/>

²<https://www.google.es/intl/es/maps/about/partners/indoormaps/>



Figura 2.1: Plano de un edificio proporcionado por Google Maps.

poró los primeros planos de ciertos edificios públicos, entre los cuales destacan aeropuertos, centros comerciales, estadios y puntos de transporte público. Gracias a esta nueva versión, *Google Maps* ayuda a determinar dónde estás, en qué planta y hacia dónde ir. Para ello, basta con hacer zoom sobre un edificio cuyo plano esté disponible en la app, y este aparecerá automáticamente y completamente detallado. En la Figuras 2.1 y 2.2 vemos un ejemplo del famoso Madison Square Garden de Nueva York.

Con estos nuevos planos podrás localizar dónde están los baños, escaleras, ascensores, entradas y salidas, etc., los cuales aparecen representados mediante iconos globalmente aceptados (ver Figura 2.1). También aparecen detallados los distintos establecimientos que se localizan en el edificio e incluye la posibilidad de hacer ciertas búsquedas, tanto generales (de cafeterías, librerías, tiendas, restaurantes...) como concretas (Starbucks, McDonald's...) (ver Figura 2.3). Otra funcionalidad que no falta en la versión de interiores es la posibilidad de señalar un destino y recibir indicaciones sobre cómo llegar a él. Para ello, aparece el habitual punto azul que te acompaña e indica tu posición, actualizando el plano con cada movimiento que lles a cabo (incluidos cambios de una planta a otra) (ver figura 2.3). Esta aplicación es un proyecto colaborativo y por ende, desde la web es posible actualizar y subir nuevos planos. Está disponible tanto para ordenador como plataformas Android e iOS.

Esta aplicación pone a nuestro servicio la utilidad de *Maps* pero en interiores. Además, nos permite colaborar, pudiendo subir nosotros mismos el plano de un edificio. Sin dudar del gran avance que esta aplicación supone en la navegación por interiores, no debemos olvidar algunas de sus desventajas: el posicionamiento, al contrario que en exteriores, no es muy preciso (en la web hablan de varios metros), y las búsquedas que puedes realizar son limitadas, no pudiendo, por ejemplo, preguntar por la ubicación de los baños; esto es, puedes ver dónde están pero no puedes seleccionarlos como destino para que te vaya indicando la ruta a seguir. Pero sobre todo, tiene el inconveniente de que no es una tecnología



Figura 2.2: Vista del interior del Madison Square Garden.

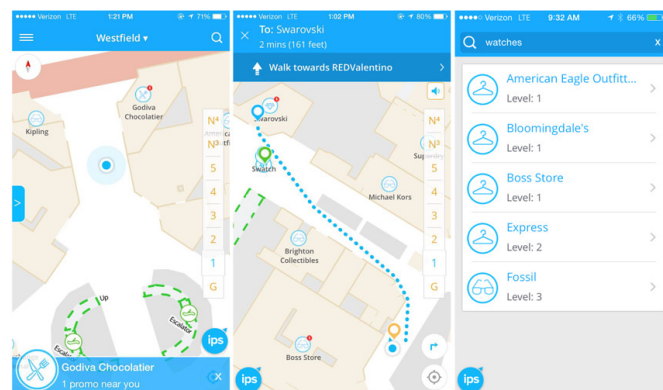


Figura 2.3: Ejemplo de navegación y búsqueda en Google Maps Indoors.

accesible: *Google Maps Indoors*³ es una aplicación completamente visual que no cuenta con soporte auditivo por lo que descarta completamente a usuarios invidentes.

2.1.2. BlindSquare

Es una de las aplicaciones de navegación más populares. Su uso se extiende a más de 130 países y está habilitada en 25 idiomas, entre los cuales se incluye el español. Esta aplicación, desarrollada para iOS y diseñada para personas con discapacidad visual, proporciona una guía completa, de origen a destino, tanto en exteriores como en interiores. Además, describe el entorno y anuncia posibles puntos de interés para el usuario (como pueden ser los lugares considerados populares o aquellos visitados frecuentemente). Su principal característica es que permite interactuar mediante voz gracias al controlador de música de Apple.

*BlindSquare*⁴ determina tu posición mediante localización *GPS* y, a partir de ahí, puede darte información sobre las proximidades utilizando *Foursquare* y *OpenStreetMap*. De este modo, es capaz tanto de guiarte a un cierto destino como de notificarte qué establecimientos hay en tu radio: restaurantes a 200m, parques más cercanos, farmacias...

Con el fin de agilizar el uso de la app, y que por tanto esta sea cómoda y rentable para

³https://www.youtube.com/watch?v=cPsTWj_03Qs

⁴<https://www.blindsquare.com>

los usuarios finales, incluye: accesos directos a funciones mediante gestos (como sacudir el móvil para que nos diga la ubicación actual y puntos cercanos) y la posibilidad de establecer filtros para recibir únicamente el tipo de información deseada. Por ejemplo, permite filtrar por restaurantes para no tener notificaciones sobre estaciones de tren o librerías.

En cuanto a la navegación por interiores, *BlindSquare*⁵ emplea un sistema de balizas bluetooth, llamadas *beacons*, que colocan en sitios estratégicos de los edificios, para solventar el problema del posicionamiento. Por lo demás, incluye las mismas posibilidades y funcionalidades que la navegación por exteriores, con la única limitación de que el edificio debe estar provisto de dichos sistemas de posicionamiento.

En su web encontramos un ejemplo de la utilización de los *beacons* en un campus⁶: una vez que entras en el edificio, uno de los *beacons* se dará cuenta de tu aplicación BlindSquare y te hará saber dónde te encuentras y cómo llegar a tu destino, indicándote los ascensores, escaleras e intersecciones más cercanas. Integrar en el campus servicios como estos promueve tanto a visitantes como a estudiantes con discapacidad visual moverse por el entorno con total autonomía y seguridad.

Entre los puntos fuertes de esta aplicación destacamos los siguientes:

- Da información sobre los metros que quedan hasta llegar a un determinado objetivo. Resulta útil porque si van disminuyendo sabes que vas por el camino adecuado.
- Utiliza indicaciones de tipo reloj (a las 10, a las 3,...) muy usadas por las personas con discapacidad visual.
- Avisa de las intersecciones.
- Cuando te da una nueva indicación y la superas, usa el sonido asociado a correcto o *check*. Así, puedes seguir sin preocuparte. Si por el contrario te equivocas, reproduce un sonido en consecuencia.
- Se pueden añadir ubicaciones en una lista de lugares marcados.
- Puedes ir girando con el móvil y te va indicando lo que tienes enfrente.
- También tiene opción de simulación, que permite prepararse un camino antes de ir.
- Te permite ser más autónomo y descubrir nuevos sitios.
- A la hora de desplazarte te indica las distintas alternativas por adelantado. Esto es, mientras que para espacios exteriores te señala la posible ruta utilizando transporte público, privado, a pie, etc. Para espacios interiores te especifica, siempre que la haya, la opción de utilizar escaleras, ascensor, escaleras mecánicas, etc., de esta manera te proporciona una idea global del espacio y de las distintas vías que puedes seguir para llegar a tu destino.
- Permite llevar las manos libres.
- Incluye un lector de códigos QR, es más cómodo porque puede dar más información que la línea braille.

Su principal punto negativo es el precio, ya que cuesta 40 libras.

⁵<https://www.youtube.com/watch?v=9jH-BdjmgB4>

⁶<https://www.blindsquare.com/2019/11/01/blindsquares-getting-straight-as-on-campus/>

Al contrario que la aplicación anterior, esta sí es una aplicación diseñada para personas con discapacidad visual. Las diferencias saltan a la vista: el modo de dar las indicaciones, avisos constantes para indicarte si vas por el camino correcto, permite más autonomía; gracias a la comunicación constante que ofrece permite llevar las manos libres, entre otras. Parece imposible pensar que el interior de un edificio pueda resultar menos seguro que una gran avenida, lo cierto es, que para personas con discapacidad visual, muchas veces es así. El interior de un gran centro comercial o una biblioteca resultan un laberinto cuando se va por primera vez, más aún si tenemos algún tipo de dificultad para leer las indicaciones que, normalmente, suelen estar en lugares altos y no adaptadas para personas con discapacidad visual. Lo que se pretende con esta aplicación es mantener la autonomía del usuario tanto dentro como fuera de un edificio ⁷.

2.1.3. Nearby Explorer

*Nearby Explorer*⁸ es otra de las aplicaciones que encuadramos en el campo de la navegación accesible por interiores y exteriores. Está habilitada tanto para Android como para iOS y su descarga se encuentra disponible en el *App Store* de manera gratuita.

La guía por exteriores se basa en la misma idea que *BlindSquare*, y por ende funciona de manera similar. Entre sus características destacan: la posibilidad de ejecutar ciertas acciones poniendo el móvil en distintas posiciones, como por ejemplo, inclinarlo verticalmente para que funcione como una brújula; y, la capacidad de filtrar la información de modo que ésta se adapte completamente a las necesidades del usuario. Entre la información que *Nearby Explorer* puede proporcionar a sus usuarios encontramos los lugares cercanos a la ubicación actual, los nombres de las calles por las que pasa, los números de los bloques de las calles por las que pasa, la distancia que hay al destino desde un punto de referencia (como casa, trabajo...), etc. Además de la posibilidad de filtrar la información deseada, las indicaciones por audio pueden ser pausadas en cualquier momento de modo que no interfieran con otras señales auditivas (como las paradas en un autobús, por ejemplo). Otra gran funcionalidad con la que cuenta *Nearby Explorer* es la de explorar una ruta por adelantado, sin tener que estar físicamente en el sitio, pudiendo incrementar o decrementar el radio de exploración.

Por otro lado, vemos que la navegación por interiores se basa en un sistema de *beacons* que sustituye a las señales GPS y se encarga de solventar el problema del posicionamiento en interiores. Pueden configurarse de dos maneras: *ad hoc* y *mapeo completo*.

En el caso de la configuración *ad hoc*, cuenta con la ventaja de que tiene una instalación muy sencilla pero aparecen los siguientes problemas:

- No se puede determinar la ubicación exacta de un *beacon*.
- No se puede obtener información del entorno a menos que te encuentres dentro del radio de detección de un *beacon*.
- Tienes que habilitar cierto soporte para detectar los *beacons* (no se detectan de manera automática).

Por el contrario, el *mapeo completo* es más robusto por lo que su instalación es más compleja pero a cambio nos proporciona una localización precisa del dispositivo por lo que tiene un comportamiento similar al de otras aplicaciones.

⁷<https://www.blindsquare.com/2019/10/24/independence-on-both-sides-of-the-door/>

⁸<https://play.google.com/store/apps/details?id=org.aph.nearbyonline&hl=es>

Desde el punto de vista de la navegación por exteriores es una aplicación completa y fácil de utilizar, que incluye una interfaz sencilla y trata de adaptarse siempre a las distintas necesidades o situaciones del usuario mediante un opciones configurables. Además, cuenta con una versión gratuita (algo poco común en aplicaciones de tal categoría) que aunque no incluye todas las funcionalidades de la versión de pago te permite probarla y familiarizarte con ella antes de tomar una decisión final. Por otro lado, la funcionalidad de navegación *indoors* no está tan desarrollada y su uso está supeditado exclusivamente a aquellos lugares que cuenten con la instalación necesaria y hayan incluido sus datos en OpenStreetMap, configurando el espacio en nodos, aristas y relaciones. Esta tarea es tediosa y a menudo parte de cero por lo que son pocos los edificios que actualmente están mapeados y pueden aprovechar la app.

2.1.4. Lazarillo

Lazarillo (?) es una aplicación que actualmente solo proporciona guía para exteriores. Inicialmente la idea era cubrir también la navegación por interiores pero su desarrollo no fue posible por problemas de financiación.

La navegación por exteriores cuenta con las funcionalidades básicas que ya hemos mencionado en las apps anteriores:

- Buscar lugares de interés, cercanos a la ubicación actual. Esta búsqueda se puede acotar filtrando por categorías que vienen predefinidas (transporte, bancos y cajeros, salud, comida, tiendas, etc.).
- Buscar una dirección específica a partir de la cual se desplegarán todas las posibles rutas (a pie, en transporte público, privado, etc.) y una vez seleccionada la ruta deseada, comenzarán las indicaciones mediante audio con la información pertinente (metros, giros a derecha e izquierda, etc.).
- Guardar una lista de lugares favoritos.
- Posibilidad de rastrear una dirección, previamente marcada con la opción “Seguir este lugar”, de modo que con independencia de a dónde nos estemos dirigiendo se activará una alerta a medida que nos acerquemos a dicha ubicación.
- Ajustar la configuración de las indicaciones, velocidad, tipo de voz...

En resumidas cuentas, *Lazarillo* es una aplicación que, como otras, busca mejorar la calidad de vida de las personas con discapacidad visual indicándoles para ello qué les rodea y proporcionándoles una mayor independencia. Ésta, sin embargo, cubre únicamente los aspectos más básicos y elementales, sin reparar en otras posibles funcionalidades o indicaciones (obstáculos, peligros...) que la convierten en una aplicación incompleta.

La app es completamente gratuita y cuenta con versión para Android y iOS.

2.1.5. Wayfindr

Wayfindr (?) nació en 2015 en Londres, con la misión de capacitar a las personas con discapacidad visual para que viajen de manera independiente a través de una navegación de audio inclusiva y accesible. Con este fin, han desarrollado el primer estándar del mundo aprobado internacionalmente para la navegación de audio accesible y ya cuentan con las primeras demos de un sistema de navegación en red basado en audio que pretende dar soporte para que las personas con discapacidad visual puedan adentrarse por esos lugares

que están repletos de señales escritas, por los que las personas que ven pasan sin pensar pero que son precisamente los que más temen y evitan aquellos que tienen discapacidad visual (estaciones de metro, tren, aeropuertos, centros comerciales, hospitales, etc.).

Este proyecto de código abierto ha realizado ya numerosas pruebas en distintos escenarios, como por ejemplo en el metro de Londres donde el funcionamiento⁹ de la aplicación es tan práctico como sencillo: se basa en una serie de *beacons*, colocados en puntos estratégicos a lo largo de las distintas estaciones de metro, que emiten unas señales que son captadas por el móvil a su paso por un cierto radio de detección. Estas señales permiten ubicar al usuario y darle la siguiente indicación para el conseguimiento de su objetivo (coger un tren o salir de la estación). O más recientemente en el metro de Los Ángeles.

Los desarrolladores recomiendan el uso de auriculares de conducción ósea, de manera que puedan escuchar otros sonidos del exterior.

La idea de este proyecto supone un gran avance para las personas que tienen algún tipo de discapacidad visual ya que pretende empoderarlas para que no se sientan retenidas por su pérdida de visión ni tengan que vivir supeditadas a una persona vidente que las ayude, para lograr así, que se rompan de una vez por todas las barreras a las que están sometidas. Para la consecución de este fin, esta organización sin ánimo de lucro proporciona a los fabricantes de navegación digital y propietarios de espacios públicos las habilidades y técnicas para proporcionar a las personas con discapacidad visual, servicios de navegación digital consistentes y de alta calidad. Wayfindr utiliza varias tecnologías para rastrear la ubicación de una persona y activar el audio de las instrucciones en su teléfono móvil en el momento adecuado para llevarlos a su destino. De esta manera, busca permitir que las personas con discapacidad visual naveguen por el mundo utilizando las instrucciones de audio de sus teléfonos inteligentes.

2.1.6. Conclusiones

Tras este breve recorrido por algunas de las aplicaciones de navegación adaptadas para personas ciegas o con visibilidad reducida podemos decir que cada vez son más las opciones. Hemos visto desde aplicaciones de navegación por exteriores, como también por interiores, llegando hasta algunas tan específicas como *Wayfindr* que está dirigida al metro de Londres concretamente. Todas ellas se rigen por un patrón común: el de la simpleza, sin conllevar por ello una reducción de la funcionalidad. Pues estas aplicaciones nos permiten filtrar la información que se quiere recibir, guardar nuestros lugares más visitados, manejarlas mediante voz o con sacudidas del teléfono..., es decir, nos proporcionan un gran abanico de posibilidades que el usuario puede ejecutar de manera sencilla.

Por otro lado, si comparamos las apps, encontramos que aquellas de navegación por interiores están aún por desarrollar ya que el mapeo del interior de los edificios debe realizarse de manera particular e individual, convirtiéndose en una tarea mucho más tediosa que la que lleva a cabo el famoso coche de *Google Maps*. Además, el posicionamiento también es más complejo ya que no es posible utilizar el sistema GPS y hay que recurrir a la triangulación de señales WIFI o a las balizas bluetooth, teniendo que estudiar de nuevo cada caso concreto.

⁹<https://www.youtube.com/watch?v=mc3KmbfxuUQ>

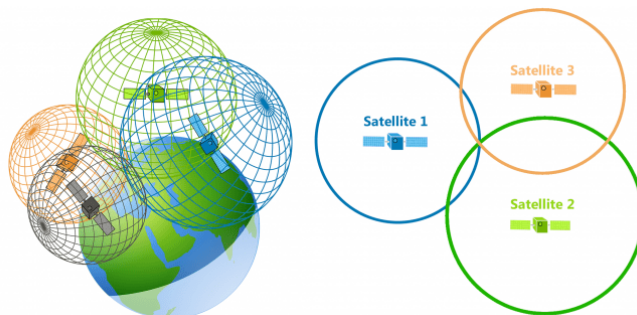


Figura 2.4: Método de triangulación GPS.

2.2. Sistemas de posicionamiento

Para la consecución de nuestro objetivo, el desarrollo de una aplicación de navegación por interiores, uno de los primeros problemas que se nos plantea es el del posicionamiento en un mapa ya que es de vital importancia poder determinar donde estamos para después indicar la ruta pertinente hacia el destino indicado. En esta sección haremos un pequeño estudio sobre las distintas tecnologías existentes que nos permiten solventar nuestro problema y determinar la posición exacta de un cierto dispositivo, y discutiremos su validez para su aplicación a este trabajo de fin de grado.

2.2.1. GPS

El Sistema de Posicionamiento Global (GPS) es un sistema de localización, diseñado por el Departamento de Defensa de los Estados Unidos con fines militares para proporcionar estimaciones precisas de posición, velocidad y tiempo. Este sistema se encuentra operativo desde enero de 1994 y se desarrolló a partir de los 24 satélites que componen la constelación NAVSTAR, cada uno de los cuales cuenta con una órbita de 26.560Km de radio y un periodo de 12h (?).

El método mediante el cual el GPS determina la altitud, longitud y latitud de cualquier objeto que se encuentre en la superficie terrestre se conoce como triangulación. Este requiere la distancia desde el dispositivo en cuestión (receptor) a tres satélites como mínimo cuya localización es conocida de antemano. Entonces, cuando el receptor detecta el primer satélite, se genera una esfera a su alrededor cuyo radio será la distancia desde el receptor hasta dicho satélite. De este modo, el receptor se encontrará en un punto de la superficie de esa esfera, aún por determinar. Repetimos el proceso con otro satélite. Al crearse esa segunda esfera, el dispositivo receptor se encontrará en alguno de los puntos de corte de ambas esferas, por lo que el resto de puntos se descartan. De nuevo, se utiliza un tercer satélite de modo que se crea una nueva esfera que cortará a las anteriores. De este modo, con el corte de las tres esferas, y teniendo en cuenta que el dispositivo se encuentra en la superficie terrestre, tendremos el punto concreto buscado. En caso de querer conocer la altitud, bastará con usar un cuarto satélite como referencia y repetir el proceso. En la Figura 2.4 vemos un esquema del proceso que acabamos de explicar empleando 3 y 4 satélites.

El problema del Sistema de posicionamiento Global es que pierde mucha precisión cuando nos encontramos bajo superficies como túneles, tejados, etc. ya que la señal se debilita enormemente y el dispositivo no es capaz de llevar a cabo la triangulación de manera exacta. Es por esto, que descartamos este sistema para nuestro propósito.

2.2.2. Wi-Fi

La técnica de posicionamiento mediante señales Wi-Fi fue una de las primeras que surgió para solventar el problema de la localización en interiores. Tal y como hemos mencionado en el apartado anterior, la señal GPS no es suficiente pues las barreras arquitectónicas como las paredes o tejados la debilitan enormemente. Por ello apareció esta técnica que se basa en leer con un dispositivo la intensidad de señal que recibe desde distintos puntos de acceso Wi-Fi del edificio. Una vez leída esta señal existen varios métodos para establecer la posición exacta del dispositivo, el más utilizado se basa en medir la intensidad con la que se recibe la señal en un cierto punto (*Received Signal Strength Indicator*, *RSSI*) desde varios puntos de acceso y, en base a eso, establecer la localización en la que se encuentra el dispositivo. Este método es fácil y barato de implementar, su mayor inconveniente es que no proporciona una buena exactitud (de 2 a 5 metros), ya que la señal es muy dependiente del ambiente: personas, paredes, etc.

Otro método que también se utiliza es el denominado *fingerprint*, este consiste en guardar la intensidad de la señal desde distintos puntos de acceso en una base de datos de acuerdo a coordenadas conocidas del cliente durante una fase sin conexión. Durante la fase de conexión, las medidas actuales RSSI de una posición que no se conoce se comparan con las guardadas en la base de datos y, aquellas que más se le parezcan, se devuelven como estimación de la posición del usuario. Esta técnica tiene mejores resultados en cuanto a la exactitud, cuyo error baja a menos de 3m (?).

Una de las grandes ventajas de la tecnología Wi-Fi es que prácticamente la totalidad de las casas, colegios y edificios en general, están equipados con una red Wi-Fi.

2.2.3. Balizas Bluetooth

Los *beacons* o balizas bluetooth son pequeños dispositivos que emiten señales de radio. Estas señales los identifican de manera única y pueden ser captadas por otros dispositivos receptores, estableciéndose así un canal de comunicación que permanece vivo siempre que los receptores permanezcan en un radio de alcance de entre 10 y 30 metros como máximo, según el dispositivo. Es importante remarcar que generalmente los *beacons* no aceptan conexiones de otros dispositivos, lo que significa que no pueden registrar qué aparatos están cerca. Por tanto, esta simplicidad conlleva la necesidad de una aplicación capaz de interpretar la señal de la baliza. Otra característica de los *beacons* es que son de bajo consumo, es decir, sus baterías tienen una duración muy prolongada (aproximadamente 2 años) con una simple pila de botón, y su coste es reducido.

Esta tecnología se hizo muy popular en 2013 cuando Apple introdujo el *iBeacon* estándar y comenzó a utilizarlos para la navegación, más concretamente, para el posicionamiento en interiores. En 2015 Google, que no quiso quedarse atrás, lanzó el protocolo Eddystone, un protocolo, que a diferencia del de Apple, es de código abierto y ofrece soporte oficial tanto para iOS como para Android. Otra ventaja que incluye la versión de Google es que proporciona dos APIs que facilitan mucho el manejo de los *beacons*, y que emite 4 paquetes distintos de información, en lugar de 1 como en el caso de los *iBeacons*. Estos paquetes son:

- **Eddystone-UID:** transmite un identificador de baliza único compuesto por 16 bytes, 10 de ellos referidos al espacio de nombres, que identifican a un grupo de *beacons*, y 6 que se refieren e identifican a la instancia particular dentro del grupo. Esta distinción entre espacio de nombres e instancia se pensó para optimizar el escaneo de *beacons*. Este paquete es idéntico al que ofrecen los *iBeacons*.

- **Eddystone-URL:** transmite una URL utilizando un formato de codificación.
- **Eddystone-TLM:** transmite información sobre la baliza. Como puede ser el nivel de la batería, los datos del sensor u otra información relevante para los administradores de balizas. Para poder usarse también como baliza necesita ir acompañado de otro tipo de marco (Eddystone-URL o Eddystone-UID).
- **Eddystone-EID:** emite un identificador encriptado que cambia periódicamente, de modo que su uso está restringido a aplicaciones y dispositivos autorizados.

Por todo esto, consideramos que el protocolo Eddystone es más ventajoso.

Por otro lado, de cara a establecer la posición exacta de un dispositivo receptor hay dos alternativas, triangular la posición a partir de la distancia obtenida por la señal de los 3 *beacons* más cercanos. El inconveniente de este método es que se requiere un número muy alto de balizas para poder cubrir por completo todo el espacio por lo que los costes de la instalación se elevan, además cualquier cambio en la estructura del edificio puede afectar a la posición de los *beacons* y por ende, al algoritmo de triangulación. El otro método consiste en establecer los *beacons* en puntos de decisión (*landmarks*) donde los usuarios esperan recibir instrucciones. Algunos de estos puntos son las puertas, escaleras, intersecciones, etc. El número de balizas necesario es mucho más reducido por lo que los costes derivados de la instalación también. No obstante, hay que tener en cuenta que es muy importante decidir con cuidado la disposición exacta de los *beacons*.

Como conclusión, aunque la alternativa tecnológica Wi-Fi es también adecuada para solventar el problema del posicionamiento en interiores y cuenta con ventajas como la de aprovechar la infraestructura del edificio sin necesitar ningún dispositivo extra, también conlleva inconvenientes como que la intensidad de las señales Wi-Fi dependen mucho del entorno y en ocasiones puede ser complicado diferenciar la posición entre plantas, si estas no se encuentran a suficiente distancia. Además, los *beacons* cuentan con ventajas como su bajo coste, flexibilidad: podemos colocarlos donde queramos (son pequeños y ligeros) mientras que los puntos de acceso Wi-Fi vienen predeterminados, y tienen una precisión de 1 a 3 metros, algo más alta que con la señal Wi-Fi¹⁰. Por ello, para este proyecto nos hemos decantado por las balizas bluetooth acompañadas del protocolo Eddystone y más concretamente por el método de los *landmarks* ya que debido a las limitaciones de presupuesto y a las características de nuestros usuarios lo consideramos más adecuado pues así podemos estudiar mejor los puntos de interés y asegurarnos una buena señal y más precisión en ciertos puntos.

¹⁰<https://www.infsoft.com/technology/positioning-technologies/bluetooth-low-energy-beacons>

Capítulo 3

Descripción del Trabajo

3.1. Reunión en el Centro de Tiflotecnología e Innovación de la ONCE

La idea de este TFG surge de la necesidad de resolver problemas reales para gente real, concretamente para personas con discapacidad visual. De esta manera, empezamos nuestro camino por lo más importante: conocer las necesidades de los usuarios finales. Con este fin y gracias a la oportunidad que nos ha brindado la Universidad Complutense de Madrid con la profesora María Guijarro al frente, hemos podido reunirnos y entrevistar a personas especializadas en el campo de las tecnologías que sufren discapacidad visual.

En este documento recogemos las notas que tomamos durante la reunión el pasado 11 de octubre de 2019 en el CTI (Centro de Tiflotecnología e Innovación) de la ONCE.

3.1.1. Introducción

La entrevista comenzó con una breve explicación, de la mano de José María Ortiz, director del Departamento de Consultoría e Innovación, sobre las principales tareas que se llevan a cabo en el centro, entre las cuales destacan:

- Ayudar a una persona con discapacidad visual en su **adaptación** al trabajo y a la vida cotidiana, proporcionándole para ello el material necesario (teclados, líneas de braille, bastones, etc.).
- Responder a **consultas** sobre el funcionamiento de dispositivos.

Luego, nos comentó los departamentos en los que se estructura el centro para que pudiésemos hacernos una idea más global de todo lo que abarca. Éstos son:

- **El departamento de Consultoría e Innovación:** donde actualmente están desarrollando el programa EDICO en colaboración con la UCM, que tiene como objetivo hacer las matemáticas accesibles mediante un editor de texto. De manera paralela se encargan del desarrollo de aplicaciones de muy diversa índole, véase apps para la biblioteca de la ONCE, de películas audio-descritas, etc.
- **Departamento de Evaluaciones y Auditoría:** donde se encargan de evaluar los productos que se van a sacar al mercado.

- **Departamento de diseño y producción:** donde se encargan de, tal y como indica su nombre, diseñar y producir elementos de adaptabilidad, como pueden ser unas plantillas con relieve de policarbonato para las vitrocerámicas. Recordemos que estas, aunque no presentan dificultad alguna para los usuarios videntes, son tediosas para aquellos que cuentan con discapacidad visual ya que la pantalla táctil no tienen ningún tipo de relieve que pueda servirles como referencia y guiarles en su uso.
- **Departamento de Asesoría en tecnología:** especializado en tecnologías accesibles.

Una vez concluida esta sección en la que nos contextualizaron, abrieron paso a la ronda de preguntas en la que pudimos acercarnos a ellos, conociendo sus problemas y necesidades en el día a día.

3.1.2. Entrevista

Durante esta parte, nos dirigimos especialmente a Mónica y José Luis Llorente, ambos ingenieros del CTI, para que, con su experiencia y conocimientos, nos explicaran lo máximo posible sobre tecnologías accesibles y nos dieran su punto de vista en las ideas que proponíamos. Por otro lado, Mónica no solo era experta en la materia sino que además era invidente, por lo que nos pudo contar su perspectiva y necesidades como usuaria.

Las preguntas avanzaron desde temas generales para conocer cómo una persona invidente se desenvuelve con la tecnología, sus gustos y qué sensaciones le despierta, hasta temas concretos dirigidos a conocer los problemas que plantea la navegación por espacios interiores.

- **¿Cómo utiliza una persona con discapacidad visual un dispositivo móvil?**

Para responder a esta pregunta, Mónica nos hace una demostración en directo, para ello emplea un móvil Xiaomi con sistema operativo Android.

Mónica nos cuenta que para la navegación por su dispositivo utiliza un lector de pantalla, es decir, un software que facilita el uso del sistema operativo. Éste sirve como guía para las personas que, como ella, tienen discapacidad visual, ya que “lee y explica” mediante voz lo que se ve en la pantalla. Los lectores de pantalla vienen siempre incluidos en el dispositivo y se pueden encontrar en la sección de Accesibilidad, en Ajustes. En el caso de Android, este software se llama *Talkback* y es configurable. Por ejemplo, dice Mónica, se podría usar mediante la línea de braille en vez de la reproducción por voz.

Luego vemos como se desplaza por las aplicaciones utilizando *flicks*, movimientos secos en los que desliza el dedo hacia uno de los lados de la pantalla (izquierda o derecha, según interese). Del mismo modo, para la navegación por la web o dentro de alguna aplicación utiliza estos movimientos hacia arriba y hacia abajo. Por último, nos muestra cómo accede a un elemento mediante doble click.

También nos habla de la posibilidad de la navegación libre, eso sí, solo cuando ya te has familiarizado con el dispositivo lo suficiente como para saber dónde tienes determinadas aplicaciones.

Lo más cansado, según Mónica, es tener que hacer un barrido por toda la pantalla hasta encontrar lo que quieres, en vez de poder ir directamente. Para agilizar un poco este proceso, Mónica, por ejemplo, agrupa las aplicaciones por carpetas, de modo que el barrido es más sencillo que si la pantalla estuviese repleta.

Para las personas con baja visión también existe la posibilidad de hacer más grandes los iconos y ajustar los colores.

- **Hemos leído que normalmente las aplicaciones se desarrollan para dispositivos iOS, ¿por qué es mejor?**

“Si que es cierto que solía ser así ya que iOS le llevaba la delantera a Android en cuanto a accesibilidad, pero cada vez se usa más Android pues las diferencias están completamente recortadas, están muy igualados y los precios son mucho más asequibles. Yo misma, antes tenía un iPhone y ahora me he pasado a Android y no hay nada que eche en falta.”, responde Mónica.

- **¿Cómo afronta una persona ciega su desplazamiento y orientación por interiores cuando pisa por primera vez dicho espacio u edificio?**

Ante esta pregunta Mónica resopla y nos contesta: *“Buufff..., ¿te vale?”*

Nos puso como ejemplo la llegada a un hospital: *“cuando entras necesitas saber, al menos, dónde está la recepción para pedir ayuda pero los carteles informativos están fuera de mi alcance, entonces entro por la puerta y pienso ¿y ahora qué?. ¿Dónde está el mostrador de recepción? No es tan fácil como echar un primer vistazo, necesitas ayuda mediante voz, algo que te describa el espacio y te vaya diciendo que hay a derecha e izquierda y a cuantos metros.”*

Nos contó que en cuanto a la descripción/guía por espacios interiores ahora mismo no hay disponible ninguna aplicación. Por ello, una vez superada la primera barrera de ubicar y localizar un cierto destino, la única opción que les queda es la de memorizar el camino. Mónica destacó que era increíble la cantidad de rutas que tiene en la cabeza.

Por todo esto, se comentó que una aplicación sería de gran ayuda para ellos, de manera que pudiesen tener una idea del edificio incluso antes de llegar a él para moverse con más seguridad. Una app que no solo les guiase a un punto concreto, sino que además describiese el edificio, indicándoles qué posibilidades les ofrece. También se mencionaron otras propuestas e ideas como tener previamente el plano del edificio para poder ir moviéndose con el dedo sobre él y que a su paso te vaya indicando las distintas salas que aparecen, o la impresión de un mapa 3D que disponga de un código QR o algo similar que fuese capturado mejor por Bluetooth que por foto que tras leerlo cargase el plano del edificio y pudiese proporcionar tanto información sobre el espacio en sí mismo (número de plantas, qué hay en cada una...) como información más precisa como puede ser averías, horarios, disponibilidad de salas, etc.

Obviamente de la mano de estas ideas surgían problemas y opiniones en contra: ¿Dónde estaría dicho mapa?, ¿Cómo encontrarlo?, ¿Todos los edificios estarán de acuerdo en facilitar los planos o puede que por motivos de seguridad no sea una idea factible?, ¿es posible llegar a un standard para que se pueda usar el mismo sistema en cualquier edificio?

- **¿Hay algún tipo de señales que os sirvan como referencia a la hora de desplazarnos por un edificio?**

“Hay señales de encaminamiento, que te indican dónde están las escaleras, ascensores, zonas de cruce, etc.”, contesta.

- **¿Cuántos edificios cuentan con estas señales?**

“La verdad que cada vez son más frecuentes y hoy en día se encuentran en casi todos los edificios, especialmente en los nuevos.”, responde.

- **¿Cómo de factible es ir con el dispositivo móvil en la mano, para realizar una foto o cualquier cosa similar?**

“Puedo hacer una foto en un momento puntual, en eso no hay problema alguno pero no es cómodo ir con el móvil en la mano constantemente porque además de que es aparatoso porque ya llevo en la mano el bastón, perro guía, etc. No es práctico, no sería la primera vez que roban un móvil a una persona invidente, es una realidad.”, contesta Mónica. “Particularmente, con respecto a la foto el problema principal sería saber a dónde enfocar”, añade.

3.1.3. Conclusiones

Tras el debate, algunas de las conclusiones que sacamos de la visita al CTI son:

- La implementación de una aplicación como la nuestra es muy útil y necesaria.
- Las modalidades más empleadas para interactuar con el móvil cuando tienes algún tipo de rastro visual son: flicks, sacudidas, mediante vibración, arrastrando o pulsando la pantalla con un dedo, dos,...
- No resulta cómodo ir barriando el espacio con la cámara del móvil.
- El uso de dispositivos adicionales como una micro cámara, en principio, no sería un problema, siempre y cuando no la tengan que llevar de manera continuada en la mano.
- En caso de auriculares, se recomienda utilizar auriculares óseos de modo que dejen el canal auditivo libre para captar otros estímulos.
- El objetivo es que el grueso de las aplicaciones sean lo más inclusivas posibles, es decir, que su uso sea apto tanto para personas videntes como invidentes.
- El feedback de la aplicación no debe saturar pero sí se aconseja que sea constante para que no se malinterprete que la aplicación ha dejado de funcionar.

3.2. Estudio de la precisión de los beacons

Antes de embarcarnos en el diseño de la aplicación vamos a hacer un estudio más en profundidad de los *beacons* con el fin de comprender por completo su funcionamiento y cómo se comportan a la hora de medir distancias, es decir, cuánto rango tiene la señal *bluetooth*, con ayuda de qué función podemos recibir esta señal e interpretarla para determinar la distancia, qué margen de error presenta, en qué lugares es más aconsejable establecer las balizas para recibir mejor la señal y por tanto reducir el error, etc.

Los *beacons* utilizados para este proyecto cuentan con una SDK (Kit de Desarrollo Software) propia de su marca (*Kontakt*) en la que incluyen funciones ya implementadas que nos permiten conocer qué *beacons* están en nuestro rango en un momento determinado y a cuántos metros están, y se va actualizando cada cierto tiempo. También incluye un sistema de categorías en función de cómo de cerca o lejos esté un cierto dispositivo. Las categorías son las siguientes:

- *IMMEDIATE*: Si el dispositivo se encuentra a menos de 0,5 metros.
- *NEAR*: Si el dispositivo se encuentra entre 0,5 y 3 metros.
- *FAR*: Si el dispositivo se encuentra a más de 3 metros.
- *UNKNOWN*: Si se ha perdido la señal del dispositivo.

Gracias a estas funciones hemos desarrollado una serie de aplicaciones que nos han servido para tener una primera toma de contacto con el entorno de programación que hemos escogido, Android Studio, y para contrastar de manera visual el error existente entre la distancia real a la que se encuentran las balizas y la distancia leída mediante la interpretación de las señales bluetooth. Las aplicaciones son las siguientes:

3.2.1. Aplicación *miniapp*

Esta aplicación constituye la primera toma de contacto con los *beacons* y con el código de la SDK. Presenta una interfaz sencilla en la que aparece una tabla con las distintas categorías de proximidad para 3 balizas y, debajo, el hueco para indicar el identificador correspondiente. En el cuadro inferior aparecen dos botones muy intuitivos con los que el usuario puede interactuar con la app: Stop Scanning y Start Scanning. De esta manera, una vez que empieza el escaneo y las balizas se encuentran en el radio de detección, se incluyen los identificadores en cuestión y se colorea en verde la categoría de proximidad estimada en cada caso según corresponda. La lectura de los *beacons* se actualiza cada 2 segundos, tiempo establecido de antemano. En la Figura 3.1 podemos ver la interfaz descrita.

La idea subyacente para desarrollar esta aplicación es la de establecer el grado de confianza que podemos tener en las categorías ofrecidas por la SDK de *Kontakt* y así, ver si podemos reutilizarlas para dar ciertas instrucciones al usuario cuando este se encuentre en un cierto rango.

3.2.2. Aplicación *cuadrantes v1*

Tras la aplicación *miniapp*, hemos desarrollado una segunda, llamada *cuadrantes v1*, capaz de captar los *beacons* al paso e indicar la categoría de proximidad y los metros exactos a los que lee las balizas. Esta información se actualiza cada 2 segundos (tiempo configurable). La Figura 3.2 muestra la interfaz principal de la aplicación.

La idea por la cual hemos creado la app *cuadrantes v1* es la de obtener más información sobre el error cometido a la hora de determinar la distancia en metros a la que se encuentran los *beacons* y ver qué factores le afectan para poder empezar a pensar los puntos claves en los que colocar las balizas y abordar, tras ello, el mapeo de la edificio. De ahí el nombre de la aplicación ya que, como veremos, los cuadrantes serán una pieza fundamental en el mapeo.

3.2.3. Resultados

A continuación presentaremos algunos de los resultados obtenidos en las distintas mediciones realizadas. Para ello, hemos creado unas gráficas que muestran cómo se comportan los *beacons* si comparamos la medida real con la estimada por la aplicación.

En la Figura 3.3 vemos las lecturas que nos ha proporcionado la aplicación *cuadrantes v1* del *beacon* con identificador CPne. En este caso, hemos realizado 4 estudios independientes marcados con distintos colores: azul -*beacon* a 2m-, naranja -*beacon* a 2,5m

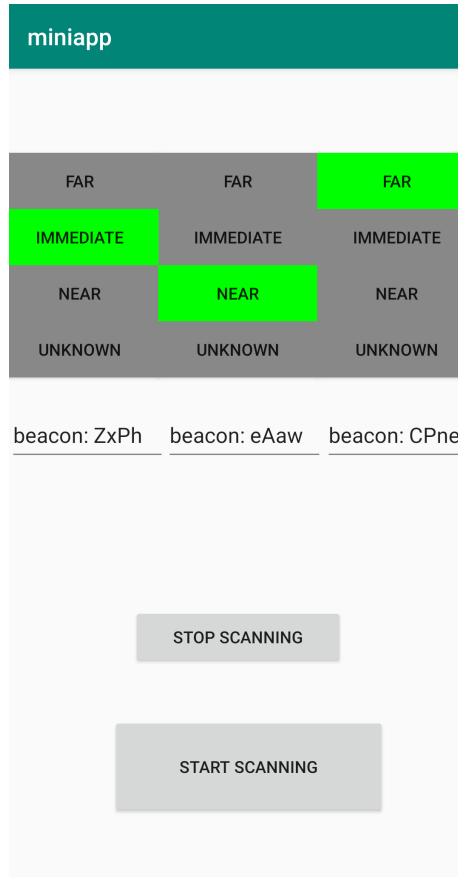


Figura 3.1: Interfaz de la aplicación miniapp.

aproximadamente-, gris -*beacon* a 5 metros- y amarillo -*beacon* a 64cm-. De esta gráfica concluimos que cuanto mayor es la distancia a la que se encuentra la baliza, véase la función grisácea (5m), más fluctúa la medida estimada convirtiéndose en un dato poco fiable. Correlativamente, a menor distancia menor error. Podemos ver que la función azul (2m) es bastante fiel a la realidad, fluctúa en un intervalo de un metro a lo largo de toda la medición. Por último, el resultado de la gráfica amarilla (64cm) es bastante preciso, presenta un error despreciable durante toda la medición.

En el caso de la Figura 3.4, en la que se estima la distancia para el *beacon* eAaw, el comportamiento es similar. En líneas generales, el valor medio estimado se corresponde con la distancia real a la que se encuentran las balizas. Sin embargo, en esta medición hemos contemplado una novedad que se ha repetido en numerosas ocasiones convirtiéndose en un patrón de comportamiento y es que durante los primeros segundos en los que arranca la aplicación, las estimaciones son menos precisas, dando lugar a picos importantes que más tarde se suavizan.

La Figura 3.5 recoge cuatro mediciones para el *beacon* 8v2c, tres de ellas tomadas para una misma distancia (4m) y una a 64 cm. Observamos de nuevo que el error es despreciable cuando la baliza se encuentra muy próxima. Por el contrario, los valores que recogen las gráficas en las que el dispositivo se encuentra a 4m están muy por debajo de lo esperado, solo una de ellas alcanza el valor real. Este factor común revela una alteración considerable de la intensidad de la señal provocada por algún factor del entorno que la debilita. Es por esto que la última gráfica plasmada en la Figura 3.6, hace referencia a una situación particular. Se trata de dos *beacons* diferentes situados a la misma distancia uno encima del

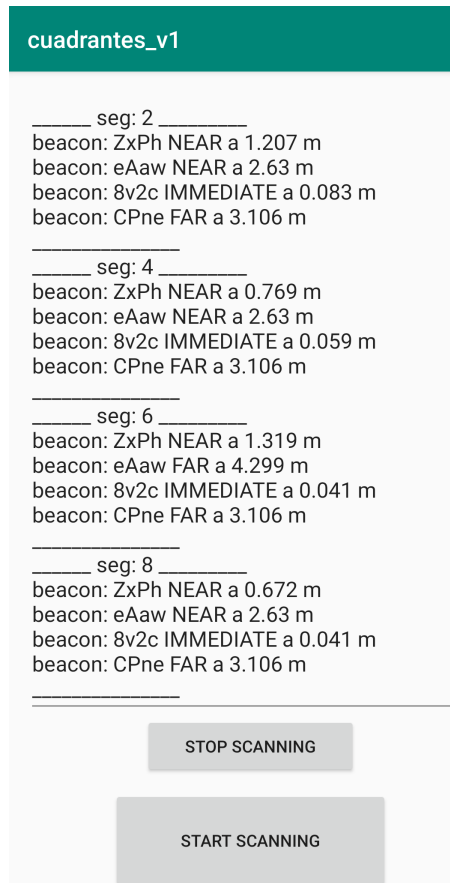


Figura 3.2: Interfaz de la aplicación cuadrantes v1.

otro. El objetivo de este estudio era ver si uno de los factores que podía alterar la intensidad de señal era la presencia de otras balizas, y efectivamente como podemos observar, la intensidad de ambas señales es bastante más baja de lo esperado, prácticamente en ningún momento alcanzan el valor real. En este caso particular, el error es tolerable ya que los *beacons* están situados a muy poca distancia pero sí nos advierte de que la señal bluetooth es sensible a interferencias provocadas por otros *beacons* del entorno. En el siguiente apartado estudiaremos qué otros posibles factores pueden ser los desencadenantes de alteraciones en la señal.

Cabe destacar que aunque en determinados casos los datos son muy fluctuantes y poco precisos, las categorías de proximidad, con los rangos establecidos, se mantienen en su mayoría fieles a la realidad.

3.2.4. Interferencias en la señal bluetooth

Los *beacons* son dispositivos que se pueden colocar tanto en interiores como en exteriores pero hay que tener en cuenta qué lugares son más aptos para su establecimiento de manera que la señal no sufra interferencias, algunas recomendaciones son:

- Lugares altos (alrededor de 2,5m) para evitar las interferencias provocadas por los cuerpos de las personas ya que estos absorben parte de la señal.
- Lugares alejados a un metro aproximadamente de elementos que pueden alterar la

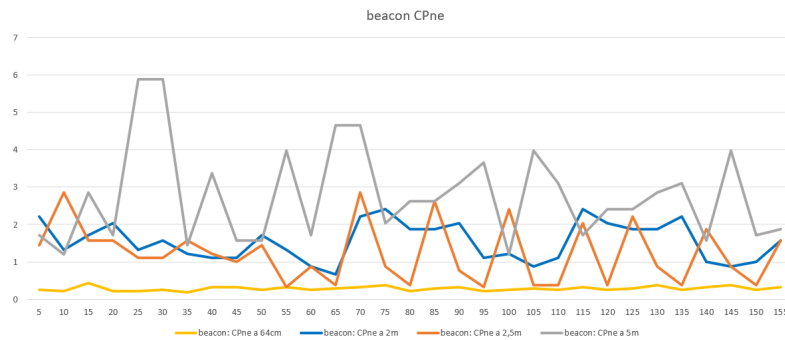


Figura 3.3: Gráfico con las distancias medidas al beacon CPne.

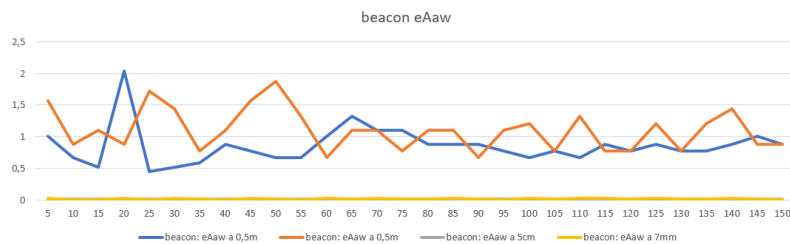


Figura 3.4: Gráfico con las distancias medidas al beacon eAaw.

intensidad de la señal como elementos metálicos, conductos de electricidad, iluminación, otras balizas, etc.

- En pasillos de menos de 4m de ancho, colocar la baliza en el medio para que cubra el espacio por igual. Si por el contrario, el ancho es mayor se deberán usar varias balizas.
- Colocar las balizas aproximadamente 1m antes de los lugares de interés (*landmarks*) en los que las personas necesitarán la siguiente instrucción. Ejemplo: puertas, ascensores, escaleras, esquinas, etc.
- Considerar la orientación adecuada de la antena direccional del *beacon*, esta depende por completo del fabricante. En ocasiones la baliza no emite una señal simétrica por completo sino que emite una señal en forma elíptica.

Otros factores difíciles de controlar que también afectan a la señal son las condiciones meteorológicas, la señal Wi-Fi, las señales bluetooth de otros dispositivos...

3.2.5. Mediciones en puntos clave de la facultad

Ahora que hemos superado la primera barrera tecnológica y tenemos una idea más clara del funcionamiento y el comportamiento de los *beacons*, nos disponemos a dar el siguiente paso hacia la resolución del problema del posicionamiento. Para ello, estudiaremos los planos de la Facultad de Informática, determinaremos cuales podrían ser, a priori, los puntos de decisión y haremos mediciones para comprobar si las señales de los *beacons* se reciben correctamente y no sufren interferencias en dichos puntos.

Los puntos de decisión son aquellos lugares de considerados de interés para el usuario o bien aquellos puntos en los que se requerirá la siguiente instrucción a seguir en la ruta. Los puntos de decisión que hemos considerado son: los destinos (aulas, cafetería, biblioteca,

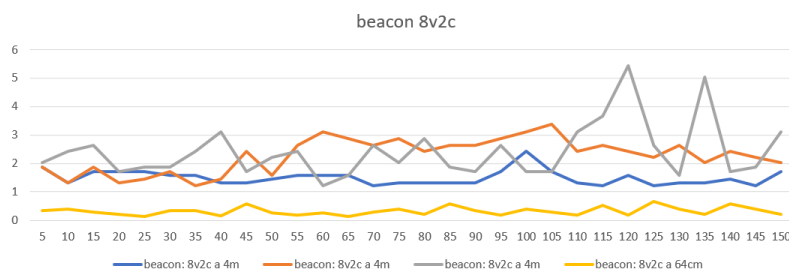


Figura 3.5: Gráfico con las distancias medidas al beacon 8v2c.

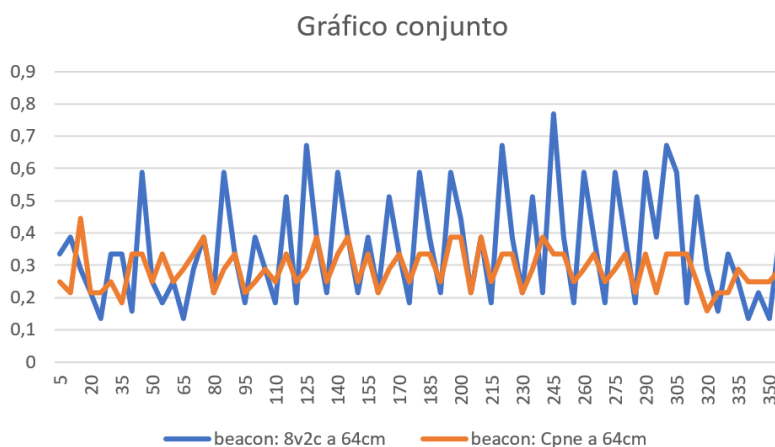


Figura 3.6: Gráfico de los beacons 8v2c y CPne superpuestos.

conserjería, secretaría, salón de actos, sala de juntas, sala de grados, etc.), las intersecciones (esquinas), los ascensores y escaleras, y las puertas de entrada al edificio.

En las Figuras 3.7 y 3.8 mostramos visualmente algunos de los puntos en los que hemos colocado los *beacons* (círculos rojos) y los puntos desde los cuales hemos llevado a cabo las mediciones correspondientes (cruces verdes). El objetivo de este estudio es conocer la ubicación óptima de los *beacons* para que a nuestro paso cerca de ellos, la distancia registrada sea lo más precisa posible, especialmente en los lugares más complicados como las intersecciones o las zonas en las que se acumulan varios puntos de interés (como el *hall* de entrada). En el ANEXO se pueden ver los resultados de estas mediciones (HABRÁ QUE AÑADIR MÁS MEDICIONES-COMPLETAR). A continuación exponemos algunas de las conclusiones recogidas:

- Los *beacons* no deben situarse demasiado cerca ya que las señales interfieren entre sí y alteran las distancias no pudiendo distinguir cuál es el *beacon* más cercano. Por este motivo, hemos acordado no poner, por el momento, balizas en los baños ya que, tanto en la primera planta como en la planta baja, se encuentran en zonas repletas de puntos de interés.
- En lugares diáfanos, como el *hall*, la señal de los *beacons* fluye con mayor libertad ya que no se encuentra con obstáculos. Es por esto, que deben situarse a mayor distancia entre sí. Uno de los problemas que se ha derivado de este hecho es cómo cubrir dicho espacio. Inicialmente colocamos una baliza en conserjería, otra en la intersección superior con el pasillo principal, otra en la intersección inferior con el pasillo que conduce a cafetería y otra enfrente para indicar la entrada al salón de

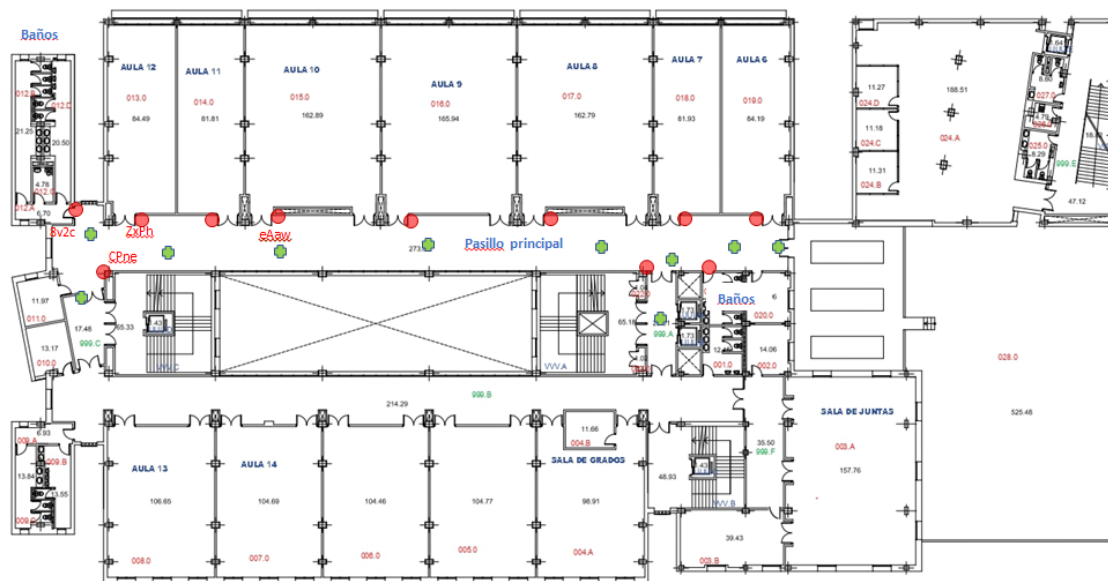


Figura 3.8: Mapa de la primera planta de la Facultad de Informática con la ubicación de los beacons (rojo) y los puntos de medición (verde).

3.3. Mapeo del edificio

Para el mapeo de la Facultad de Informática nos hemos apoyado en el proyecto *Generador interactivo de instrucciones de guía sobre plataformas móviles (?)*. De este, hemos reutilizado el sistema de estructuración basado en cuadrantes que contienen un identificador único (para diferenciar más fácilmente en qué planta estamos), y hemos prescindido de las posiciones sureste y noroeste ya que al emplear un sistema de posicionamiento basado en puntos de decisión no vamos a triangular la posición exacta del usuario y, por tanto, no necesitamos conocer las coordenadas (x,y) concretas de los cuadrantes. En su lugar hemos incluido el identificador (único) del *beacon* asociado a dicho cuadrante. Por este mismo motivo, aquellos cuadrantes que no tienen asociada una baliza bluetooth carecen de interés ya que el programa te sitúa en el cuadrante con el *beacon* más cercano desde tu posición. Es por esto que hemos aunado los cuadrantes formando unos nuevos más grandes. En la Figura 3.10 mostramos la planta 1 con los cuadrantes originales (en rojo), su identificador (en verde) y la posición final de los *beacons* (en amarillo) según lo que acordamos en la Sección 3.2.5, mientras que en la Figura 3.11 encontramos la versión final, en la que solo permanecen los cuadrantes con baliza. Además podemos observar que hemos suprimido los cuadrantes de las aulas, sala de juntas, etc. esto se debe a que nuestra aplicación no te guía dentro de estas estancias sino que funciona como guía de puerta a puerta.

Para poder establecer una red de cuadrantes que nos permita generar una ruta válida pasando a través de ellos, hemos mantenido la información de conectados con cada cuadrante por los puntos cardinales (norte, sur, este y oeste).

Esta estructura, junto con la información sobre la planta (coordenada z) en la que te encuentras, queda guardada en unos ficheros .xml almacenados en el servidor en una carpeta llamada xml_modif que se encuentra en el mismo directorio de la aplicación compilada del servidor .jar. Estos ficheros son:

- edificio.xml: Archivo principal, recuperado del proyecto mencionado, que indexa las

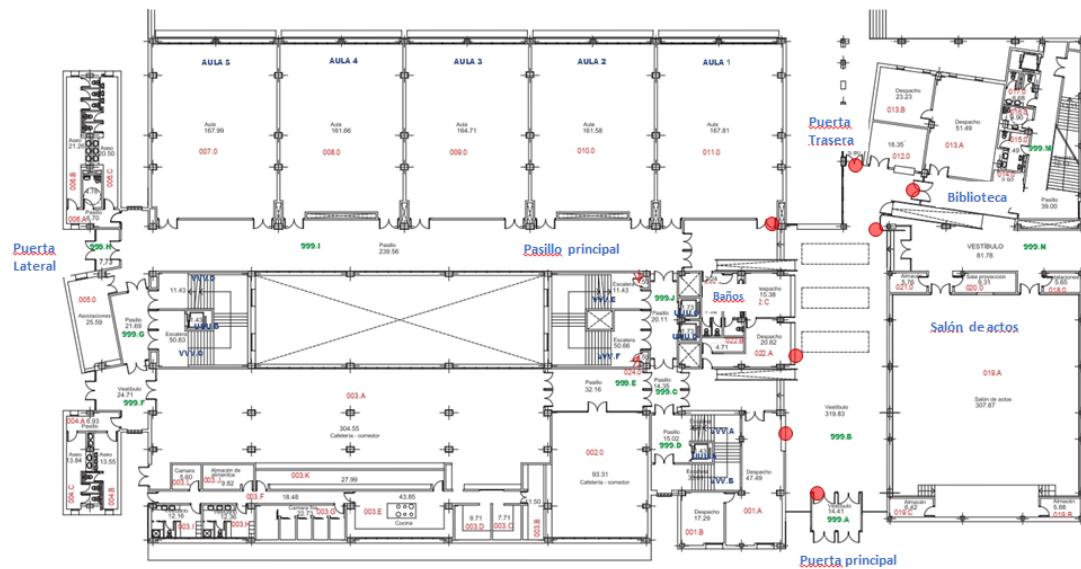


Figura 3.9: Mapa de la planta baja de la Facultad de Informática con la ubicación definitiva de los beacons (rojo) en el hall.

diferentes plantas del edificio.

- plantaX.xml: Archivo que representa una planta del edificio.

Otras novedades que hemos incluido en los archivos plantaX.xml son: añadir los metros que ocupan los cuadrantes para poder dar una instrucción más precisa, y añadir un apartado *info* para poder informar al usuario, si lo desea, de qué hay a su paso por la ruta, esta idea la tuvimos tras investigar distintas aplicaciones de guía para personas con discapacidad visual y tras la reunión con la ONCE, en la que nos acercamos mucho a las necesidades de nuestros usuarios. Por otro lado, como podemos ver en la Figura 3.12 hemos incluido el mapeo de la planta baja y hemos conectado las distintas plantas para incluir rutas de una a otra (novedad con respecto a los trabajos predecesores).

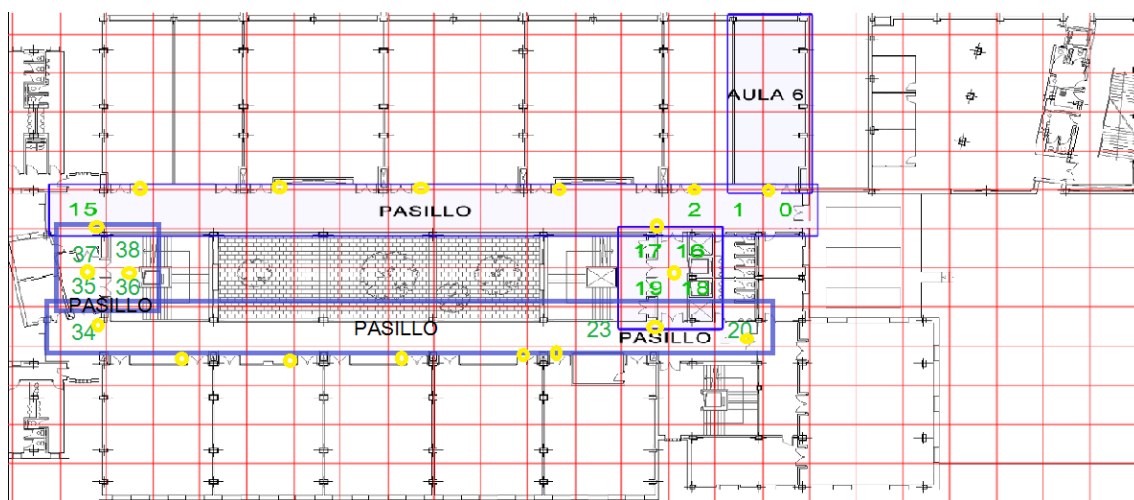


Figura 3.10: Primera versión del mapeo de la primera planta de la Facultad de Informática.

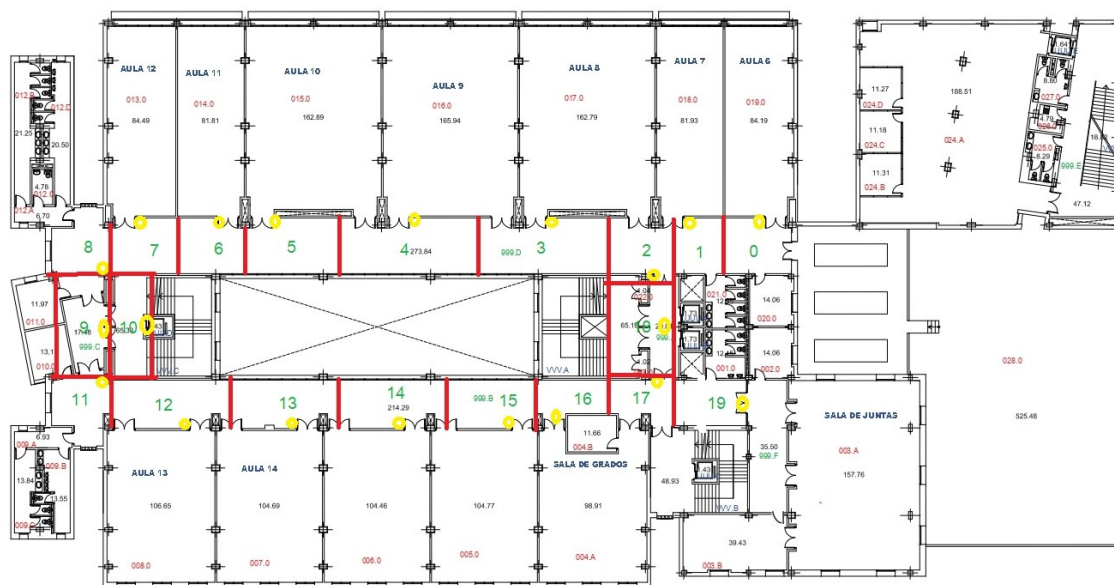


Figura 3.11: Versión final del mapeo de la primera planta de la Facultad de Informática.

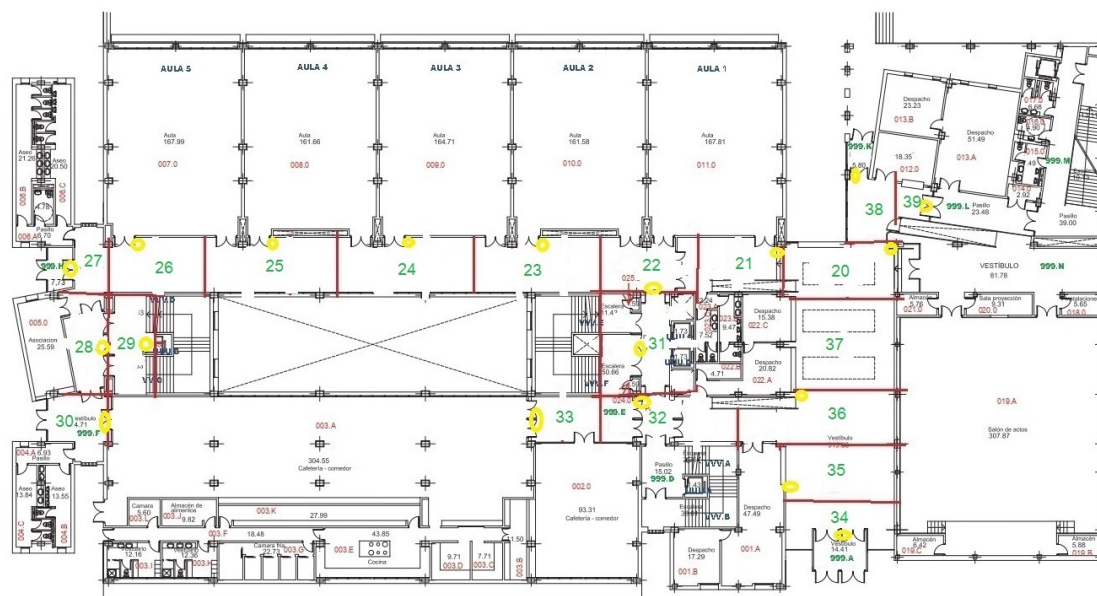


Figura 3.12: Mapeo de la planta baja de la Facultad de Informática.

Capítulo 4

Diseño e implementación

Hemos planteado la aplicación como un modelo cliente-servidor, en la que el servidor se encargará de ejecutar el programa que lleve a cabo todos los cálculos y se los proporcionará al cliente (dispositivo móvil) cuando este lo solicite. De esta manera, nuestro programa estará mejor organizado, será mas rápido y evitaremos que nuestro dispositivo móvil se quede sin batería en el cálculo de datos. A continuación detallaremos el diseño y la implementación tanto del cliente como del servidor.

4.1. Servidor

El servidor es una parte indispensable del proyecto ya que se encarga de realizar los cálculos más pesados para no sobrecargar al dispositivo móvil. Las principales tareas de las que se encarga son:

- Permanecer a la escucha de cualquier cliente que solicite conexión.
- Solventar el posicionamiento del cliente conectado.
- Generar la mejor ruta (más corta y mejor adaptada) desde la posición actual hasta el destino indicado.
- Enviar la siguiente instrucción al cliente.

La aplicación servidor se divide en código java y los archivos .xml en los que se encuentra la información del edificio. Buena parte del código que los conforma ha sido reutilizado de trabajos anteriores, en concreto del proyecto *Generador interactivo de instrucciones de guía sobre plataformas móviles (?)* y del proyecto *Sistema de guía por voz en interiores (?)*. Sin embargo se han introducido cambios notorios para el desarrollo de esta aplicación que comentaremos a continuación.

4.1.1. Archivos XML

-completar

4.1.2. Código java

Para entender mejor la estructura de nuestro servidor, veamos las principales clases de las que se compone y su función. En la 4.1 se puede ver el diagrama con las dependencias de cada una de ellas:

- *MainClienteAndroid*: como su propio nombre indica, es el *Main* del servidor. Se hizo otra versión para probar el servidor sin necesidad de acceder a él mediante la aplicación móvil, es por ello que lleva el identificador de Cliente Android en el nombre. Esta clase se encarga de la conexión directa con el cliente, abre un *ServerSocket* que queda a la espera de clientes y, una vez llegan, los atiende. Es decir, lee sus posiciones de origen y destino al que se quieren dirigir y devuelve la siguiente instrucción que deben seguir. Para ello se apoya en distintas clases que detallamos a continuación.
- *Edificio*: esta clase es muy importante, pues guarda la información relativa al mapeo: los cuadrantes y las estancias. En nuestro caso, estas últimas se corresponden con las plantas del edificio, para la Facultad de Informática tenemos dos estancias, la planta baja y la primera planta. Para llevar a cabo su objetivo, *Edificio* lee el archivo *edificio.xml* que tiene como elementos las distintas plantas y de estas, carga la lista de cuadrantes y estancias con ayuda de la clase *CargaXML*.
- *Estancia* y *Cuadrante*: sus propios nombres son muy descriptivos. La clase *Estancia* guarda la información de los cuadrantes que hay en cada planta y cuenta con funciones que permiten saber si un cuadrante está en una determinada estancia o añadir un cuadrante a una estancia. En el caso de la clase *Cuadrante*, contamos con una serie de estructuras con las que almacenamos toda la información referente a estos, véase su identificador, el *beacon* asociado, la información relevante de cada cuadrante, los metros que ocupa, la planta en la que está situado y, una de las cosas más importantes, con quién está conectado.
- *ListaCuadrantes*: esta clase constituye una parte muy importante ya que es la encargada de establecer la relación entre el identificador del cuadrante y el identificador del *beacon* asociado, y de formar la matriz de adyacencia a partir de la información de conexión de cada cuadrante. Es aquí donde introducimos los pesos que tiene cada conexión en función de la dificultad que supone la misma para una persona con discapacidad visual. Veremos esto con más detalle en la Sección 4.1.3. Esta clase también se encargará de obtener la lista de cuadrantes que conformarán la ruta de origen a destino mediante el algoritmo de *Dijkstra*.
- *Persona*: esta clase almacena la información referente a la ruta (lista de cuadrantes) que tiene que seguir el cliente para alcanzar el destino desde su posición origen. Para ello, llama a la función *caminoConDijkstra* de la clase *ListaCuadrantes*.
- *GenerarRuta*: Es probablemente la clase más importante del servidor. Su objetivo es devolver la siguiente instrucción que debe seguir el cliente para completar la ruta. La función encargada que contiene toda la lógica relativa a este propósito es *generar*. Esta función es una de las que más modificaciones ha sufrido con respecto a los trabajos predecesores, pues no solo la hemos adaptado para personas con discapacidad visual sino que también hemos incluido mucha complejidad al añadir cambios de planta y demás casuística que no estaba previamente incluida. La principal adaptación que hemos añadido es la de proporcionar instrucciones mucho más precisas e informadas. Por ejemplo: en lugar de finalizar la ruta con el mensaje “Ha llegado a su destino.”, ahora se indica dónde está el destino dependiendo de la ruta que se haya seguido y la orientación del usuario. Podría ser: “Su destino está a la derecha” o a la izquierda, etc. Otra novedad es que la ruta incluye la posibilidad de informar anticipadamente sobre lo que el usuario se va a encontrar a su paso, como pueden ser pequeños escalones, baños, la cafetería, el despacho de Delegación de Alumnos, etc. Esto es

de adyacencia y de esta manera, el cálculo de la ruta más corta entre dos cuadrantes se reduce al algoritmo de *Dijkstra*.

Sin embargo, no debemos olvidar que nuestra aplicación tiene un usuario final muy concreto: personas con discapacidad visual. Es por ello que la ruta debe ser lo más sencilla posible, libre de obstáculos y otros elementos que puedan entorpecerles, por lo que en ocasiones la ruta óptima no coincide con la más corta. Tras hacer numerosas pruebas de distintas rutas, hemos detectado un ejemplo de ello: al comenzar en la puerta principal de la facultad y terminar en la puerta trasera de la cafetería, el algoritmo de *Dijkstra* sugiere que el camino óptimo es acortar por detrás de conserjería (cuadrante 32) ya que es la ruta más corta pero no la óptima para una persona con discapacidad visual pues ese pasillo es más estrecho, la gente se suele aglomerar (están los ascensores, la gente continúa su camino a la cafetería, etc.) y cuenta con más giros. Por ello, hemos considerado que es más conveniente continuar la ruta por delante de Conserjería y luego girar a la izquierda. Para lograrlo hemos modificado la matriz de adyacencia, añadiendo más peso en aquellas conexiones que hemos considerado menos adecuadas. En este caso particular, a las conexiones entre los cuadrantes 31-32 y 32-22. De esta manera hacemos que el paso por el pasillo de los ascensores se limite únicamente al caso en el es estrictamente necesario pasar por allí, es decir, cuando se va a hacer un cambio de planta.

4.1.4. Detalles técnicos del posicionamiento

A diferencia de todos los proyectos previos de guía por la Facultad de Informática (comenzando por AVANTI, PONER REFERENCIA), el nuestro introduce una novedad notoria: los *beacons*. Esto supone un gran cambio en el sistema de posicionamiento ya que eliminamos por completo la triangulación y nuestro servidor recibe información solo del *beacon* más cercano al cliente. Es entonces cuando el servidor, a través de la clase *ListaCuadrantes*, aproxima el cuadrante en el que se encuentra el cliente y saca la ruta a seguir.

El cliente va llamando al servidor cuando actualiza su posición actual, de esta manera el servidor actualiza la ruta y genera la nueva instrucción. Una vez que la ruta ha finalizado, el servidor se lo indica al cliente con una instrucción de finalización. Véase, “Su destino se encuentra a la derecha, el recorrido ha finalizado”.

4.2. Cliente

Capítulo 5

holi

Capítulo 6

Conclusiones y Trabajo Futuro

Conclusiones del trabajo y líneas de trabajo futuro.

Antes de la entrega de actas de cada convocatoria, en el plazo que se indica en el calendario de los trabajos de fin de máster, el estudiante entregará en el Campus Virtual la versión final de la memoria en PDF. En la portada de la misma deberán figurar, como se ha señalado anteriormente, la convocatoria y la calificación obtenida. Asimismo, el estudiante también entregará todo el material que tenga concedido en préstamo a lo largo del curso.

Chapter 7

Introduction

Introduction to the subject area. This chapter contains the translation of Chapter 1.

Chapter 8

Conclusions and Future Work

Conclusions and future lines of work. This chapter contains the translation of Chapter 6.

Apéndice **A**

Título del Apéndice A

Contenido del apéndice

| | |
|----------|----------|
| Apéndice | B |
|----------|----------|

Título del Apéndice B

Este texto se puede encontrar en el fichero Cascaras/fin.tex. Si deseas eliminarlo, basta con comentar la línea correspondiente al final del fichero TFMTeXiS.tex.

*–¿Qué te parece desto, Sancho? – Dijo Don Quijote –
Bien podrán los encantadores quitarme la ventura,
pero el esfuerzo y el ánimo, será imposible.*

*Segunda parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes*

*–Buena está – dijo Sancho –; firmela vuestra merced.
–No es menester firmarla – dijo Don Quijote–,
sino solamente poner mi rúbrica.*

*Primera parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes*

