



Trabajo de Fin de Grado

ULL-Navigation. Tecnología de
realidad virtual en entornos
universitarios

Alejandro Hernández Padrón

La Laguna, 9 de mayo de 2018

D. **Francisco de Sande González**, con DNI nº 42.067.050-G profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

C E R T I F I C A

Que la presente memoria titulada:

“ULL-Navigation. Tecnología de realidad virtual en entornos universitarios”

ha sido realizada bajo su dirección por D. **Alejandro Hernández Padrón**, con DNI nº 42221533L

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 9 de mayo de 2018

Agradecimientos

Mis agradecimientos al profesor Francisco de Sande González por su labor como tutor de este proyecto y orientando este trabajo.
XXX

Licencia



© Esta obra está bajo una licencia de Creative Commons
Reconocimiento-NoComercial-CompartirIgual 4.0
Internacional.

Resumen

Este documento constituye el trabajo de investigación del alumno durante el proceso de desarrollo de una aplicación para dispositivos móviles Android mediante el uso de técnicas de realidad virtual(AR) combinadas con tecnicas de geolocalizacion .

Partimos de los conocimientos de programación en Java adquiridos en la asignatura: “Diseño arquitectónico y patrones” cursada en el itinerario de Ingeniería del Software. Esta asignatura, impartida en el tercer curso del Grado en Ingeniería Informática de la Universidad de La Laguna, ha sido la que ha sentado los fundamentos a partir de los cuáles se ha desarrollado la aplicación.

Durante este proyecto, el alumno ha conseguido adquirir independencia en su trabajo, visión y planificación realizando tareas de investigación, desarrollo y documentación, que han dado como resultado la obtención de conocimientos durante el proceso de trabajo.

También se ha investigado sobre las posibles tecnologías de realidad virtual presentes en la actualidad y su uso e implementacion en sistemas Android.

Palabras clave: Aplicaciones Android, Java, dispositivos móviles, programación, Realidad Aumentada XXX

Abstract

MI RESUMEN en Inglés

Para hacer un espaciado grande XXX

Keywords: *Application for Android, Java, mobile devices, programming, Augmented Reality ... XXX*

Índice general

Introducción	1
1. Objetivos	2
2. Herramientas y Tecnologías	3
2.1. Introduccion	3
2.1.1. Android Studio	3
2.1.2. LaTeX	4
2.1.3. Github	5
2.2. Tecnologías utilizadas	5
2.2.1. El Sistema Operativo Android	5
2.2.2. Los Beacons	6
2.2.3. CouchBase Server	13
2.2.4. La librería AltBeacon	15
2.2.5. El algoritmo de trilateración	16
3. XXX Título capítulo	18
3.1. Aplicaciones móviles en entornos universitarios	18
3.1.1. Descarga automática de material	19
4. La aplicación BulletPoint	20
4.1. Casos de uso elegidos	20
4.1.1. Localización de transporte público, horarios e información de la parada	20
5. Conclusiones y líneas de trabajo futuras	23
5.0.1. Conclusiones	23
5.0.2. Conclusions	23
5.0.3. Líneas de trabajo futuras	23
6. Presupuesto y puesta en marcha	25

ULLNavigation

II

Bibliografía

26

Índice de figuras

2.1. Android Studio, un IDE flexible e intuitivo.	4
2.2. Uno de los beacons de la compañía Estimote	7
2.3. Uno de los beacons de la compañía Estimote en formato Pegatina	7
2.4. Representación de un beacon emitiendo mediante Bluetooth a un dispositivo móvil	7
2.5. Interior de un beacon de Estimote	8
2.6. Números identificativos de los beacons	9
2.7. Ejemplificación del rango de un beacon	9
2.8. La aplicación de Clevedon School	11
2.9. La aplicación de Levi's Stadium	12
2.10. La aplicación del Aeropuerto Internacional de Orlando	13
2.11. Sincronización de CouchBase Server con CouchBase Lite mediante Sync Gateway	14
2.12. En el algoritmo de trilateración se utilizan las localizaciones conocidas de varios puntos de referencia y las distancias entre el sujeto y cada punto de referencia.	16
3.1. Servicios de localización a través de la Universidad	18
3.2. La biblioteca de la Universidad de Salamanca contiene más de 1.000.000 de ejemplares lo que puede dificultar la localización de algunos títulos.	19

Introducción

Este documento comprende el trabajo de investigación y desarrollo realizado por el alumno en la consecución de su Trabajo de Fin de Grado (TFG), con el que culminará sus estudios del Grado en Ingeniería Informática cursados en la Escuela Superior de Ingeniería y Tecnología (ESIT) de la Universidad de la Laguna (ULL).

Capítulo 1

Objetivos

Este TFG tiene los siguientes objetivos principales:

- Por un lado se pretende ampliar los conocimientos en tecnologías móviles en el sistema operativo *Android* [1] y en el desarrollo de aplicaciones para este sistema operativo.
- Otro objetivo presente en este TFG es que el alumno investigue y profundice en las técnicas y tecnologías de realidad virtual presentes en la actualidad.
- Por otro lado, también se pretende que el alumno se familiarice con el uso de herramientas de control de versiones utilizando *Github* [2] y de edición de textos técnicos utilizando *LaTeX* [3].
- Por último, tras las labores de investigación y recopilación de información correspondientes, se espera que el alumno aplique los conocimientos adquiridos para desarrollar una aplicación funcional que cubra las necesidades propuestas.

Capítulo 2

Herramientas y Tecnologías

Este capítulo tiene como objetivo presentar las distintas herramientas software y tecnologías empleadas por el alumno en el desarrollo de ULL-*Navigation*.

2.1. Introduccion

A continuación se explicarán brevemente las distintas herramientas software utilizadas en el proyecto.

2.1.1. Android Studio

Android Studio [4] es el IDE (Entorno de Desarrollo Integrado) oficial para el desarrollo de aplicaciones en Android, basado en IntelliJ IDEA [5]. Android Studio ofrece una serie de funcionalidades que han facilitado a la desarrolladora numerosas tareas, entre las cuales podemos destacar:

- Un sistema de compilación basado en Gradle[6] que ha simplificado tanto la inserción de dependencias de las distintas librerías que se han tenido que utilizar, como la compilación de la aplicación.
- Un emulador rápido y fácil de utilizar que ha ayudado a visualizar las distintas pantallas durante el desarrollo aunque no ha sido de mucha utilidad para probar el funcionamiento al ser dependiente la app de la tecnología Bluetooth.
- La facilidad para publicar cambios a aplicaciones ya funcionando sin tener que eliminar y volver a crear un nuevo APK parando la app.
- Un sistema de visualización de las diferentes pantallas muy completo, con soporte visual para añadir componentes y cambiar atributos fácilmente.

- Un sistema de depuración, con una interfaz sencilla e intuitiva.



Figura 2.1: Android Studio, un IDE flexible e intuitivo.

Se ha utilizado este IDE frente a otros como Eclipse + ADT [7] debido a que en la actualidad es el IDE oficial con soporte de Google. Se ha preferido aprender a utilizar este entorno con vistas al futuro, ya que parece que se consolidará como el preferido para los desarrolladores Android.

2.1.2. LaTeX

LaTeX es un sistema de composición de textos, orientado a la creación de documentos que presenten una alta calidad tipográfica. Por sus características y posibilidades, es usado especialmente en la generación de artículos y publicaciones científicas que incluyen, entre otros elementos, expresiones matemáticas, gráficos o figuras.

LaTeX está formado por un gran conjunto de macros de TeX, escrito por Leslie Lamport en 1984, con la intención de facilitar el uso del lenguaje de composición tipográfica, creado por Donald Knuth. LaTeX es software libre bajo licencia LPPL.

Se ha decidido utilizar este sistema debido al carácter profesional que aporta a los documentos. Ha sido una buena oportunidad para aprender a usar un sistema de composición de texto como este, ya que en un futuro puede ser beneficioso el saber manejar esta herramienta.

Si bien es cierto, que el uso de esta herramienta frente a otros editores más familiares ha sido algo tedioso en el inicio, es verdad que una vez acostumbrada a su uso ha resultado ser muy eficaz. En el proceso de aprendizaje se recurrió principalmente a manuales por internet, alguno a destacar en español sería [8]

2.1.3. Github

GitHub[2] es una forja (plataforma de desarrollo colaborativo) para alojar proyectos que utiliza el sistema de control de versiones Git. Utiliza el framework Ruby on Rails por GitHub, Inc. (anteriormente conocida como Logical Awesome). Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

Se ha decidido crear un repositorio en esta plataforma para poder llevar un control y una trazabilidad del proyecto. El tutor y el alumno han trabajado en este repositorio de manera conjunta. En el caso del tutor, principalmente para revisar el seguimiento semanal y llevar un control de las tareas. En el caso del alumno, para tener un repositorio donde subir los distintos elementos que se han ido generando a lo largo del trabajo. Aparte de este repositorio, también se ha abierto un segundo repositorio [9] asociado a la oficina del software libre (OSL) para subir el código una vez terminado como parte del programa de apoyo a trabajos finales libres (PATFL) [10] de la ULL.

Mediante el uso de este repositorio, el alumno ha conseguido ampliar sus conocimientos en Git y familiarizarse con la interfaz de GitHub. Previamente se había utilizado como repositorios GitLab, SVN y RTC en otros proyectos, por lo que no ha sido una complicación mayor utilizar este sistema.

2.2. Tecnologías utilizadas

A continuación se revisan las distintas tecnologías utilizadas en el desarrollo de la aplicación.

2.2.1. El Sistema Operativo Android

Android es un sistema operativo que emplea Linux en la interfaz del hardware. Los componentes del SO subyacentes se codifican en C o C++ pero las aplicaciones se desarrollan en Java. De esta manera Android asegura una amplia operatividad en una gran variedad de dispositivos debido a dos hechos: la interfaz en Linux ofrece gran potencia y funcionalidad para aprovechar el hardware, mientras que el desarrollo de las aplicaciones en Java permite que Android sea accesible para un gran número de programadores conocedores del código.

Este SO fue diseñado principalmente para dispositivos móviles con pantalla táctil: smartphones, tablets y otros dispositivos como televisores o automóviles. Fue desarrollado inicialmente por Android Inc., empresa que fue respaldada económicamente por Google y más tarde adquirida por esta misma empresa.

Actualmente tiene una gran comunidad de desarrolladores creando aplicaciones para extender la funcionalidad de los dispositivos. A fecha de hoy existen más de un millón de aplicaciones disponibles para la tienda oficial de Apps de Android, Google Play [11] sin tener en cuenta las aplicaciones de otras tiendas no oficiales, como por ejemplo, la tienda de aplicaciones de Samsung Apps [12].

2.2.2. Los Beacons

Los “*Beacons*” [13] (la traducción del término sería a “*balizas*” o “*faros*”) son una tecnología emergente que desde hace algunos años intenta abrirse paso en el mercado. Como su propio nombre indica, estos dispositivos intentan ser un mecanismo de guía, dando una solución al posicionamiento en interiores, donde otras tecnologías, como el GPS [14] o el Wifi dejan de funcionar o resultan imprecisas. Sin embargo, estos no son los únicos usos de los beacons, actualmente muchas empresas están ampliando sus usos a otros campos.

A continuación se intentará responder a las preguntas más frecuentes que nos pueden surgir con respecto a esta tecnología:

- ¿Qué es un beacon?
- ¿Cómo funcionan estos dispositivos?
- ¿Qué rango de alcance poseen?
- ¿Con qué dispositivos móviles son compatibles?
- ¿Qué ventajas y desventajas tienen con respecto a otras tecnologías?
- ¿Qué usos se le ha dado a esta tecnología hasta ahora?

¿Qué es un Beacon?

Para quienes no conozcan este término, en el marco en el que nos movemos, hace referencia a un pequeño dispositivo (sus tamaños varían de uno a otro, pero siempre de tamaño reducido) que emite señales de onda corta utilizando la tecnología Bluetooth [15]. Estas señales contienen una pequeña cantidad de información que es recibida por dispositivos móviles con tecnología Bluetooth dentro de un rango de cobertura variable dependiendo del beacon y su configuración. Normalmente, la intensidad de esta señal y su frecuencia son configurables.

El funcionamiento de un beacon es sencillo: El dispositivo emite una señal ininterrumpida (véase Figura 2.4) que es captada por los dispositivos móviles dentro de su radio de cobertura. La señal nos ofrece información que sirve para



Figura 2.2: Uno de los beacons de la compañía Estimote



Figura 2.3: Uno de los beacons de la compañía Estimote en formato Pegatina



Figura 2.4: Representación de un beacon emitiendo mediante Bluetooth a un dispositivo móvil

detectar y localizar estos dispositivos. Esta señal es captada por una aplicación previamente instalada en un dispositivo móvil que esté programada para recibirla y reaccionar de manera acorde a la información recibida.

Hay que tener en cuenta que esta señal es unidireccional: los beacons son capaces de enviar pero no están preparados para recibir. También hay que tener en cuenta, que la mayoría de los beacon actuales en el mercado transmiten información preconfigurada, confiando en la aplicación móvil para utilizar la información; sin embargo es muy posible que esto cambie en un futuro, ampliando las posibilidades de los beacons.

¿Como funcionan estos dispositivos?

Los beacons usan Bluetooth Low Energy (BLE) [16], una versión del protocolo Bluetooth diseñada para usar mucha menos energía y enviar menos información.

Los beacons funcionan con baterías cuyo tiempo de vida depende de la configuración establecida, teniendo en cuenta la emisión de la señal (intensidad y frecuencia) y tiempo de hibernación. Sus tiempos de vida son variables, pudiendo durar desde un mes hasta varios años.



Figura 2.5: Interior de un beacon de Estimote

Independientemente de lo que se pueda pensar, los beacons en sí mismos no transmiten información significativa, transmiten identificadores cortos (a modo de información configurable), que son interpretados por una aplicación que sepa lo que ha de hacer cuando detecte esta información y que es la que se encarga de procesarla y realizar la acción pertinente.

Este identificador se divide en tres partes:

- “*UUID*” [17] : corresponde con una ID dada por el fabricante e identifica el beacon en cuestión, nosotros utilizaremos el término MAC para referirnos a este identificador.
- ID Superior : configurables y utilizadas con un significado específico que puede identificar una acción o parámetro.
- ID Inferior: configurables y utilizadas al igual que la superior con un significado específico que se puede usar para identificar una acción o parámetro.

¿Qué rango de alcance poseen?

Actualmente los beacons en el mercado presentan un rango de aproximadamente 70 metros de radio sin obstáculos. Está demostrado que este rango disminuye significativamente al atravesar paredes de metal o ladrillo; otros materiales disminuyen en menor medida el rango.

UUID	B8407F30- F5F8- 466E- AFF9- 25556857F66D
MINOR	34956
MAJOR	58549

Figura 2.6: Números identificativos de los beacons

Las aplicaciones que trabajan con beacons suelen definir acciones en tres rangos de distancia (véase Figura 2.7) principalmente:

- Lejos: diseñado para que el dispositivo móvil pueda lanzar una acción cuando el usuario se encuentre en el rango exterior de un beacon, es decir, cuando se entra en el rango del beacon.
- Cerca: diseñado para que el dispositivo móvil pueda lanzar una acción cuando el usuario se encuentre en el rango interior del beacon.
- Inmediato: diseñado para que el dispositivo móvil pueda lanzar una acción cuando el usuario se encuentre muy cercano al beacon.

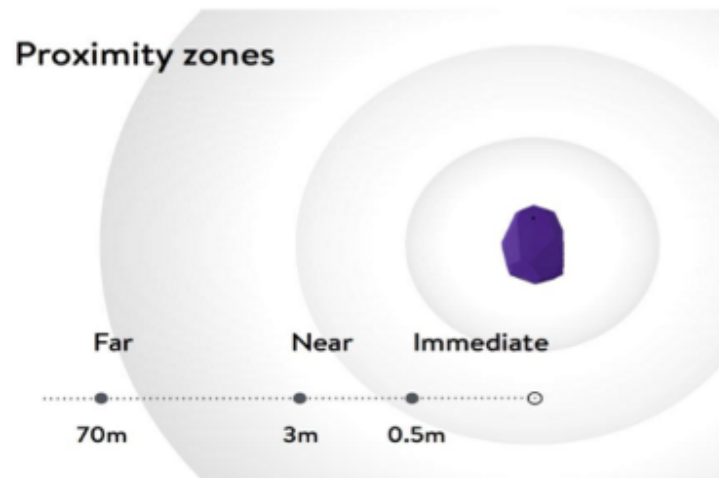


Figura 2.7: Ejemplificación del rango de un beacon

Sin embargo, esto depende de como esté diseñada la aplicación, es posible lanzar acciones a una distancia determinada sin tener en cuenta estos rangos mencionados anteriormente, ya que en todo momento es posible conocer la distancia a la que nos encontramos del beacon.

¿Con qué dispositivos funcionan?

Las beacons son compatibles con todos los dispositivos que soporten el protocolo Bluetooth Low Energy, pero para que las señales de los beacons sean detectadas por un dispositivo, se ha de tener activado el Bluetooth.

En dispositivos con IOS7 [18] o superior el dispositivo móvil puede estar constantemente buscando dispositivos BLE. El sistema operativo se encarga de activar las aplicaciones implicadas cuando registra la entrada de un beacon en su rango incluso estando cerradas las aplicaciones.

En dispositivos Android [1] el sistema operativo no está preparado para escanear dispositivos BLE, por lo que son las aplicaciones las que tienen que encargarse de escanear las proximidades buscando beacons. Esto supone que las aplicaciones tienen que estar funcionando y despiertas (aunque sea en segundo plano). Existen librerías que solucionan esta limitación haciendo que la aplicación escanee cada cierto tiempo. Sin embargo, no es muy eficaz y suele tener incompatibilidades con el sistema operativo. Un ejemplo de estas incompatibilidades lo podemos ver en el hilo de discusión [19] de los desarrolladores de la librería AltBeacon que se ha utilizado para el desarrollo de **BulletPoint** y que mencionaremos más tarde.

Por último en dispositivos Windows Phone [?] o Blackberry [20] existen diferentes niveles de compatibilidad pero en los que soportan BLE, su funcionamiento es similar al de los dispositivos Android, por lo que no nos pararemos a analizarlo.

¿Qué ventajas y desventajas tienen con respecto a otras tecnologías?

A la hora de hablar de los beacons existen una serie de ventajas pero también podemos encontrar algunas desventajas que revisaremos a continuación.

Las principales ventajas que se distinguen a la hora de hablar de los beacons son las siguientes:

- A diferencia de la tecnología GPS, la activación del Bluetooth consume mucha menos batería.
- Las aplicaciones desarrolladas suelen ser dependientes de la red de datos al necesitar información.
- A diferencia de la tecnología GPS, sigue funcionando con precisión en el interior de los edificios.

En cuanto a las desventajas, podemos destacar las siguientes:

- Dependen de aplicaciones instaladas en el dispositivo móvil para funcionar.
- Es necesario tener el Bluetooth activado, lo que consume batería durante el tiempo que esté activado.

- Su utilidad depende de la voluntad de terceros de utilizar estos dispositivos, configurarlos y distribuir las aplicaciones.

¿Qué usos se le ha dado a esta tecnología hasta ahora?

Por ahora esta tecnología se ha utilizado en entornos muy diversos y con distintas funcionalidades. Entre los más conocidos podríamos destacar los siguientes:

Clevedon School

Este ejemplo es bastante significativo para nosotros ya que se aplicó en el mismo entorno en el que queremos trabajar, una institución de enseñanza universitaria. Después de desplegar cerca de 1200 iPads la universidad de Clevedon (Figura 2.8) utilizó esta tecnología junto con su aplicación universitaria ya existente.



Figura 2.8: La aplicación de Clevedon School

Han sido capaces de crear una manera fácil para que los profesores puedan añadir recursos que se envían automáticamente a los alumnos transitando diferentes zonas en diferentes horarios. Para realizar este trabajo de manera eficiente fue necesaria la creación de una interfaz para la gestión de los recursos en las diferentes beacons.

Esta interfaz junto con la aplicación móvil es capaz de:

- Programar los recursos para distribuirse a una hora del día especificada.
- Programar el material para ser distribuido en un momento determinado durante una clase o evento.
- Poner los recursos a disposición del alumnado que se encuentre en una localización específica.

Utilizando estos tres recursos, la aplicación, la interfaz y los beacons han sido capaces de crear un entorno interactivo y eficiente motivando tanto al profesorado como al alumnado.

Cleveland Cavaliers Stadium y Levi's Stadium

Dos de los ejemplos más conocidos han sido los despliegues que se han realizado en estos dos estadios (Ver Figura 2.9).

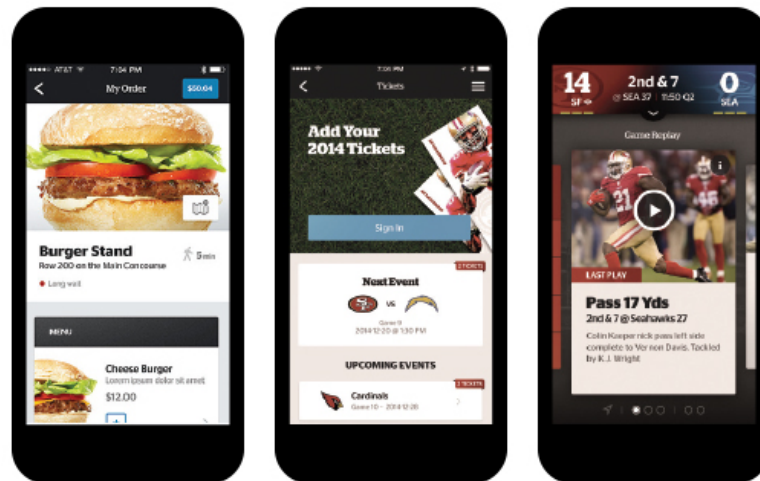


Figura 2.9: La aplicación de Levi's Stadium

Por un lado tenemos el despliegue del estadio de Levi's , cuya intención ha sido la de ayudar a sus seguidores a navegar por el estadio dadas sus dimensiones. En este caso los beacons (de la compañía Aruba Networks) se utilizan en conjunto con puntos de acceso y repetidores situados por toda la infraestructura de manera que queda el estadio cubierto. Con la aplicación los fans también son capaces de ver repeticiones de las jugadas y pedir comida directamente desde sus dispositivos móviles.

Un punto importante de este despliegue ha sido la monitorización continua del funcionamiento de los beacons, incluyendo si están en funcionamiento o necesitan batería nueva. Los beacons son también más económicos que los puntos de acceso WiFi, lo cual les ha beneficiado.

En el caso del estadio de Cleveland, los beacons se encargan de proveer al usuario de información personalizada dependiendo del lugar y la hora. En algunos casos vídeos, ofertas promocionales y contenido adicional.

Orlando Int'l Airport

Otro despliegue exitoso de esta tecnología ha sido en el aeropuerto internacional

de Orlando, donde mediante el uso de los beacons y de una aplicación móvil propia han sido capaces de proporcionar una serie de funcionalidades de vital importancia en una infraestructura como el aeropuerto:

- Navegación paso por paso a través de cerca de 1000 establecimientos o servicios dentro del aeropuerto.
- Actualizaciones inmediatas de la información de los vuelos.
- Instrucciones a puntos de interés críticos como puntos de recogida de equipaje, puertas de embarque o puestos de información.

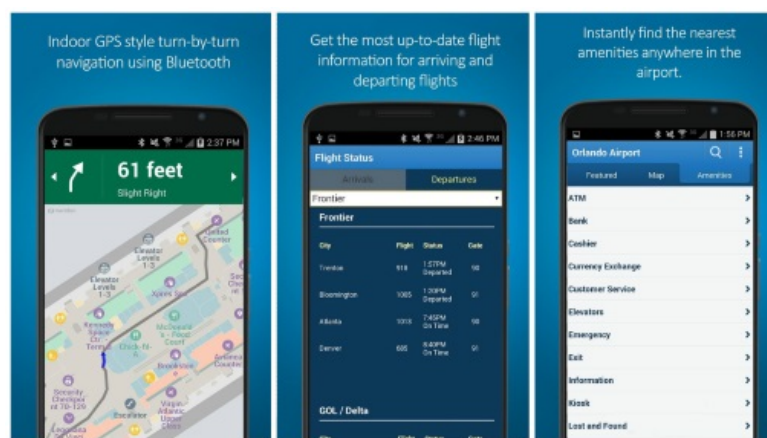


Figura 2.10: La aplicación del Aeropuerto Internacional de Orlando

El siguiente punto sería ampliar la opción a los establecimientos de ofrecer anuncios o promociones, opción que mantienen abierta y no se descarta en un futuro.

Esta información ha sido extraída de: [21]

2.2.3. CouchBase Server

Couchbase Server [22] es una base de datos NoSQL [23] con una arquitectura distribuida orientada al rendimiento, escalabilidad y disponibilidad. Permite desarrollar aplicaciones de manera sencilla y rápida combinando la flexibilidad del JSON [24] y la tecnología NoSQL.

¿Por qué utilizar Couchbase Server?

Hemos decidido utilizar esta tecnología por su flexibilidad y potencia para almacenar documentos fácilmente. Además resulta muy sencillo integrarla con

la tecnología móvil mediante el uso de una base de datos reducida dentro del dispositivo móvil (véase Figura 2.11). La sincronización con la base de datos principal del servidor se realiza mediante una puerta de sincronización.

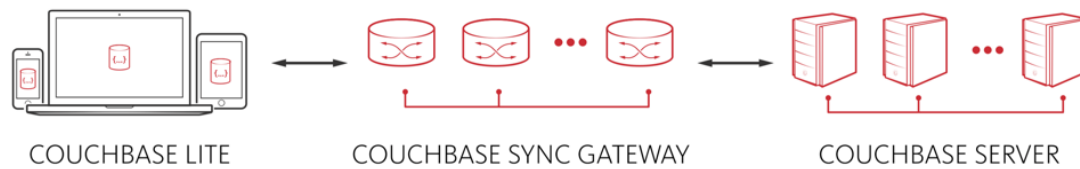


Figura 2.11: Sincronización de CouchBase Server con CouchBase Lite mediante Sync Gateway

En este caso se ha configurado el servidor en un ordenador portátil haciendo uso de las indicaciones de la página web del producto [25]. Se procederá a desglosar brevemente los pasos seguidos a la hora de configurar el servidor.

Configuración de la arquitectura

Para configurar el servidor se han seguido los siguientes pasos:

- Descargar la versión Community del producto desde la página web siguiendo el enlace [26].
- Seguir los pasos de la página de desarrolladores para la instalación y configuración [27].
- Una vez configurado CouchBase Server procederemos a descargar Sync Gateway que será el servicio que se encargue de sincronizar el contenido de nuestra aplicación al servidor, para ello lo descargaremos de la página al igual que el servidor siguiendo el enlace [26] .
- Para vincular el servidor con Sync Gateway es necesario hacer uso de un fichero de configuración con el que lanzaremos el servicio Sync Gateway.
- Una vez configurado Sync Gateway, ya tenemos el canal de configuración entre el servidor y la aplicación, para utilizar la base de datos móvil seguiremos los pasos desglosados en: [28] .

En este caso se ha tenido que conectar el dispositivo al ordenador. Tanto el ordenador como el dispositivo móvil han de estar en la misma red y hemos utilizado la dirección IP de la máquina para realizar las peticiones del móvil al servidor (alojado en el portátil) al utilizar la API. De esta manera se ha comprobado el funcionamiento del servidor, del servicio de sincronización y de la base de datos versión Lite en el dispositivo móvil.

2.2.4. La librería AltBeacon

¿Qué es AltBeacon?

Se puede definir AltBeacon como una especificación que:

- Define el formato del protocolo publicitario que los beacons transmiten a través de Bluetooth Low Energy (BLE).
- Intenta crear un mercado abierto y competitivo para implementaciones usando proximidad con los beacons.
- Puede ser utilizada gratuitamente, sin cuotas ni compromisos.
- No favorece a ningún proveedor sobre otro. Las limitaciones vienen determinadas por los estándares técnicos de cada uno.

A continuación se profundizará en el funcionamiento y configuración de la librería AltBeacon, que cumple con la especificación AltBeacon y que se ha utilizado para trabajar en Android.

Configuración

Para trabajar con esta librería en Android Studio solo hemos tenido que importarla mediante el uso de Gradle a nuestro proyecto como se explica en [29].

También es posible descargarla desde la página oficial, donde además podremos encontrar diferentes versiones de la misma. Para ello podemos hacer uso del siguiente enlace [30] y seleccionar la versión deseada.

Funcionamiento

La funcionalidad de esta librería se centra en dos elementos principales:

- Monitorización (“*Monitoring*”) que sería algo como supervisar, saber qué beacons se encuentran en una región o si ha entrado o salido un beacon de una región.
- Rastreo (“*Ranging*”), que permite saber a que distancia se encuentran los beacons en todo momento dentro de una región.

Utilizando estas dos funcionalidades la aplicación es capaz de controlar, monitorizar y rastrear los distintos beacons en una determinada región. En la página web de la librería [31] se pueden encontrar ejemplos básicos de cómo se realizan estas dos funciones en la sección “*Samples*”, además en la sección “*Documentación*” se pueden encontrar también algunos artículos, que pueden resultar interesantes dependiendo del tipo de aplicación que se esté desarrollando.

2.2.5. El algoritmo de trilateración

Durante el desarrollo de este trabajo se nombrará en diferentes puntos el término trilateración. A continuación explicaremos qué es la trilateración y cómo se ha aplicado el algoritmo al trabajar con beacons en **BulletPoint**.

¿Qué es la trilateración?

La trilateración es un método matemático para determinar las posiciones relativas de objetos utilizando la geometría de triángulos de forma análoga a la triangulación. La triangulación utilizada en la tecnología GPS, utiliza medidas de ángulo junto con al menos una distancia para calcular la posición del sujeto.

A diferencia de ésta, aunque muchas veces se confundan los términos por la similitud, la trilateración utiliza las localizaciones conocidas de dos o más puntos de referencia y las distancias entre el sujeto y cada punto de referencia (véase Figura 2.12). Para determinar de forma única y precisa la localización relativa de un punto en un plano bidimensional utilizando solo la trilateración, se necesitan generalmente al menos 3 puntos de referencia.

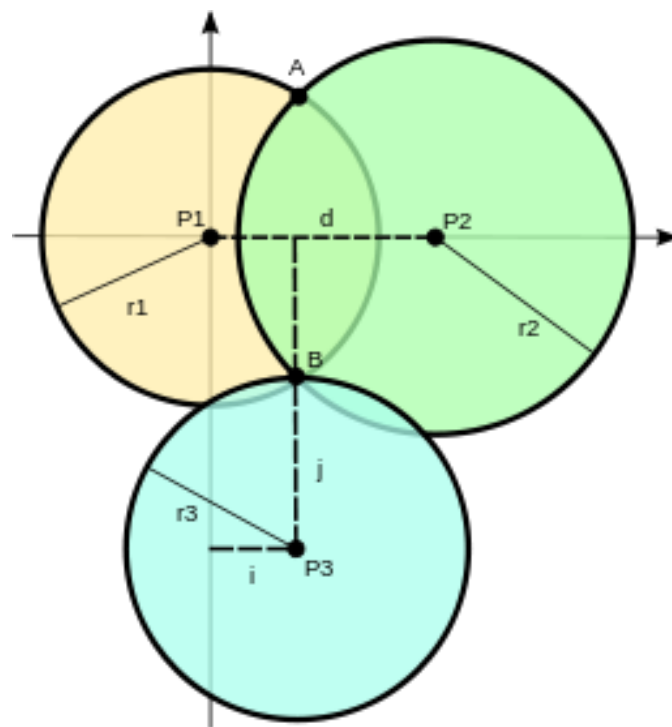


Figura 2.12: En el algoritmo de trilateración se utilizan las localizaciones conocidas de varios puntos de referencia y las distancias entre el sujeto y cada punto de referencia.

En el desarrollo de la aplicación se ha utilizado un algoritmo de trilateración implementado en Java que utiliza el algoritmo de Levenberg-Marquardt [32] de Apache Commons Math [33] tal y como explica el autor en [34].

En nuestro caso, las posiciones de los beacons son determinadas en una imagen separados entre sí a menos de 70 metros. Los beacons se sitúan en estas posiciones elegidas en la imagen y se obtiene una posición “X” e “Y” que queda registrada para cada beacon en la aplicación (Listado ??).

Las distancias medidas hasta la posición de cada beacon vienen dadas por la distancia registrada por la aplicación. Aparte de estos dos datos, para obtener una posición “X” e “Y” donde representar el punto, es necesario aplicar una escala que convierta los metros a píxeles. Tras la aplicación del algoritmo de trilateración, se consigue como resultado la posición del usuario en la imagen representado por “X” e “Y” en píxeles.

Capítulo 3

XXX Título capítulo

En este capítulo ...

3.1. Aplicaciones móviles en entornos universitarios

- XXX
- XXX



Figura 3.1: Servicios de localización a través de la Universidad

El funcionamiento sería el siguiente: el usuario transita por las inmediaciones del campus universitario. El usuario accede al sistema de navegación dentro de la aplicación, la cual le muestra entonces su ubicación como un punto de color sobre

el mapa del campus. Este mapa tiene marcados puntos de interés que contienen información de diferente tipo dependiendo del punto marcado: nombre, historia, página web, teléfono de contacto, trámites asociados... son algunos de los datos que podría mostrar. El mapa se va actualizando dependiendo de la posición del usuario permitiendo volver a la vista más alejada en cualquier momento para una visualización más general.

3.1.1. Descarga automática de material

Otro posible uso de la tecnología beacon tiene que ver con las bibliotecas o lugares de almacenamiento de material. El estudiante se acercaría a la biblioteca buscando un libro específico. Partimos de que en la app estaría registrada la localización de los libros disponibles en los estantes. De esta manera la aplicación indicaría al alumno o alumna la posición del libro que busca. Para lugares amplios donde hay gran cantidad de material (véase la Figura 3.2), incluso podría guiar al usuario por las instalaciones hasta llegar a su objetivo, informarle del número de ejemplares disponibles o de la fecha prevista de entrada de algún título.



Figura 3.2: La biblioteca de la Universidad de Salamanca contiene más de 1.000.000 de ejemplares lo que puede dificultar la localización de algunos títulos.

Capítulo 4

La aplicación BulletPoint

Basándonos en los casos de uso revisados en el capítulo anterior, en este capítulo se discutirán los casos de uso que han sido elegidos e implementados en la aplicación **BulletPoint**. Comentaremos la aplicación centrándonos en el desarrollo de la misma, así como en diferentes partes a destacar del código que puedan resultar interesantes.

4.1. Casos de uso elegidos

4.1.1. Localización de transporte público, horarios e información de la parada

Objetivo

El objetivo de este caso de uso es el conocer, en tiempo real, qué autobuses pasan por la parada en la que nos encontramos, hacia dónde se dirigen, y cuánto tiempo falta para que lleguen a la parada. En caso de necesidad de información adicional, la aplicación está preparada para remitirnos a navegar por la página web [35] de la empresa Transportes interurbanos de Tenerife, S.A.(TITSA), donde podemos ver datos adicionales sobre el autobús seleccionado.

En un principio nuestra idea era utilizar simplemente la API de TITSA, puesto que se creía que proporcionaría toda esta información; sin embargo, en algunos casos los destinos no se correspondían con la realidad y las rutas aparecían erróneas. Ante esta situación, se contactó con personal de TITSA involucrado en el desarrollo de esta API quien nos comentó que esto ocurría en algunos casos por la manera en la que estaba planteada la API.

En el primer método, utilizando una librería de peticiones HTTP, se envía una consulta por GET a la API de TITSA que devuelve una respuesta en XML. Esta respuesta se parsea con un handler de XML (véase Listado ??).

```
1 package com.bulletpoint.ull.bulletpoint.busclasses;
2
3 public class Arrival {
4     //This class is used to store the data coming from TITSA.
5     private String stopCode; //Stop code, each beacon is associated with one of these.
6     private String stopName;
7     private String destination; //It's coming faulty from TITSA, so we will parse TITSA webpage
8     //    looking for it.
9     private String hour;
10    private String travelId;
11    private String lineNumber;
12    private String minutesForArrival;
13
14    public String getStopCode() {
15        return stopCode;
16    }
17
18    public void setStopCode(String stopCode) {
19        this.stopCode = stopCode;
20    }
21
22    public String getStopName() {
23        return stopName;
24    }
25
26    public void setStopName(String stopName) {
27        this.stopName = stopName;
28    }
29
30    //...
31 }
```

Listado 4.1: La clase *Arrival* donde quedan contenidos los datos de cada llegada.

En cuanto se accede al módulo de parking de **BulletPoint**, la aplicación muestra una lista con los diferentes aparcamientos y el número de plazas de cada recinto. Esta lista se obtiene mediante una consulta a una API diseñada en colaboración con Alberto Morales, quien se ha encargado de configurar el servicio para facilitar a la aplicación la información necesaria. De la lista de aparcamientos, el usuario ha de seleccionar al que quiere acceder, identificando así la imagen y zonas a dibujar. Esta selección da paso a un escáner que comienza a detectar la posición del usuario y a situarlo en la imagen. Si el usuario se encuentra en una de las zonas designadas para entrar o salir del parking, se lanzará la acción, que en este caso consiste en una petición de apertura contra el servidor. En este caso cada zona posee un identificador de barrera que es el que usa la petición para determinar que barrera debe abrir.

Todos estos factores han de ser tomados en cuenta a la hora de realizar el despliegue de los beacons. Otro factor a tener en cuenta es la altura. Los dispositivos es recomendable levantarlos cierta altura sobre el nivel del suelo. A la hora de probarlos se ha intentado, en la medida de lo posible, mantenerlos a un nivel elevado sobre la altura de las personas y siempre intentando mantener esta medida de altura para los diferentes beacons. En cuanto al despliegue de la aplicación, el código fuente de **BulletPoint** se encuentra disponible para su descarga en [9], bajo licencia Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional [36].

Capítulo 5

Conclusiones y líneas de trabajo futuras

En este capítulo se presentarán las conclusiones a las que se ha llegado tras realizar este TFG y discutiremos posibles líneas de trabajo futuras.

5.0.1. Conclusiones

5.0.2. Conclusions

Nowadays, beacon technology is still on a development phase. By itself can be quite limited due to its functioning. The physical and positional limitations require a deep analysis in order to place devices the best way. On the other hand, each specific beacon provider tries to use his own SDKs to develop the applications. Depending on the case, fees might be applied so, working with this technology can end up being not money-saving and make things difficult for many developers.

5.0.3. Líneas de trabajo futuras

- Acceso a instalaciones deportivas, educativas o de diversa índole.
- Contratación de servicios en diferentes puntos: alquiler de bicicletas en puntos de alquiler, pago de entradas en las inmediaciones de un evento.
- Anuncios o promociones de diverso tipo, relacionados tanto con la universidad como con establecimientos comerciales cercanos al campus.
- Servicios de guía dentro del campus, con puntos de interés para el alumnado.

Son múltiples las posibilidades que se le pueden dar a esta tecnología. En otros sectores como el turismo o la medicina también están empezando a tener relevancia,

con lo que podemos confirmar que no es una tecnología aislada, sino que poco a poco se está abriendo paso en el mercado.

Capítulo 6

Presupuesto y puesta en marcha

Bibliografía

- [1] Android: **Android** 2008, [<https://www.android.com/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 2, 10
- [2] Github: **Repositorio Github** 2008, [<https://github.com/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 2, 5
- [3] LaTeX3 Project: **LaTeX** 1985, [<https://www.latex-project.org/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 2
- [4] IntelliJ: **Android Studio** 2014, [<https://developer.android.com/studio/index.html>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 3
- [5] IntelliJ IDEA: **IntelliJ IDEA** 2001, [<https://www.jetbrains.com/idea/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 3
- [6] **Gradle** 2007, [<http://gradle.org/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 3
- [7] Google, Inc: **Eclipse con ADT plugin** 2012, [<https://marketplace.eclipse.org/content/android-development-tools-eclipse>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 4
- [8] **Manual de Latex en español** 2004, [<http://www.fing.edu.uy/~canale/latex.pdf>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 4
- [9] **Repositorio de la aplicación ULL-NAVIGATION** [<https://github.com/alehdezp/TFG-ULL-NAVIGATION>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 5, 22
- [10] **Plataforma del Programa de Apoyo a Trabajos Finales Libres** [<http://osl.ull.es/programa-apoyo-trabajos-finales-libres/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 5

- [11] **Google Play** [<https://play.google.com/store>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 6
- [12] **Samsung Apps** [<http://www.samsung.com/es/apps/mobile/galaxyapps/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 6
- [13] **Beacon** 2014, [<http://meridianapps.com/news/2014/11/04/beacons>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 6
- [14] **Sistema de Posicionamiento global** [https://es.wikipedia.org/wiki/Sistema_de_posicionamiento_global]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 6
- [15] Jaap Haartsen y Mattisson Sven: **Bluetooth** 1994, [<https://es.wikipedia.org/wiki/Bluetooth>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 6
- [16] Nokia: **Bluetooth Low Energy** 2010, [https://en.wikipedia.org/wiki/Bluetooth_low_energy]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 7
- [17] **UUID Universal Unique Identifier** [https://en.wikipedia.org/wiki/Universally_unique_identifier]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 8
- [18] Apple Inc: **IOS7** [https://es.wikipedia.org/wiki/IOS_7]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 10
- [19] **Discusiones del funcionamiento de AltBeacon en Android Deep Sleep mode** [<https://github.com/AltBeacon/android-beacon-library/issues/21>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 10
- [20] Blackberry: **Blackberry** 1999, [<https://es.wikipedia.org/wiki/BlackBerry>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 10
- [21] Mareco D: **3 Brilliant example of how beacon technology is transforming mobility** [<http://www.securedgenetworks.com/blog/3-brilliant-examples-of-how-beacon-technology-is-transforming-mobility>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 13

- [22] **CouchBase Server** [<http://www.couchbase.com/nosql-databases/couchbase-server>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 13
- [23] **Tecnología NoSQL** [<https://es.wikipedia.org/wiki/NoSQL>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 13
- [24] **El formato de texto JSON** [<https://es.wikipedia.org/wiki/JSON>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 13
- [25] **Página web de CouchBase** [<http://www.couchbase.com/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 14
- [26] **Descarga de CouchBase Server** [<http://www.couchbase.com/nosql-databases/downloads>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 14
- [27] **Guía de Instalación de CouchBase Server** [<http://developer.couchbase.com/documentation/server/4.5/getting-started/installing.html>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 14
- [28] **Guía de Desarrollo usando CouchBase Lite** [<http://developer.couchbase.com/documentation/mobile/1.2/develop/training/build-first-android-app/index.html>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 14
- [29] **Dependencia Gradle de AltBeacon** [<https://altbeacon.github.io/android-beacon-library/configure.html>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 15
- [30] **Diferentes versiones de Alt Beacon** [<https://altbeacon.github.io/android-beacon-library/download.htm>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 15
- [31] **Ejemplos de utilización de la librería AltBeacon** [<https://altbeacon.github.io/android-beacon-library/samples.html>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 15
- [32] **Algoritmo de Levenberg-Marquardt** [https://en.wikipedia.org/wiki/Levenberg-Marquardt_algorithm]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 17

- [33] **Librería Apache Commons** [<http://commons.apache.org/proper/commons-math/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 17
- [34] **Algoritmo de Trilateración** [<https://github.com/lemmingapex/Trilateration>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 17
- [35] **Página web de Transportes interurbanos de Tenerife, SA** [<http://www.titsa.com/>]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 20
- [36] **Licencia Creative CommonsReconocimiento-NoComercial-CompartirIgual 4.0 Internacional** [https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es_ES]. [[Disponible electrónicamente. Último acceso, julio de 2018]]. 22