



UNIVERSIDAD UTE

**FACULTAD DE CIENCIAS DE LA INGENIERÍA E
INDUSTRIAS**

**CARRERA DE INGENIERÍA INFORMÁTICA Y
CIENCIAS DE LA COMPUTACIÓN**

**APLICACIÓN ANDROID DE UBICACIÓN EN INTERIORES
PARA INVIDENTES POR MEDIO DE UNA INTERFAZ DE
INDICACIONES SONORAS**

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN**

RICARDO GABRIEL CONDE MUÑOZ

DIRECTOR: ING. DIEGO ORDÓÑEZ, Ph.D.

Quito, agosto 2020

© Universidad UTE. 2020

Reservados todos los derechos de reproducción

FORMULARIO DE REGISTRO BIBLIOGRÁFICO

TRABAJO DE TITULACIÓN

DATOS DE CONTACTO	
CÉDULA DE IDENTIDAD:	1723583371
APELLIDO Y NOMBRES:	CONDE MUÑOZ RICARDO GABRIEL
DIRECCIÓN:	LAS CASAS
EMAIL:	ricardo.conde@ute.edu.ec
TELÉFONO FIJO:	023200632
TELÉFONO MOVIL:	0987759430

DATOS DE LA OBRA	
TÍTULO:	APLICACIÓN ANDROID DE UBICACIÓN EN INTERIORES PARA INVIDENTES POR MEDIO DE UNA INTERFAZ DE INDICACIONES SONORAS.
AUTOR O AUTORES:	CONDE MUÑOZ RICARDO GABRIEL
FECHA DE ENTREGA DEL PROYECTO DE TITULACIÓN:	2020-07-30
DIRECTOR DEL PROYECTO DE TITULACIÓN:	ING. DIEGO ORDÓÑEZ, Ph.D.
PROGRAMA	PREGRADO <input checked="" type="checkbox"/> POSGRADO <input type="checkbox"/>
TÍTULO POR EL QUE OPTA:	INGENIERO EN INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN.
RESUMEN: Mínimo 250 palabras	<p>La navegación en las personas, sin importar su condición, es una acción que debe ser desarrollada con la mayor autonomía posible, esto incluye de igual manera a las personas que poseen algún tipo de discapacidad visual que le dificulte realizar estas acciones. Si bien el posicionamiento en interiores ha sido abarcado en un gran número de investigaciones, aún no existe un consenso de qué tecnología o técnica es la mejor para dicho posicionamiento, lo que dificulta el desarrollo de un sistema que abarque</p>

	<p>tanto el posicionamiento interior combinado con un sistema de navegación para personas con discapacidad visual. En el presente trabajo se muestra una aplicación Android que permite a una persona invidente, mediante indicaciones sonoras, en una ubicación interior, ubicarse y desplazarse evitando obstáculos predefinidos. Mediante el uso de la metodología Mobile-D, se desarrolló una aplicación en la cual se utiliza una grid-map para la representación gráfica del ambiente interior y de los obstáculos que están presentes en este, la ruta a seguir en el ambiente interior es desarrollada con el uso del algoritmo A*, la orientación del usuario es detectada gracias al uso de la combinación del magnetómetro con el acelerómetro presentes en el teléfono, se implementaron dos enfoques de indicaciones sonoras basadas en voz (TTS) y pitidos (arpegios de piano) ambas emitidas en sonido estéreo, para el posicionamiento interior se utilizaron balizas BLE simuladas como puntos de acceso, y para la resolución del posicionamiento, se implementó la trilateración por el método del centroide. Mediante pruebas de simulación de un ambiente interior de 63 m², se pudo observar que el enfoque de indicaciones sonoras por pitidos (arpegios de piano) tuvo un mejor funcionamiento al momento de la navegación, aunque el enfoque de indicaciones sonoras por voz es más descriptivo al momento de la navegación.</p>
<p>PALABRAS CLAVES:</p>	<p>Navegación, Trilateración, Mapa-Cuadrícula, Orientación, Balizas BLE.</p>
<p>ABSTRACT:</p>	<p>Navigation in people, regardless of their condition, is an action that must be carried out with the greatest possible autonomy, this also includes people who have some type of visual disability that makes it difficult for them to carry out these actions. Although indoor positioning has been covered in a large number of investigations, there is still no consensus on which technology or</p>

	<p>technique is best for such positioning, making it difficult to develop a system that encompasses both indoor positioning combined with a navigation system for the visually impaired. This work shows an Android application that allows a blind person, by means of sound prompts, in an interior location, to locate and move around avoiding predefined obstacles. By using the Mobile-D methodology, an application was developed in which a grid-map is used to graphically represent the interior environment and the obstacles that are present in it, the route to follow in the interior environment is developed. With the use of the A * algorithm, the user's orientation is detected thanks to the use of the combination of the magnetometer with the accelerometer present in the phone, two approaches of voice-based sound indications (TTS) and beeps (piano arpeggios) were implemented. both emitted in stereo sound, simulated BLE beacons were used for internal positioning as access points, and trilateration by the centroid method was implemented for positioning resolution. Using simulation tests of a 63 m2 indoor environment, it was observed that the approach of sound indications by beeps (piano arpeggios) had a better performance at the time of navigation, although the approach of sound indications by voice is more descriptive when moment of navigation.</p>
<p>KEYWORDS</p>	<p>Navigation, Trilateration, Grid-Map, Orientation, BLE Beacons.</p>

Se autoriza la publicación de este Proyecto de Titulación en el Repositorio Digital de la Institución.



CONDE MUÑOZ RICARDO GABRIEL

C.I. 1723583371

DECLARACIÓN Y AUTORIZACIÓN

Yo, **CONDE MUÑOZ RICARDO GABRIEL**, C.I. 1723583371 autor del trabajo de titulación: **Aplicación Android de Ubicación en Interiores para Invidentes por medio de una Interfaz de Indicaciones Sonoras** previo a la obtención del título de **INGENIERO EN INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN** en la Universidad UTE.

1. Declaro tener pleno conocimiento de la obligación que tienen las Instituciones de Educación Superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación de grado para que sea integrado al Sistema Nacional de información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.
2. Autorizo a la BIBLIOTECA de la Universidad UTE a tener una copia del referido trabajo de titulación de grado con el propósito de generar un Repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Quito, 30 de julio del 2020



CONDE MUÑOZ RICARDO GABRIEL
C.I. 1723583371

CERTIFICACIÓN DEL TUTOR

En mi calidad de tutor, certifico que el presente trabajo de titulación que lleva por título **Aplicación Android de Ubicación en Interiores para Invidentes por medio de una Interfaz de Indicaciones Sonoras**, para aspirar al título de **INGENIERO EN INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN**, fue desarrollado por **CONDE MUÑOZ RICARDO GABRIEL**, bajo mi dirección y supervisión, en la Facultad de Ciencias de la Ingeniería e Industrias; y que dicho trabajo cumple con las condiciones requeridas para ser sometido a las evaluaciones respectivas de acuerdo a la normativa interna de la Universidad UTE.



ING. DIEGO ORDÓÑEZ, Ph.D.

DIRECTOR DEL TRABAJO

C.I. 1710449016

DEDICATORIA

Quiero extender esta dedicatoria a mi madre Jenny Muñoz por su apoyo incondicional durante toda mi vida, a mis familiares y amigos por su constante apoyo durante la etapa universitaria.

AGRADECIMIENTOS

Quiero extender mis agradecimientos, en primer lugar, a Dios por su constante protección durante todo el camino.

A mi madre, Jenny Muñoz, por su apoyo y presencia incondicional durante toda mi vida.

A mi tutor, Ing. Diego Ordóñez, Ph.D., por su guía durante el desarrollo del presente trabajo.

A los profesores presentes durante mi etapa universitaria, por sus constantes enseñanzas en clases.

ÍNDICE DE CONTENIDOS

	PÁGINA
RESUMEN	1
ABSTRACT	2
1. INTRODUCCIÓN	3
1.1. JUSTIFICACIÓN	3
1.2. TRABAJOS RELACIONADOS	3
1.2.1. REPRESENTACIÓN GRÁFICA DE UN AMBIENTE INTERIOR Y ALGORITMO DE RUTA	3
1.2.2. ORIENTACIÓN DEL USUARIO	4
1.2.3. INDICACIONES SONORAS	4
1.2.4. POSICIONAMIENTO INTERIOR	5
1.3. MARCO TEÓRICO	6
1.3.1. SISTEMA OPERATIVO ANDROID	6
1.3.1.1. Arquitectura	6
1.3.1.2. Versiones	7
1.3.2. SISTEMAS DE POSICIONAMIENTO EN INTERIORES	7
1.3.2.1. Descripción	7
1.3.3. BLUETOOTH LOW ENERGY	8
1.3.3.1. Historia	8
1.3.3.2. Descripción	8
1.3.3.3. Balizas BLE	9
1.3.4. TRILATERACIÓN	9
1.3.4.1. Descripción	9
1.3.4.2. Funcionamiento	9
1.3.5. SENSORES	10
1.3.5.1. Acelerómetro	10
1.3.5.2. Magnetómetro	11
1.3.5.3. Sensores del teléfono	11
1.3.6. ALGORITMO A*	11
1.3.6.1. Descripción	11
1.4. OBJETIVOS	12
1.4.1. OBJETIVO GENERAL	12

1.4.2. OBJETIVOS ESPECÍFICOS	12
2. METODOLOGÍA	13
2.1. DESCRIPCIÓN METODOLOGÍA MOBILE-D	13
2.2. EXPLORACIÓN	13
2.3. INICIACIÓN	13
2.4. PRODUCCIÓN	14
2.4.1. REPRESENTACIÓN GRÁFICA DEL AMBIENTE INTERIOR	14
2.4.2. ALGORITMO DE RUTA	15
2.4.3. ORIENTACIÓN DEL USUARIO	15
2.4.4. UTILIZACIÓN DE TECNOLOGÍA BLE	16
2.4.5. INDICACIONES SONORAS	17
2.4.6. POSICIONAMIENTO INTERIOR	17
2.5. ESTABILIZACIÓN Y PRUEBAS DEL SISTEMA	18
3. RESULTADOS Y DISCUSIÓN	20
3.1. ESTRUCTURA DE LA APLICACIÓN	20
3.1.1. VERSIÓN DE API	20
3.1.2. ESTRUCTURA DE CLASES	20
3.1.3. ESTRUCTURA BASE DE DATOS	20
3.1.4. INTERFACES GRÁFICAS DE USUARIO	21
3.2. REPRESENTACIÓN GRÁFICA DEL AMBIENTE INTERIOR	23
3.2.1. GRID-MAP (MAPA DE CUADRÍCULA)	23
3.2.2. OBSTÁCULOS	24
3.3. ALGORITMO DE RUTA	26
3.3.1. APLICACIÓN DEL ALGORITMO A*	26
3.4. ORIENTACIÓN DEL USUARIO	27
3.4.1. UTILIZACIÓN DE SENSORES DEL TELÉFONO	27
3.5. UTILIZACIÓN DE TECNOLOGÍA BLE	29
3.5.1. ADMINISTRACIÓN DE BALIZAS BLE	29
3.6. INDICACIONES SONORAS	31
3.6.1. DEFINICIÓN Y UTILIZACIÓN DE INDICACIONES SONORAS	31
3.7. POSICIONAMIENTO INTERIOR	35
3.7.1. CALIBRACIÓN DE BALIZAS	35
3.7.2. DISTANCIA ENTRE BALIZA Y TELÉFONO	36

3.7.3. APLICACIÓN DE LA TRILATERACIÓN	36
4. CONCLUSIONES Y RECOMENDACIONES	40
4.1. CONCLUSIONES	40
4.2. RECOMENDACIONES	41
BIBLIOGRAFÍA	43
ANEXOS	47

ÍNDICE DE TABLAS

	PÁGINA
Tabla 1. Herramientas de Software.	13
Tabla 2. Herramientas de Hardware.	14
Tabla 3. Categorías consideradas para la creación de la grid-map.	25
Tabla 4. Rangos para la definición del tipo de giro.	28
Tabla 5. Indicaciones sonoras pertenecientes al grupo “giros rectos”.	32
Tabla 6. Indicaciones sonoras pertenecientes al grupo “giros medios”.	32
Tabla 7. Indicaciones sonoras pertenecientes al grupo “avance e indicación de punto de interés”	33
Tabla 8. Balizas utilizadas durante el posicionamiento interior.	36

ÍNDICE DE FIGURAS

	PÁGINA
Figura 1. Pila de arquitectura de Android (<i>Google, 2019a</i>).	6
Figura 2. Diferencia entre posicionamiento exterior e interior (<i>Mendoza-Silva et al., 2019</i>).	7
Figura 3. Protocolos iBeacon, AltBeacon y EddyStone (<i>Hernández-Rojas et al., 2018</i>).	9
Figura 4. Cruce de tres círculos en un mismo punto (trilateración) (<i>Timón & de León, 2019</i>).	10
Figura 5. Ecuación de representación de un círculo (<i>Timón & de León, 2019</i>).	10
Figura 6. Funcionamiento de un acelerómetro (<i>Stanley & Lee, 2018</i>).	11
Figura 7. Sistemas de coordenadas en Android (orientación natural) (<i>Google, 2019b</i>).	11
Figura 8. Fórmula algoritmo A* (<i>Needham & Hodler, 2019</i>).	12
Figura 9. Ejemplo de una grid-map de una ambiente interior (<i>Abu Doush et al., 2016</i>).	14
Figura 10. Comparativa entre algoritmo Dijkstra y A* (<i>Chen et al., 2017</i>).	15
Figura 11. Generación de un grafo en base a una grid-map (<i>Eraghi et al., 2015</i>).	15
Figura 12. Fórmula distancia Manhattan (<i>Qian et al., 2019</i>).	15
Figura 13. Fórmula filtro de paso bajo (<i>Asmara et al., 2017</i>).	16
Figura 14. Calibración ángulo azimuth en base a la orientación del usuario.	16
Figura 15. Fórmula para obtener la distancia entre una baliza y dispositivo central (<i>Nguyen et al., 2017</i>).	17
Figura 16. Fórmula filtro de paso bajo destinado al RSSI (<i>Mussina & Aubakirov, 2018</i>).	17

Figura 17. Conjunto de fórmulas para el método del centroide (<i>Subedi et al., 2016</i>).	18
Figura 18. Diagrama de clases de la aplicación.	20
Figura 19. Diagrama ERD de la base de datos de la aplicación.	21
Figura 20. Interfaces de usuario de la aplicación.	23
Figura 21. Diferencias entre grid-maps.	24
Figura 22. Ejemplo de un ambiente interior.	25
Figura 23. Análisis extra empleado al algoritmo A*.	26
Figura 24. Secuencia de navegación en una ruta de ejemplo.	27
Figura 25. Método de obtención del ángulo de un nodo próximo en una ruta generada.	28
Figura 26. Orientación de un usuario.	29
Figura 27. Función onScanResult para la detección de una baliza BLE.	30
Figura 28. Diferentes encabezados para la detección de balizas BLE.	30
Figura 29. Detección de balizas BLE.	31
Figura 30. Rutas realizadas para la prueba de las indicaciones sonoras.	34
Figura 31. Calibración de una baliza a un metro de distancia.	35
Figura 32. Código para calcular la distancia entre una baliza y un dispositivo central.	36
Figura 33. Datos para el cálculo de la trilateración mediante el método del centroide.	36
Figura 34. Código para calcular la trilateración mediante el método del centroide.	37
Figura 35. Posicionamiento interior priorizando el seguimiento de ruta.	37
Figura 36. Posicionamiento interior con actualización de posición.	38
Figura 37. Posicionamiento interior con actualización de posición fijada a su centro más próximo.	38

ÍNDICE DE ANEXOS

	PÁGINA
ANEXO 1. CÓDIGO FUENTE DE LA APLICACIÓN	47
ANEXO 2. MANUAL DE USUARIO DE LA APLICACIÓN	48

RESUMEN

La navegación en las personas, sin importar su condición, es una acción que debe ser desarrollada con la mayor autonomía posible, esto incluye de igual manera a las personas que poseen algún tipo de discapacidad visual que le dificulte realizar estas acciones. Si bien el posicionamiento en interiores ha sido abarcado en un gran número de investigaciones, aún no existe un consenso de qué tecnología o técnica es la mejor para dicho posicionamiento, lo que dificulta el desarrollo de un sistema que abarque tanto el posicionamiento interior combinado con un sistema de navegación para personas con discapacidad visual. En el presente trabajo se muestra una aplicación Android que permite a una persona invidente, mediante indicaciones sonoras, en una ubicación interior, ubicarse y desplazarse evitando obstáculos predefinidos. Mediante el uso de la metodología Mobile-D, se desarrolló una aplicación en la cual se utiliza una grid-map para la representación gráfica del ambiente interior y de los obstáculos que están presentes en este, la ruta a seguir en el ambiente interior es desarrollada con el uso del algoritmo A*, la orientación del usuario es detectada gracias al uso de la combinación del magnetómetro con el acelerómetro presentes en el teléfono, se implementaron dos enfoques de indicaciones sonoras basadas en voz (TTS) y pitidos (arpegios de piano) ambas emitidas en sonido estéreo, para el posicionamiento interior se utilizaron balizas BLE simuladas como puntos de acceso, y para la resolución del posicionamiento, se implementó la trilateración por el método del centroide. Mediante pruebas de simulación de un ambiente interior de 63 m², se pudo observar que el enfoque de indicaciones sonoras por pitidos (arpegios de piano) tuvo un mejor funcionamiento al momento de la navegación, aunque el enfoque de indicaciones sonoras por voz es más descriptivo al momento de la navegación.

Palabras Claves: Navegación, Trilateración, Mapa-Cuadrícula, Orientación, Balizas BLE.

ABSTRACT

Navigation in people, regardless of their condition, is an action that must be carried out with the greatest possible autonomy, this also includes people who have some type of visual disability that makes it difficult for them to carry out these actions. Although indoor positioning has been covered in a large number of investigations, there is still no consensus on which technology or technique is best for such positioning, making it difficult to develop a system that encompasses both indoor positioning combined with a navigation system for the visually impaired. This work shows an Android application that allows a blind person, by means of sound prompts, in an interior location, to locate and move around avoiding predefined obstacles. By using the Mobile-D methodology, an application was developed in which a grid-map is used to graphically represent the interior environment and the obstacles that are present in it, the route to follow in the interior environment is developed With the use of the A * algorithm, the user's orientation is detected thanks to the use of the combination of the magnetometer with the accelerometer present in the phone, two approaches of voice-based sound indications (TTS) and beeps (piano arpeggios) were implemented. both emitted in stereo sound, simulated BLE beacons were used for internal positioning as access points, and trilateration by the centroid method was implemented for positioning resolution. Using simulation tests of a 63 m² indoor environment, it was observed that the approach of sound indications by beeps (piano arpeggios) had a better performance at the time of navigation, although the approach of sound indications by voice is more descriptive when moment of navigation.

Keywords: Navigation, Trilateration, Grid-Map, Orientation, BLE Beacons.

1. INTRODUCCIÓN

1. INTRODUCCIÓN

1.1. JUSTIFICACIÓN

La forma en la que el mundo permite nuestra movilidad, tomando en cuenta que todos poseemos capacidades diferentes, puede ser crucial para tener un buen estilo de vida. La movilidad es un proceso fundamental humano. La forma en la que nuestro mundo funciona depende de formas de movilidad desiguales (Adey, 2017). Actualmente el posicionamiento en exteriores ha sido estandarizado y explorado mediante la utilización del GPS o técnicas que permiten conocer la posición de un usuario en una red celular. Sin embargo, tomando en cuenta el GPS, es posible que no brinde la información precisa requerida para moverse en un espacio interior. Necesitamos saber en dónde estamos y cuál es la dirección correcta a seguir (Kolodziej & Hjelm, 2017). Para abarcar el apartado interior, existen diferentes tecnologías IPS, una de ellas es la tecnología Bluetooth Low Energy, a través de balizas BLE, que logra una localización aceptable y consume menos energía que, por ejemplo, la utilización de Wifi (Z. Wang, Yang, & Dong, 2017). La navegación de personas con visión dentro de un espacio interior es desarrollada sin presentar mayor inconveniente, pero las personas que presentan una discapacidad visual regularmente requieren una planificación y coordinación con otras personas para este fin. Aunque ya existen varias soluciones que permiten tener una vida independiente a una persona con discapacidad visual, aún sigue siendo un reto el desarrollo de un sistema que combine una navegación para invidentes, y un sistema de posicionamiento en interiores (Tao, Ding, & Ganz, 2017).

1.2. TRABAJOS RELACIONADOS

1.2.1. REPRESENTACIÓN GRÁFICA DE UN AMBIENTE INTERIOR Y ALGORITMO DE RUTA

(Abu Doush, Alshatnawi, Al-Tamimi, Alhasan, & Hamasha, 2016) realizaron una aplicación móvil enfocada a personas ciegas o con alguna discapacidad visual, mediante la utilización de WiFi, Bluetooth e identificación de radiofrecuencias, que permita a los usuarios alcanzar un objeto determinado. Como algoritmo de ruta modifican el algoritmo Dijkstra agregando un conjunto de pautas enfocadas a personas ciegas, y como representación del ambiente interior realizan una grid-map con celdas de 40cm² de área. Obtuvieron una precisión de 10cm.

(Chen, Shaw, & Lin, 2017) realizaron un robot capaz de moverse en un ambiente interior evitando obstáculos aleatorios. Como parte de su trabajo se presenta una comparación entre Dijkstra, A* y Ant, teniendo mejor resultado A*. Como añadido se presentan mejoras implementadas al algoritmo A*. De igual forma (Qian, Ma, & Hong, 2019) realizan mejoras al algoritmo A* enfocado para robots en un ambiente interior. (Eraghi, López-Colino, de

Castro, & Garrido, 2015) proponen un nuevo algoritmo de navegación implementado en una grid-map enfocado a robots, entre su idea principal se basaron en como una persona ciega navega en un ambiente interior.

(Abdullah & Korial, 2015) realizaron una estructura de navegación enfocada a personas ciegas o con discapacidad visual mediante el uso del algoritmo A* en un ambiente interior, mediante pruebas de simulación, el algoritmo A* es un método confiable y válido para guiar a una persona ciega o con discapacidad visual.

1.2.2. ORIENTACIÓN DEL USUARIO

(Asmara, Al Huda, Andoko, & Handayani, 2017) realizaron un trabajo que permite a personas con discapacidad visual, mediante el uso de sensores de un teléfono móvil, caminar en línea recta mediante una guía auditiva.

(Satan, 2018) realizaron un sistema de navegación en ambientes interiores mediante la utilización de balizas Bluetooth, como algoritmo de ruta se utiliza Dijkstra, por motivos de experimentación se realizaron correcciones a los valores de dirección en el ambiente interior de prueba.

1.2.3. INDICACIONES SONORAS

(Cecílio, Duarte, & Furtado, 2015) realizaron un sistema de navegación basado en teléfonos móviles y sensores conectados, ofreciendo una navegación guiada e información sobre su entorno.

(Bai, Lian, Liu, Wang, & Liu, 2017) desarrollaron un dispositivo de guía inteligente, en forma de unas gafas, que ayuda a personas con discapacidad visual a realizar una navegación segura. Para personas ciegas se manejaron tres distintas indicaciones sonoras, mientras que para personas con problemas visuales (no totalmente ciegas) se ofrece una mejora basada en RA.

(Meliones & Sampson, 2018) realizaron una aplicación llamada Blind MuseumTourer, que está enfocada en una navegación en interiores para personas ciegas o con discapacidad visual que les permita visitas accesibles y autoguiadas a museos.

(Sato et al., 2017) presentan un trabajo denominado NavCog3, el cual es un sistema de navegación en interiores enfocado a personas con discapacidad visual en el cual se brinda información de navegación paso a paso, puntos de referencia y puntos de interés cercanos a la posición del usuario. (Kacorri, Ohn-Bar, Kitani, & Asakawa, 2018) proponen dos distintos tipos de medidas de comportamiento de ruta, enfocadas al diseño de ruta en la navegación en interiores paso a paso. En su investigación hacen uso de NavCog3.

(Oh, Kao, & Min, 2017) realizaron un sistema de navegación enfocado a personas con discapacidad visual en el cual las instrucciones de ruta son

cargadas en su teléfono cuando este llega a un lugar especificado para esto, el sistema no utiliza técnicas de posicionamiento en interiores.

(Cheraghi, Namboodiri, & Walker, 2017) desarrollaron un sistema llamado GuideBeacon, el cual es un sistema de orientación en interiores enfocado a personas con discapacidad visual o con alguna desorientación. Este sistema incluye la utilización de balizas BLE.

(Guerrón Paredes, Cobo, Martín, & Serrano, 2018) implementaron una aplicación basada en realidad virtual para personas ciegas en teléfonos inteligentes que permiten realizar una vista a un espacio interior desconocido de antemano. Para el desarrollo de la aplicación se utilizaron patrones de voz, pitidos, vibraciones y gestos.

1.2.4. POSICIONAMIENTO INTERIOR

(Cabrera-Goyes & Ordóñez-Camacho, 2018) realizaron una investigación en la cual definían la relación existente entre la distancia y RSSI, además, mediante trilateración, lograron obtener un posicionamiento con un alto promedio de precisión en un entorno de experimentación. Cabe señalar que no se utilizó ningún dispositivo especializado para IPS, únicamente se utilizó dispositivos bluetooth estándar.

(Nguyen, Johnson, Nguyen, & Randles, 2017) realizaron un enfoque de posicionamiento interior mediante la utilización de dispositivos BLE, en el cual hacen uso de un método de posicionamiento mejorado de la técnica de mínimos cuadrados para poder resolver la trilateración.

(Mussina & Aubakirov, 2018) realizaron una investigación sobre el posicionamiento interior, utilizando la trilateración y dispositivos BLE. Se evalúan distintas formas de filtrado de los valores de RSSI.

(Röbesaat, Zhang, Abdelaal, & Theel, 2017) realizaron un método de posicionamiento interior mediante la fusión de las técnicas de trilateración y dead reckoning (cálculo muerto), además de la aplicación del filtrado Kalman como algoritmo de fusión. Utilizaron un dispositivo Android y diferentes módulos BLE.

(Subedi, Kwon, Shin, Hwang, & Pyun, 2016) realizaron un estudio enfocado al posicionamiento interior mediante el uso del centroide ponderado, utilizando balizas BLE.

(W. Wang & Wang, 2015) realizaron una investigación enfocada en el posicionamiento interior para robots en el cual, mediante el uso de sensores inalámbricos, comprobaron el funcionamiento de la trilateración mediante centroide y centroide ponderado.

1.3. MARCO TEÓRICO

1.3.1. SISTEMA OPERATIVO ANDROID

Google define Android como una compilación de software abierto basado en el Kernel de Linux, cuya creación está destinada para un amplio número de dispositivos (Google, 2019a).

Android es un sistema operativo basado en Linux enfocado a dispositivos móviles; desarrollado en el año 2005 por la empresa homónima Android Inc., la cual en un futuro fue adquirida por Google. Desde un inicio, Google puso a disposición Android bajo una licencia Apache de código abierto, permitiendo que cualquiera que desee utilizar Android pueda hacerlo. Esto permitió que varias empresas desarrolladoras de hardware puedan hacer uso de este sistema operativo implementando sus propias modificaciones según sea el caso (DiMarzio, 2016).

Durante el desarrollo para Android se utilizan diferentes conjuntos de herramientas: Java, XML y Linux. Java es el lenguaje de programación principal para el desarrollo de aplicaciones para Android. XML es utilizado para describir el diseño de una aplicación, y para la creación del archivo AndroidManifest.xml que contiene información sobre las características y administración de la aplicación. El Kernel de Linux es la base del sistema operativo Android (Burd, 2016).

1.3.1.1. Arquitectura

Android está dividido en capas, las cuales realizan diferentes acciones que brindan un soporte al sistema operativo, además, presenta un sistema jerárquico, ya que la capa inferior brinda servicios a las capas superiores para su funcionamiento (Mahalik, Bommisetty, Skulkin, & Tamma, 2018). En la Figura 1 se presentan las diferentes capas de la arquitectura de Android.



Figura 1. Pila de arquitectura de Android (Google, 2019a).

1.3.1.2. Versiones

Android evoluciona constantemente, actualizando sus componentes. Cada versión de Android esta diferenciada por tres elementos: número de versión, nivel de API, y el nombre de un dulce; Ejemplo: Android 4.4 - API 19 – Kitkat. Al desarrollar una aplicación para Android, se debe especificar el nivel de API objetivo, dicho nivel de API es un número entero que se especifica al momento de la codificación de la aplicación (Gao, Li, Bissyand, & Klein, 2019).

Cuando se especifica un nivel mínimo de API, la aplicación se podrá ejecutar en un mayor número de dispositivos, pero perdería funcionalidades recientemente implementadas. Por otro lado, un nivel de API alto agrega más funcionalidades para el desarrollo de la aplicación, pero se limita el número de dispositivos en los cuales se puede ejecutar. Al momento de escoger un nivel de API se opta por el nivel mínimo aceptado (Burd, 2016).

1.3.2. SISTEMAS DE POSICIONAMIENTO EN INTERIORES

1.3.2.1. Descripción

Los sistemas de posicionamiento pueden estar enfocados a un ámbito global o local. Para el posicionamiento global existen diferentes sistemas que en conjunto son denominados Sistema Global de Navegación por Satélite (GNSS por sus siglas en inglés). El GNSS ha tenido un gran resultado para el posicionamiento global (Mendoza-Silva, Torres-Sospedra, & Huerta, 2019). La gran familiarización con los sistemas de posicionamientos se debe en mayor medida al uso del GPS, el cual ha tenido un buen funcionamiento en espacios exteriores, pero en ambientes interiores da resultados insatisfactorios. Desde inicios de los 2000, se han desarrollados sistemas que permitan superar los problemas del GPS en espacios interiores, los cuales son denominados IPS (Ferreira, Fernandes, Catarino, & Monteiro, 2017).

Los sistemas de posicionamiento local son aquellos que se utilizan o aplican para escenarios o aplicaciones donde el posicionamiento por GNSS no es conveniente. Los sistemas de posicionamiento en interiores (IPS por sus siglas en inglés) son los sistemas de posicionamiento locales que permiten obtener una posición dentro de edificios (Mendoza-Silva et al., 2019). La Figura 2 muestra la diferencia entre un posicionamiento exterior y un posicionamiento interior.

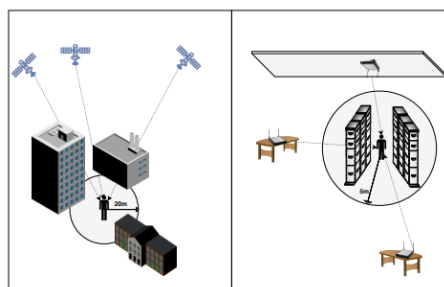


Figura 2. Diferencia entre posicionamiento exterior e interior (Mendoza-Silva et al., 2019).

1.3.3. BLUETOOTH LOW ENERGY

1.3.3.1. Historia

Bluetooth SIG decidió reinventar su pila de protocolos utilizados hasta el momento, enfocándose en la velocidad y en una arquitectura simple, para así cumplir con una mayor rapidez y con un menor consumo de energía.(Aftab, 2017). Bluetooth Low Energy o también conocido como Bluetooth Smart es una tecnología que fue introducida en el 2010 por el Bluetooth SIG. Sin embargo, esta tecnología estaba siendo desarrollada por Nokia entre los años 2001 y 2006, y la cual era conocida por el nombre Wibree. No fue hasta el año 2007 donde algunos miembros del Bluetooth SIG decidieron añadir la tecnología Wibree al Core de Bluetooth, terminando este proceso en el año 2010. Fue en la versión 4.0 de la especificación Bluetooth donde finalmente fue añadida como Bluetooth Low Energy (Bhargav, 2017).

1.3.3.2. Descripción

BLE es una tecnología de comunicación a corto alcance (Zhang et al., 2019). Esta tecnología opera en la banda ISM (Industrial, Científica y Médica) de 2400 a 2482 GHz. La banda está subdividida en 40 canales con espacio entre cada canal de 2Mhz. Estos 40 canales están subdivididos en 37 canales dedicados a datos, y los 3 canales restantes son publicitarios. Los canales publicitarios permiten que un dispositivo BLE pueda emitir paquetes publicitarios sin necesidad de una retroalimentación de un dispositivo receptor (Rosenthal & Reynolds, 2019).

BLE posee dos fases de comunicación las cuales son: publicidad y comunicación de datos (Das, Pathak, Chuah, & Mohapatra, 2016). En la primera fase, el dispositivo BLE (periférico) se anuncia a través de los 3 canales dedicados para este propósito (Das et al., 2016); el dispositivo que actúa como escáner de dispositivos BLE puede actuar con la información publicitaria o conectarse al dispositivo periférico para mayor información (Jameel & Dungen, 2015). En la segunda fase, una vez establecida una conexión con un dispositivo periférico BLE, se hace uso de los 37 canales dedicados a datos para la comunicación (Das et al., 2016); se envían y reciben paquetes entre el escáner y el periférico BLE (Jameel & Dungen, 2015).

Su utilización está enfocada en aplicaciones relacionadas con salud, estado de forma o posicionamiento en interiores, siendo en el campo de la salud una de las principales utilidades de los dispositivos con BLE. Dichos dispositivos destacan por su corto uso de energía y reducido tamaño (Bellavista et al., 2019).

Android ofrece soporte para BLE desde Android 4.3 (API 18), pero a partir de Android 5.0 (API 21) el soporte para BLE se revisó y se procedió a extraerlo en un paquete propio denominado android.bluetooth.le. Para el uso de BLE en Android es necesario otorgar los permisos de BLUETOOTH y

BLUETOOTH_ADMIN (Kriz, Maly, & Kozel, 2016). Además de los permisos de Bluetooth, también es necesario otorgar permisos de ubicación ya sea ACCESS_FINE_LOCATION o si la aplicación está enfocada a una API 28 o inferior se puede utilizar ACCESS_COARSE_LOCATION (Google, 2020a).

1.3.3.3. Balizas BLE

El protocolo beacon (baliza) fue presentado por primera vez por Apple, el cual se llama iBeacon. Dicho protocolo rápidamente se convirtió en un nuevo sistema que puede utilizarse para el anuncio de una posición o el posicionamiento en interiores. El protocolo beacon es un protocolo simple de transferencia, en el cual se agregan diferentes campos en el paquete de datos. Comúnmente, los campos más frecuentes son: UUID, mayor y menor. (Bellavista et al., 2019).

Las balizas BLE son dispositivos de reducido tamaño que transmiten información sin la necesidad de conectarse con un escáner BLE. Dicha información es enviada en los paquetes publicitarios en concordancia con la especificación del Bluetooth SIG. El Bluetooth SIG no ha definido un estándar para la realización de balizas, actualmente se pueden diferenciar tres implementaciones principales (Figura 3): iBeacon (Apple), Altbeacon (Radius Network), y EddyStone (Google) (Hernández-Rojas, Fernández-Caramés, Fraga-Lamas, & Escudero, 2018).

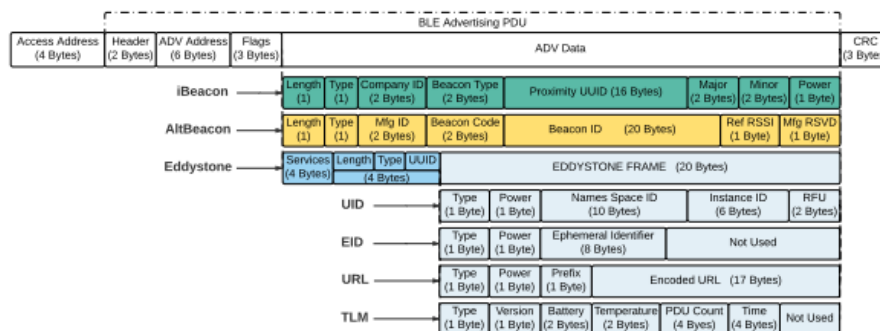


Figura 3. Protocolos iBeacon, AltBeacon y EddyStone (Hernández-Rojas et al., 2018).

1.3.4. TRILATERACIÓN

1.3.4.1. Descripción

Es un método que sirve para determinar la posición de un dispositivo inalámbrico en base a las distancias entre dicho dispositivo y los diferentes puntos de acceso alrededor del mismo. Básicamente, este método se basa en obtener la dirección de las señales para posteriormente calcular su intersección. Para su desarrollo requiere al menos tres puntos de acceso que emitan una intensidad de señal (Kolodziej & Hjelm, 2017).

1.3.4.2. Funcionamiento

Como se mencionó anteriormente, para determinar la posición de un punto dado en un plano, se necesitan al menos tres puntos de referencia. Dados

tres puntos P: {P1, P2, P3} y tres distancias R: {r1, r2, r3}, podemos trazar tres circunferencias C: {punto central, radio}; obtendríamos las siguientes circunferencias: C1: {P1, r1}, C2: {P2, r2}, C3: {P3, r3}. En el caso de que únicamente utilizemos dos circunferencias, cualquiera que estas sean, estas se cortarían en dos puntos distintos (a menos que ambas fueran tangentes), lo cual generaría ambigüedad con respecto de la ubicación de nuestro punto objetivo A, por lo cual, la tercera circunferencia ayuda a eliminar este tipo de ambigüedad (Timón & de León, 2019). En la Figura 4 se puede apreciar gráficamente lo descrito en este apartado.

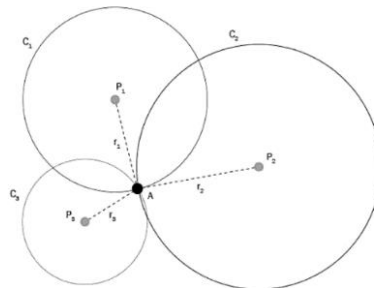


Figura 4. Cruce de tres círculos en un mismo punto (trilateración) (Timón & de León, 2019).

Cada circunferencia se puede expresar como una ecuación de segundo grado (Figura 5) tomando en cuenta que ya se tienen con los datos del radio y puntos centrales de cada circunferencia. Para la resolución utilizaremos un sistema de ecuaciones de las circunferencias, y nuestro resultado sería un punto $P(x, y)$ (Timón & de León, 2019).

$$(x - a)^2 + (y - b)^2 = r^2$$

Figura 5. Ecuación de representación de un círculo (Timón & de León, 2019).

1.3.5. SENSORES

1.3.5.1. Acelerómetro

Es un condensador, el cual está formado por dos placas de metal separadas por un material dieléctrico. Una de las placas es móvil, permitiendo detectar cualquier tipo de aceleración o fuerza que se aplique al dispositivo en el que se encuentra dicho condensador. La información que genera el sensor viene dada en voltios, que a su vez, provienen del movimiento que realizó el material dieléctrico debido a una acción de fuerza o aceleración provocada (Rubio González, Gómez Palomo, Letón Molina, Rodrigo San Juan, & Chaos García, 2017). La Figura 6 muestra el funcionamiento de un acelerómetro.

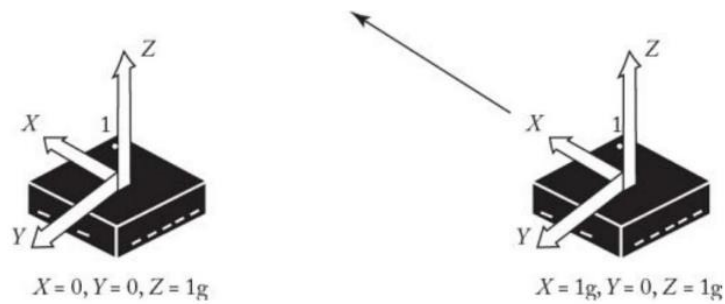


Figura 6. Funcionamiento de un acelerómetro (Stanley & Lee, 2018).

1.3.5.2. Magnetómetro

Un magnetómetro es utilizado para poder medir la intensidad del campo magnético de la Tierra (cantidades vectoriales presentes en la Tierra caracterizadas por su fuerza y dirección). Este sensor se puede subdividir en: magnetómetros vectoriales y magnetómetros escalares. Los primeros obtienen como resultado un vector 3D del campo magnético, mientras que el segundo obtiene como resultado la magnitud del vector del campo magnético (Stanley & Lee, 2018). Este sensor actúa regularmente como brújula en un dispositivo (Rubio González et al., 2017).

1.3.5.3. Sensores del teléfono

El sistema de coordenadas en Android consta de 3 ejes (x, y, z), visto de una orientación natural o predeterminada del dispositivo (Figura 7), entre los sensores que utilizan este sistema de coordenadas están el sensor de aceleración, gravedad, giroscopio, geomagnético, entre otros (Google, 2019b).

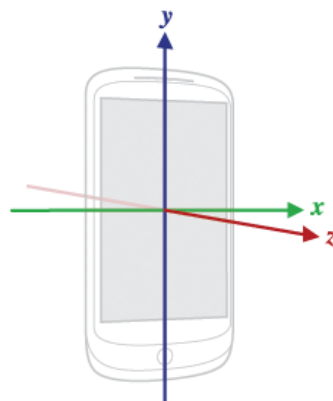


Figura 7. Sistemas de coordenadas en Android (orientación natural) (Google, 2019b).

1.3.6. ALGORITMO A*

1.3.6.1. Descripción

El algoritmo A* fue creado en el año 1968 por Peter Hart, Nils Nilsson y Bertram Raphael (Needham & Hodler, 2019). El algoritmo A* utiliza distancias positivas en un grafo, al igual que Dijkstra (Mathivet, 2018), pero a diferencia de este, A* encuentra una ruta más corta con mayor rapidez, esto gracias a

que utiliza información adicional (heurística) para el análisis del siguiente camino a tomar (Needham & Hodler, 2019). El funcionamiento del algoritmo A* radica en escoger cuál de las rutas generadas se expandirá en su bucle principal, esto lo logra gracias a la utilización de una heurística desde el punto actual hasta el punto final. En la Figura 8 se puede observar la fórmula utilizada por el algoritmo A* (Needham & Hodler, 2019). Donde $g(n)$ es el costo desde el nodo actual hasta el nodo final o destino y $h(n)$ es el cálculo de una heurística desde el nodo actual hasta el nodo final o destino.

$$f(n) = g(n) + h(n)$$

Figura 8. Fórmula algoritmo A* (Needham & Hodler, 2019).

1.4. OBJETIVOS

1.4.1. OBJETIVO GENERAL

Desarrollar un prototipo funcional de tipo aplicación Android que permita a una persona invidente, mediante una interfaz de indicaciones sonoras, ubicarse y desplazarse en espacios interiores, evitando obstáculos predefinidos.

1.4.2. OBJETIVOS ESPECÍFICOS

- Desarrollar un prototipo con una interfaz visual y auditiva para la aplicación Android. Enfatizando en las mejores indicaciones sonoras utilizadas para personas invidentes.
- Integrar el prototipo funcional con las aplicaciones IPS existentes en el proyecto de investigación.
- Determinar las configuraciones apropiadas para el uso de dispositivos BLE (Beacons) en interiores para el desarrollo del prototipo funcional.

2. METODOLOGÍA

2. METODOLOGÍA

2.1. DESCRIPCIÓN METODOLOGÍA MOBILE-D

Para el desarrollo del trabajo de investigación se tomó como base la metodología de desarrollo ágil enfocada a dispositivos móviles Mobile-D. Brevemente, Mobile-D está compuesta por cinco fases las cuales son: exploración, iniciación, producción, estabilización y prueba del sistema (Pani & Mishra, 2016).

2.2. EXPLORACIÓN

En esta fase se procedió con una conceptualización sobre la aplicación IPS desarrollada. Mediante la consulta bibliográfica, se procedió con la definición de varios puntos que son necesarios para el desarrollo de la aplicación. Dichos puntos son los siguientes: representación gráfica de los obstáculos (mapa del ambiente interior), algoritmo de ruta utilizado en el mapa, uso de sensores para detectar la orientación del usuario, detección y utilización de tecnología BLE, indicaciones sonoras y posicionamiento interior. Estos puntos serán tratados con más detalle en la fase de producción (numeral 2.4.).

2.3. INICIACIÓN

En esta fase se definieron las diferentes herramientas, tanto de software como de hardware, para el desarrollo del prototipo. En la Tabla 1 se presentan las diferentes herramientas de software utilizadas para el desarrollo del prototipo, a su vez, en la Tabla 2 se muestran las herramientas de hardware para el desarrollo del prototipo.

Tabla 1. Herramientas de Software.

Herramienta	Descripción
Windows 10 Home 64-bits	Sistema operativo donde fue desarrollado el prototipo.
Android Studio 3.4	IDE donde fue desarrollado el prototipo.
simpleUMLCE v0.01	Plugin utilizado para la generación del diagrama de clases de la aplicación.
Lucidchart	Página web enfocada a la realización de diagramas, donde se realizó el diagrama ERD de la base de datos de la aplicación.
Google Maps	Aplicación utilizada para comprobar y calibrar la brújula del dispositivo.
Beacon Simulator (Android)	Aplicación para poder simular una baliza BLE.
Beacon Scope (Android)	Aplicación para poder simular una baliza BLE, utilizada en dispositivos sin soporte BLE.
Locate Beacon (Android)	Aplicación utilizada para calibración del valor TxPower de una baliza BLE.
Quick Beacon (iOS)	Aplicación para poder simular una baliza BLE en dispositivos iOS.

Tabla 2. Herramientas de Hardware.

Herramienta	Descripción
HP Laptop 15-bs0xx, Intel Core i5-72000, CPU 2,50 GHz, 8 GB Ram.	Equipo donde se desarrolló el prototipo. Se utilizó como simulador de una baliza BLE.
Xiaomi Mi A2 (Android 10 - Pie)	Equipo principal donde se probó la aplicación. Se utilizó como simulador de una baliza BLE.
Huawei Mate 20 Lite (Android 10 - Pie)	Equipo de apoyo para el desarrollo. Se utilizó como simulador de una baliza BLE.
iPhone 11 Pro (iOS 13.6)	Se utilizó como simulador de una baliza BLE.
Sony Xperia Z3 (Android 6.6 – MarshMallow)	Equipo de apoyo para el desarrollo. Se utilizó como simulador de una baliza BLE.
Audífonos JBudsBand Wireless Neckband Headset	Audífonos de prueba para las indicaciones sonoras.

2.4. PRODUCCIÓN

En esta fase se procedió con el desarrollo del prototipo funcional, tomando como pauta los diferentes puntos mencionados en la fase de exploración.

2.4.1. REPRESENTACIÓN GRÁFICA DEL AMBIENTE INTERIOR

Primeramente, era necesario definir un espacio donde se pudiera representar el ambiente interior con los obstáculos y “puntos a considerar” que este posee. Para lograr este apartado, es necesario realizar un mapa del ambiente interior. Para esto se hace una representación a través de la creación de una grid-map (mapa de cuadrícula) (Abu Doush et al., 2016). En la Figura 9 se puede observar un ejemplo de una grid-map que representa un espacio interior.

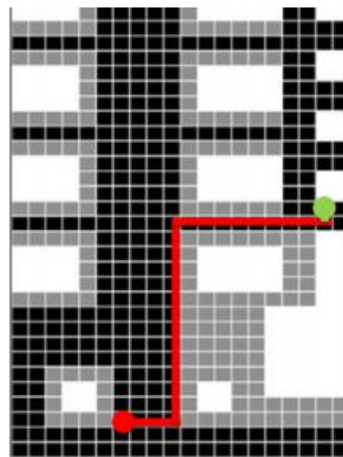


Figura 9. Ejemplo de una grid-map de una ambiente interior (Abu Doush et al., 2016).

Una vez definida la grid del ambiente interior, es necesario definir los obstáculos de este ambiente interior. Para esto se especifica una serie de colores distintivos que permiten determinar las celdas por las cuales es posible transitar (Abu Doush et al., 2016).

2.4.2. ALGORITMO DE RUTA

Para el desarrollo de este apartado se consideraron dos tipos de algoritmo (Dijkstra y A*) el rendimiento que presenta Dijkstra es menor eficiente que el que presenta A* en el peor de los casos (Figura 10) (Chen et al., 2017), por lo cual, en base a esta comparación, y en base a la implementación del algoritmo A* en una grid-map enfocado a personas ciegas o con alguna discapacidad visual (Abdullah & Korial, 2015) se hace uso del algoritmo A* para el desarrollo de la aplicación.

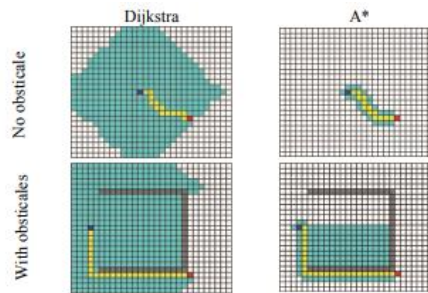


Figura 10. Comparativa entre algoritmo Dijkstra y A* (Chen et al., 2017).

Para lograr implementar el algoritmo A* en la grid-map del ambiente interior, es necesario generar un gráfico en base a la información del mapa (por donde se puede transitar y por donde no). Lo más eficiente es generar dicho gráfico en base a nuestro nodo o celda actual y evaluar los vecinos válidos para transitar a nuestro alrededor (Eraghi et al., 2015). Dicho procedimiento se puede observar en la Figura 11.

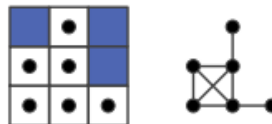


Figura 11. Generación de un grafo en base a una grid-map (Eraghi et al., 2015).

Para su funcionamiento, A* requiere la definición de una heurística, por lo cual se utilizó la distancia Manhattan para el algoritmo (Qian et al., 2019). La Figura 12 muestra la fórmula de la distancia Manhattan utilizada, donde n_x y n_y son las coordenadas del nodo actual, y $goal_x$ y $goal_y$ son las coordenadas del nodo destino.

$$H(n) = |n_x - goal_x| + |n_y - goal_y|$$

Figura 12. Fórmula distancia Manhattan (Qian et al., 2019).

2.4.3. ORIENTACIÓN DEL USUARIO

En la documentación de Android existe un apartado que muestra la forma de obtener la orientación del dispositivo en base al norte magnético de la tierra

mediante la combinación del acelerómetro y magnetómetro (Google, 2020b). La fusión del acelerómetro con el magnetómetro consiste en la ayuda por parte del acelerómetro en la corrección de la medición del campo magnético realizada por el magnetómetro en base a la información de inclinación obtenida por el acelerómetro (Asmara et al., 2017).

Se decidió aplicar un filtro de paso bajo a las lecturas obtenidas en bruto desde el acelerómetro y magnetómetro del dispositivo, esto debido a que las lecturas obtenidas a veces tienen información innecesaria denominada ruido, esto provoca un declive en la precisión de la información obtenida y por ende un rendimiento menor (Asmara et al., 2017). La Figura 13 muestra la fórmula de filtro de paso bajo utilizada, donde y_{i-1} es el valor obtenido anteriormente, x_i es el valor actual obtenido, y α es el valor umbral de frecuencia que se desea pasar.

$$y_i = \alpha x_i + (1 - \alpha) y_{i-1}$$

Figura 13. Fórmula filtro de paso bajo (Asmara et al., 2017).

Para fines de navegación y orientación, es necesario definir la orientación en la cual se va a realizar la representación del mapa interior, ya que no se cuenta con la información de en qué dirección se encuentra un obstáculo determinado. Basado en (Satan, 2018), el procedimiento para lograr esto consta en guardar la dirección (ángulo azimut en base al norte magnético obtenido por los sensores del teléfono) en la cual se va a proceder a realizar la grid-map del mapa interior (Figura 14).

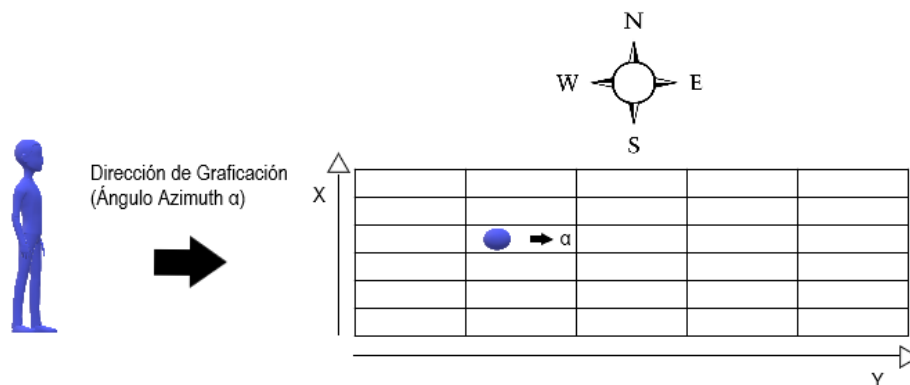


Figura 14. Calibración ángulo azimut en base a la orientación del usuario.

2.4.4. UTILIZACIÓN DE TECNOLOGÍA BLE

Para la detección de balizas BLE se procedió con la utilización de la API nativa de Android destinada a BLE (Google, 2020a). Debido a que no existe un estándar en el desarrollo de balizas (Figura 3) (Hernández-Rojas et al., 2018), para el manejo de las mismas se procedió con la utilización de la librería Android Beacon Library v2.17 (Radius Networks, 2019).

2.4.5. INDICACIONES SONORAS

En base a las indicaciones sonoras implementadas por: (Cecílio et al., 2015), (Bai et al., 2017), (Meliones & Sampson, 2018), (Sato et al., 2017), (Kacorri et al., 2018), (Oh et al., 2017), (Cheraghi et al., 2017), (Guerrón Paredes et al., 2018). Se procedió con la utilización de indicaciones sonoras basadas en voz y tonos, aplicables en diferentes situaciones que se puedan presentar en la navegación. Este apartado es explicado con mayor profundidad en la sección de resultados y discusión.

2.4.6. POSICIONAMIENTO INTERIOR

Primeramente, para el posicionamiento, es necesario obtener la distancia entre el dispositivo central (teléfono) y una baliza cualquiera, con la ayuda del RSSI y de su TxPower, para el cual se hace uso del registro de sombreado normal (Nguyen et al., 2017). La Figura 15 muestra la fórmula utilizada donde $RSSI$ es el valor RSSI emitido por la baliza, A es el TxPower de la baliza y n es la constante de pérdida de ruta.

$$d = 10^{\frac{RSSI-A}{10n}}$$

Figura 15. Fórmula para obtener la distancia entre una baliza y dispositivo central (Nguyen et al., 2017).

Los valores de RSSI obtenidos de la baliza en cuestión necesitan ser filtrados, para lo cual se aplica un filtro de paso bajo a ese valor (Mussina & Aubakirov, 2018). La Figura 16 muestra la fórmula del filtro utilizado donde α es valor umbral, $RSSI_n$ es el valor actual ingresado y $RSSI_{n-1}$ es el valor anterior registrado.

$$RSSI = \alpha * RSSI_n + (1 - \alpha) * RSSI_{n-1}$$

Figura 16. Fórmula filtro de paso bajo destinado al RSSI (Mussina & Aubakirov, 2018).

El valor de referencia para el exponente de pérdida de ruta recomendado para ambientes que se encuentren en edificios y a su vez que estos contengan obstáculos, oscila entre 4 y 6 (Röbesaat et al., 2017).

Para la resolución de la trilateración, se procedió con el uso del método de centroide (Subedi et al., 2016). La Figura 17 muestra el conjunto de fórmulas utilizadas para resolver la trilateración, donde X_i y Y_i son las coordenadas de las balizas que emiten la señal RSSI, W_i es la ponderación que se le da a cada baliza, y d_i es la distancia obtenida entre el dispositivo central (teléfono) y una baliza en cuestión.

$$\begin{aligned}
 x_w &= \frac{\sum_{i=1}^m x_i w_i}{\sum_{i=1}^m w_i} \\
 y_w &= \frac{\sum_{i=1}^m y_i w_i}{\sum_{i=1}^m w_i} \\
 w_i &= \frac{1}{d_i^p}
 \end{aligned}$$

Figura 17. Conjunto de fórmulas para el método del centroide (*Subedi et al., 2016*).

Para el cálculo del valor de w_i , es necesario definir un grado a la distancia, en este caso se implementó con un valor de 1 (W. Wang & Wang, 2015).

2.5. ESTABILIZACIÓN Y PRUEBAS DEL SISTEMA

En estas dos fases se procedió con la realización de pruebas del sistema. Mas específicamente, tomando como base los puntos tratados en la fase de producción, la representación del ambiente interior se realizó en base a simulaciones de diferentes ambientes interiores, permitiendo resaltar los diferentes apartados considerados para este fin.

Las pruebas de ruta, orientación, utilización de tecnología BLE, indicaciones sonoras y posicionamiento interior fueron realizadas en la representación en una grid-map de un ambiente interior real de 63m2.

El algoritmo de ruta se evaluó mediante la simulación de un camino que se pueda presentar en un ambiente interior, verificando su consistencia según se recorra el camino, y que además permita transitar por lugares permitidos. La ruta también es evaluada cuando se realiza las pruebas de indicaciones sonoras.

La orientación del usuario fue evaluada en el ambiente interior real, representada en la aplicación mediante una grid-map, permitiendo evaluar si la dirección del usuario tenía una concordancia aceptable entre la aplicación y la situación real.

La utilización de la tecnología BLE fue evaluada mediante la prueba de detección de balizas BLE por medio de la aplicación, comparando los datos establecidos en el simulador de balizas BLE con los datos obtenidos y mostrados en la aplicación.

Las indicaciones sonoras se evaluaron mediante una simulación de posicionamiento ideal, de manera táctil en el mapa, permitiendo seleccionar nuestra posición actual. A medida que las indicaciones sonoras nos guían el camino a seguir, se procedía a marcar la nueva posición de manera táctil en el mapa, permitiendo evaluar las indicaciones sonoras sin la interferencia de algún error de posicionamiento.

Para finalizar, se realizó las pruebas del posicionamiento interior, en la cual se evaluaron tres distintas acciones: seguimiento de posición o ruta,

actualización de posición, y actualización y centrado de posición a su celda más próxima.

Todos estos temas tratados son explicados con mayor profundidad en el apartado de resultados y discusión.

3. RESULTADOS Y DISCUSIÓN

3. RESULTADOS Y DISCUSIÓN

3.1. ESTRUCTURA DE LA APLICACIÓN

3.1.1. VERSIÓN DE API

La aplicación fue compilada en una versión Sdk de nivel 28, el nivel mínimo de Sdk fue determinado en 21, y el nivel Sdk de destino fue determinado en 26 (mínimo aceptado por Google Play a partir de la segunda mitad del 2018).

3.1.2. ESTRUCTURA DE CLASES

Entre las diferentes clases que presenta la aplicación, para el tema de representación gráfica de un espacio interior se encargan las clases “Imagen” y “Plano”, en la primera se muestra la imagen correspondiente al mapa interior para realizar el posicionamiento, y la segunda corresponde a la creación del mapa interior. Las diferentes operaciones con la base de datos se realizan a través de la clase “ConsultaBD”, el manejo de la tecnología BLE se realiza en la clase “BluetoothLE”, los cálculos necesarios para la trilateración se encuentran en la clase “Trilateracion”, y la generación de indicaciones sonoras se encuentra en la clase “Indicaciones”. La estructura de clases de aplicación se la puede observar en la Figura 18. El código fuente de la aplicación se puede visualizar en el ANEXO 1.

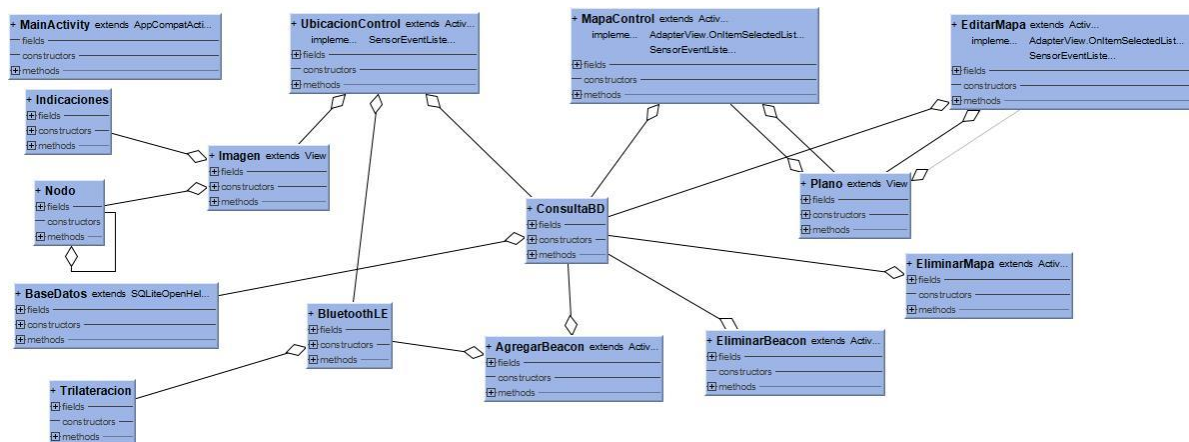


Figura 18. Diagrama de clases de la aplicación.

3.1.3. ESTRUCTURA BASE DE DATOS

Para la realización de la base de datos se procedió con la utilización de SQLite. La tabla “Mapa” guarda la información sobre el mapa interior creado, la tabla “Beacon” guarda información sobre la baliza que será utilizada posteriormente en el posicionamiento, y la tabla “Mapa_Beacon” guarda información sobre la relación entre un mapa y varias balizas que estén asociadas a este, además, almacena la posición de las balizas en el mapa. Cabe señalar que el tipo de dato “Blob” permitió guardar la imagen del mapa en formato de Byte [], y el campo “utiliza_beacon” en la tabla “Beacon” permite saber si una baliza puede ser utilizada en un mapa o a su vez eliminada de la

aplicación (valor true sí se utiliza, valor false sí no se utiliza). En la Figura 19 se puede observar el diagrama ERD de la base de datos.

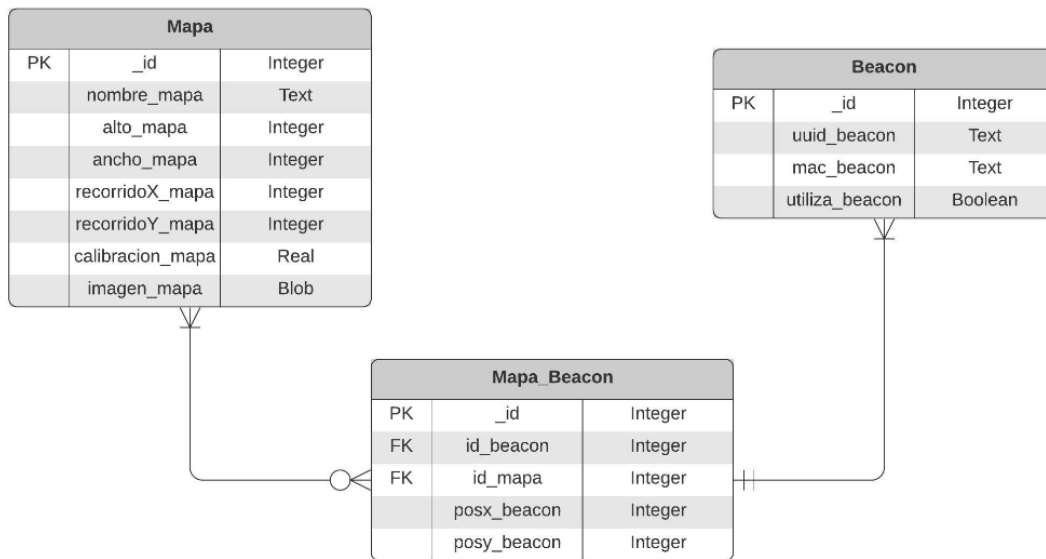


Figura 19. Diagrama ERD de la base de datos de la aplicación.

3.1.4. INTERFACES GRÁFICAS DE USUARIO

La aplicación consta de siete interfaces de usuario identificadas con el código GUI-n (Figura 20). El manual de usuario de la aplicación se puede visualizar en el ANEXO 2.

La GUI-1 es la interfaz principal, consta de seis botones que direccionan a cada apartado de la aplicación.

La GUI-2 corresponde a la pantalla “Cargar Mapa”, aquí es donde se realiza el posicionamiento, consta de un botón “Iniciar” que permite iniciar el posicionamiento interior, los campos de texto “alto” y “ancho” muestran las dimensiones del mapa que es cargado (la aplicación carga el mapa relacionado con la baliza más próxima al teléfono, al momento de entrar a esta página).

La GUI-3 corresponde a la pantalla “Crear Mapa”, aquí se realiza la grid-map del ambiente interior. Consta de cinco botones: “Medidas”, “Calibrar”, “Guardar”, “Beacons Actuales”, y “Beacons Disponibles”. El primero define las medidas (alto y ancho) del mapa interior (en metros); el segundo permite calibrar el azimut en el cual se va a graficar (Figura 14), el ícono en forma de flecha ayuda como guía a la calibración; el tercero permite guardar el mapa generado, el cuarto presenta las balizas seleccionadas para ese mapa y da la posibilidad de eliminar una por una, y el quinto muestra las balizas disponibles en la aplicación para ser agregadas a un mapa en específico. El campo de texto “Nombre Imagen” permite asignarle un nombre al mapa generado, los campos de texto “Alto” y “Ancho” permiten definir las medidas del ambiente interior. El primer spinner permite seleccionar la categoría de representación

gráfica que deseamos implementar, y el segundo spinner muestra las balizas agregadas con anterioridad.

La GUI-4 corresponde a la pantalla “Editar Mapa”, aquí se realiza la edición de un mapa guardado en la aplicación. Los botones “Calibrar” y “Actualizar” son de parecido funcionamiento a los botones “Calibrar” y “Guardar” de la GUI-3. El botón “Mostrar Mapas” permite seleccionar el mapa que se va a editar. El botón “Beacon Actuales” permite visualizar las balizas actuales del mapa, y permite quitar una baliza del mapa si fuese el caso, el botón “Beacons Disponibles” permite visualizar las balizas que pueden ser añadidas al mapa. El campo de texto “Nombre Imagen” presenta igual funcionamiento que su homónimo de la GUI-3, de igual forma los dos spinners. A diferencia de la GUI-3 aquí no es posible cambiar las medidas del ambiente interior, ya que la grid generada cambiaría de tamaño y por ende los obstáculos y puntos de interés ya no estarían alineados con el mapa, por lo cual los campos de texto “Alto” y “Ancho” únicamente muestran las dimensiones del mapa seleccionado.

La GUI-5 corresponde a la pantalla “Eliminar Mapa”, aquí se realiza la eliminación de un mapa de la aplicación. El botón “Mostrar Mapas” tiene igual funcionamiento que su homónimo de la GUI-4, el botón “Eliminar Mapa” permite eliminar el mapa seleccionado, y el campo de texto “Nombre Imagen” muestra el nombre del mapa que se va a eliminar.

La GUI-6 corresponde a la pantalla “Agregar Beacons”, aquí se agregan las balizas a la aplicación para su posterior uso. El botón “Buscar Beacons” permite iniciar el escaneo (durante 20 segundos) de balizas BLE, el botón “Mostrar Beacons” permite visualizar las balizas encontradas durante el escaneo realizado y seleccionar las balizas que serán guardadas en la aplicación, y el botón “Guardar Beacons” permite guardar las balizas seleccionadas. El campo de texto “UUIDS Beacons” muestra la lista de las balizas seleccionadas para ser guardadas.

La GUI-7 corresponde a la pantalla “Eliminar Beacons”, aquí se eliminan las balizas de la aplicación. El botón “Mostrar Beacons” y el campo de texto “UUIDS Beacons” presentan igual funcionamiento que sus homónimos de la GUI-6. El botón “Eliminar Beacons” permite eliminar las balizas seleccionadas de la aplicación.

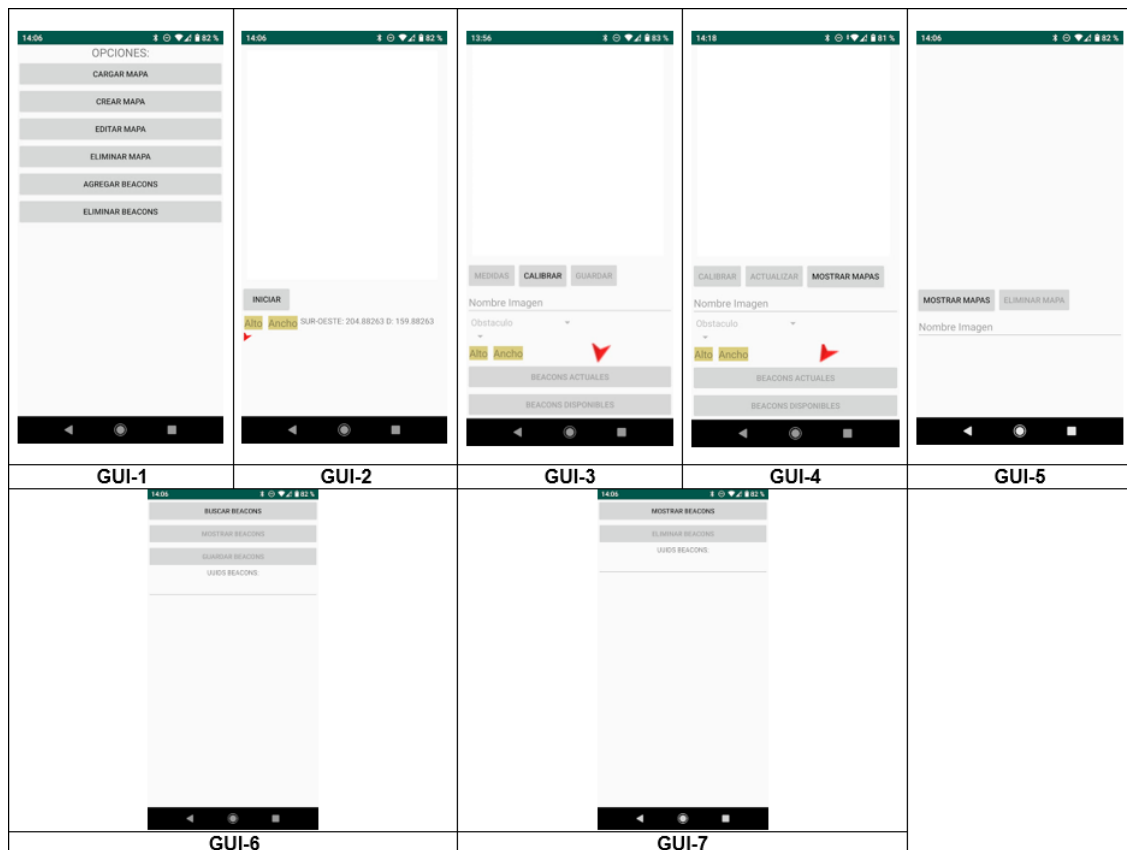


Figura 20. Interfaces de usuario de la aplicación.

3.2. REPRESENTACIÓN GRÁFICA DEL AMBIENTE INTERIOR

3.2.1. GRID-MAP (MAPA DE CUADRÍCULA)

Para determinar la dimensión del lienzo donde se realiza la grid es necesario definir, primero, la dirección de la representación gráfica (azimut), y posteriormente, definir las medidas del ambiente interior. El lienzo viene con un tamaño por defecto 1000x1200 píxeles, en base a este tamaño se decide aumentar o disminuir el tamaño del lienzo en base a las dimensiones ingresadas referentes al mapa interior. Únicamente se realiza una modificación al tamaño del lienzo si la división de celdas de la grid no llega a ser exacta con los valores iniciales del lienzo, también por este motivo cuando se ingresa las dimensiones del ambiente interior se realiza un redondeo de estos valores, estos dos inconvenientes se dan porque el lienzo únicamente trabaja con valores exactos (no decimales) de píxeles. Cabe señalar que las dimensiones del ambiente interior ingresados en metros son transformadas a centímetros para que sea una representación más relacionada con el lienzo en cuanto a escalas.

Cada celda perteneciente a la grid-map es de una dimensión específica, para el desarrollo de este trabajo de investigación se definió un tamaño de 50x50 cm, su similar en píxeles tanto en el eje x como en el eje y del lienzo dependen de la modificación de tamaño realizado al mismo. Como el lienzo no es de

dimensión $n \times n$ se puede presentar el caso de que presenten valores distintos de pixeles tanto en el eje x como en el eje y , pero en términos de representación del ambiente interior son equivalentes.

En la Figura 21 se presentan diferentes ejemplos de grid-map. En la imagen GM-1 presenta una grid-map de un ambiente interior de 8x8 m, la GM-2 presenta una grid-map de un ambiente interior de 16x16 m, la GM-3 presenta una grid-map de un ambiente interior de 8x6 m, y la GM-4 presenta una grid-map de un ambiente interior de 15x14 m. Las imágenes GM-1 y GM-3 presentan dimensiones menores a 10 m, a su vez, GM-2 y GM-4 presentan dimensiones mayores a 10 m. GM-1 y GM-3 presentan celdas de mayor tamaño que GM-2 y GM-4, pero en los cuatro casos cada celda equivale a 50x50 cm.

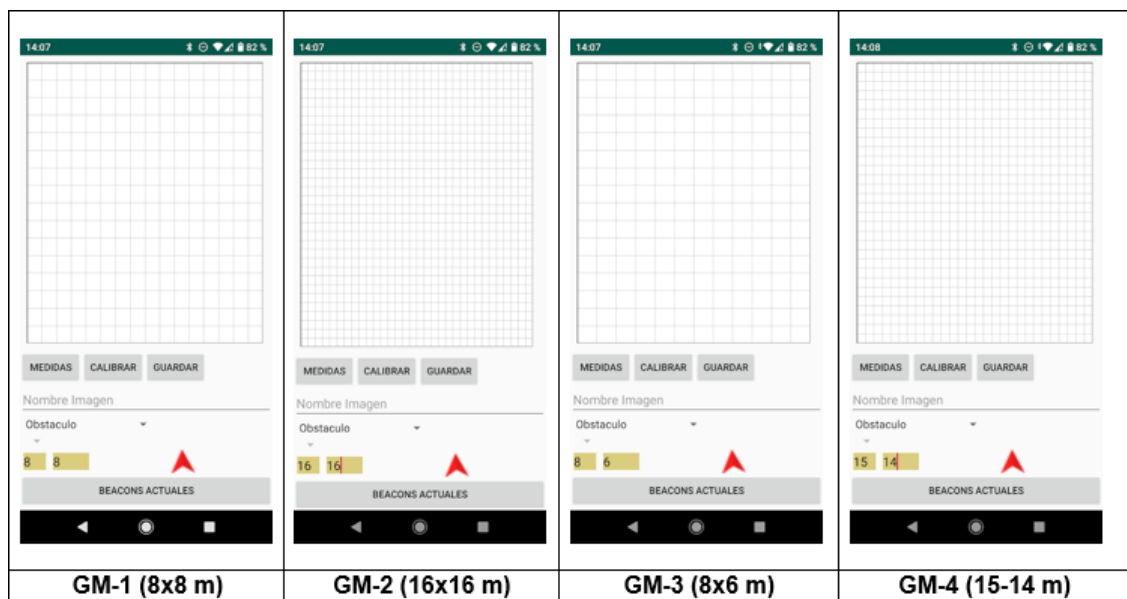


Figura 21. Diferencias entre grid-maps.

3.2.2. OBSTÁCULOS

Para la diferenciación de las celdas se procedió con la definición de colores que representen un obstáculo, punto de interés, o camino libre. Cada celda dentro de la grid posee un color que permite distinguir su categoría para un posterior análisis, en la Tabla 3 se puede observar dicha clasificación. Para poder cambiar el color de la celda únicamente basta con presionar la celda donde queremos agregar o quitar una categoría en específico.

Tabla 3. Categorías consideradas para la creación de la grid-map.

Categoría	Color	Descripción
Camino libre	Blanco	Celda libre de obstáculos o de cualquier otro elemento a considerar.
Obstáculo	Negro	Celda que posee algún obstáculo que debe ser sorteado en la navegación.
Puerta	Verde	Celda que posee una puerta, no se considera un obstáculo.
Escalera	Amarillo	Celda que posee una escalera, no se considera un obstáculo.
Elevador	Cyan	Celda que posee un elevador, no se considera un obstáculo.
Punto de interés	Magenta	Celda que posee un elemento no contemplado en las anteriores categorías, pero requiere ser mencionado al momento de la navegación. No se considera un obstáculo.

Como es difícil determinar todos los objetos de interés que pueden estar en un ambiente interior, se procedió con la definición específica de escaleras, puertas y elevadores como elementos más comunes, dejando un grupo en específico para los objetos no mencionados con anterioridad (no toma en cuenta las balizas). Las balizas son representadas en el mapa mediante un punto azul en la esquina superior izquierda de la celda donde se encuentra la baliza, igualmente que la categorización, las balizas se ingresan a la pantalla presionando el punto en específico donde se encuentra y ese punto en específico es el que se guarda para el posterior posicionamiento interior.

En la Figura 22 se muestra un ejemplo de un ambiente interior de 6x6 m en el cual se muestran de manera más clara todas las categorías mencionadas con anterioridad.



Figura 22. Ejemplo de un ambiente interior.

(Abu Doush et al., 2016) definieron al momento de la creación del mapa interior únicamente tres tipos de colores asociados a una “categoría” en

específico: negro (camino libre), gris (cercano a un obstáculo), y blanco (obstáculo). Con las categorías implementadas en la aplicación desarrollada, se puede tener referencia de algunos puntos de interés más comunes como puertas o escalera, y una categoría más general donde puedan ser representados los puntos de interés menos comunes. Si bien (Abu Doush et al., 2016) manejan una situación “cerca de un obstáculo”, en la aplicación desarrollada se definió un tamaño de celda mayor, lo que permite contemplar su posición real (obstáculos, puntos de interés y usuario) en el ambiente interior y un margen a su alrededor.

3.3. ALGORÍTMO DE RUTA

3.3.1. APLICACIÓN DEL ALGORITMO A*

El movimiento en el mapa consta de 8 direcciones posibles (arriba, abajo, izquierda, derecha, diagonales), además, el valor $g(n)$ (Figura 8) entre un nodo principal y un nodo hijo es determinado por la distancia euclidiana. La definición del ángulo azimut de calibración (Figura 14) permite definir la celda o nodo de frente, esta es añadida en el eje de las y , a partir de este nodo se van definiendo los otros nodos posibles. Para el desarrollo del trabajo de investigación se implementó un análisis adicional al momento de evaluar que vecino es válido para transitar. Específicamente, consiste en que un nodo diagonal es válido para transitar, si y solo si, el nodo diagonal es válido para transitar (no es un obstáculo) y si los nodos próximos al nodo diagonal, también evaluados en un principio, son nodos válidos. Por ejemplo (Figura 23), en la situación A, X es nuestra ubicación actual, sin el análisis adicional, nuestros vecinos válidos son: 1, 2, 3, 4, 5, 6. La situación B, igualmente, X es nuestra ubicación actual, con el análisis adicional, nuestros nodos válidos en este caso serían: 3, 4, 5. Si bien, en tema de programación, realmente la situación A no presentaría ningún problema, pero en la realidad un movimiento, por ejemplo, del nodo X al nodo 1 no sería muy práctico y dependiendo de la situación puede que no sea factible, igualmente, un movimiento desde el nodo X al nodo 2 o al nodo 6 presentarían este inconveniente.

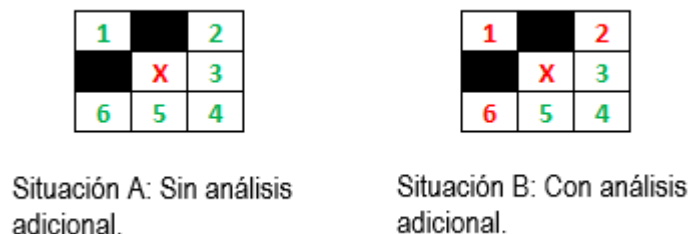


Figura 23. Análisis extra empleado al algoritmo A*.

En la Figura 24 se puede apreciar un movimiento a través de la ruta de ejemplo generada por el algoritmo A*.

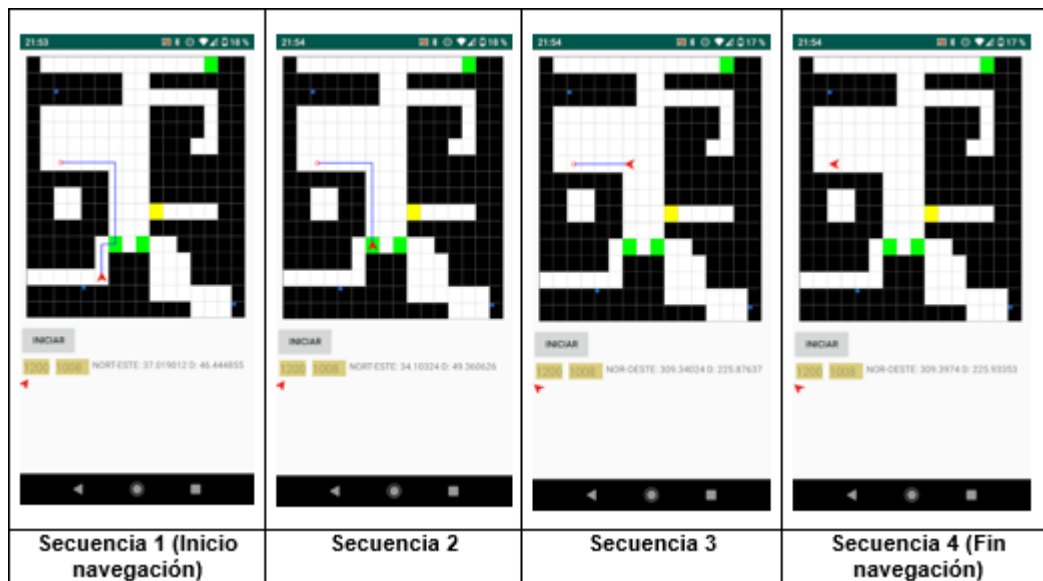


Figura 24. Secuencia de navegación en una ruta de ejemplo.

La generación de los nodos para que puedan ser evaluados por el algoritmo A* presenta un buen funcionamiento, el camino es generado por celdas que son válidas para la navegación. En comparación con el enfoque propuesto por (Eraghi et al., 2015) de realizar un grafo tomando como centro la posición actual sin evaluar obstáculos, en el algoritmo A* implementado en la aplicación se realiza la evaluación de todos los nodos vecinos a nuestra ubicación, además, con el análisis extra empleado se permite tener un grafo parecido al que se encuentra en la Figura 11. De igual manera, la heurística Manhattan presenta un buen funcionamiento, en comparación con el trabajo realizado por (Qian et al., 2019) donde se realiza un movimiento en 4 direcciones añadiendo, también, más información de heurística. La aplicación desarrollada maneja un enfoque de ocho direcciones dejando un peso determinado por la distancia euclidiana entre dos nodos.

De igual manera que la comparación hecha por (Chen et al., 2017) (entre los algoritmos Dijkstra, A* y Ant) donde el algoritmo A* tiene un mejor funcionamiento sin realizar alguna mejora, y el correcto funcionamiento del algoritmo A* en la investigación de (Abdullah & Korial, 2015), La generación de la ruta en pantalla es rápida y correcta, por lo cual el funcionamiento del algoritmo A* en la grid-map generada en la aplicación es bueno.

3.4. ORIENTACIÓN DEL USUARIO

3.4.1. UTILIZACIÓN DE SENSORES DEL TELÉFONO

El valor del umbral utilizado al momento de la aplicación del filtro de paso bajo se estableció en 0.75. Primeramente, en base al ángulo azimut calibrado (Figura 14), establecemos el ángulo en el cual se encuentra el siguiente nodo de la ruta generada en base a nuestra posición actual, como el ángulo calibrado se asigna al nodo superior en el eje de las y , el ángulo calibrado

equivale a 0° y en base a este se puede asignar el ángulo en el que se encuentra el siguiente nodo, siempre se hace este análisis en sentido horario, esto permite obtener la distancia entre el ángulo actual (dirección del usuario) y el ángulo del siguiente nodo, y la dirección de giro correspondiente (izquierda o derecha). La Figura 25 muestra un extracto de código que permite definir el ángulo de un nodo próximo.

```
private float ObtenerAnguloNodoHijo(Nodo nodoPadre, Nodo nodoHijo, float anguloGrid) {
    //Determina el ángulo en el cual se encuentra el siguiente nodo
    //Nodo de frente == Norte 0°
    float giro = 0f;
    int xPadre, yPadre, xHijo, yHijo;
    xPadre = nodoPadre.getCoordenadaX();
    yPadre = nodoPadre.getCoordenadaY();
    xHijo = nodoHijo.getCoordenadaX();
    yHijo = nodoHijo.getCoordenadaY();
    if (xPadre == xHijo && yPadre > yHijo) giro = 0; //Frente
    if (xPadre < xHijo && yPadre > yHijo) giro = 45; //Diagonal Superior Derecha
    if (xPadre < xHijo && yPadre == yHijo) giro = 90; //Derecha
    if (xPadre < xHijo && yPadre < yHijo) giro = 135; //Diagonal Inferior Derecha
    if (xPadre == xHijo && yPadre < yHijo) giro = 180; //Atras
    if (xPadre > xHijo && yPadre < yHijo) giro = 225; //Diagonal Inferior Izquierda
    if (xPadre > xHijo && yPadre == yHijo) giro = 270; //Izquierda
    if (xPadre > xHijo && yPadre > yHijo) giro = 315; //Diagonal Superior Izquierda
    giro += anguloGrid;
    if (giro > 360) giro -= 360;
    return giro;
}
```

Figura 25. Método de obtención del ángulo de un nodo próximo en una ruta generada.

En la navegación es muy difícil acertar específicamente un ángulo de dirección por lo cual se definieron diferentes rangos en los cuales se da por aceptado un ángulo en específico, dichos rangos se los puede visualizar en la Tabla 4, los rangos están considerados en un rango de ángulos de 0° a 180°, donde α es la distancia entre dos ángulos. Los tipos de giro son utilizados posteriormente para la emisión de indicaciones sonoras.

Tabla 4. Rangos para la definición del tipo de giro.

Condición	Tipo de giro
$\alpha \leq 10^\circ$	No hay giro
$10^\circ < \alpha < 85^\circ$	Giro leve
$85^\circ \leq \alpha \leq 95^\circ$	Giro recto (90° aproximadamente)
$95^\circ < \alpha < 170^\circ$	Giro fuerte
$170^\circ \leq \alpha \leq 180^\circ$	Medio giro (180° aproximadamente)

En comparación con el trabajo realizado por (Asmara et al., 2017) donde los sensores presentan un buen funcionamiento, aunque se pueden presentar inferencias en las lecturas debido a factores externos, en la aplicación desarrollada la utilización del filtro de paso mejora la oscilación presente en el direccionamiento evitando, en mayor medida, movimientos bruscos muy repetitivos, y lograr mantener en mayor medida, una relación invariable entre la dirección del usuario y el ángulo azimut de calibración, aunque, el movimiento de la mano cuando se sostiene el teléfono también puede causar intermitencias en los datos obtenidos por los sensores del teléfono.

La relación de dirección del usuario y el mapa interior también presento buenos resultados, si bien, en algunas direcciones puede que no presente una referencia exacta, pero con el rango implementado permite abarcar una parte

de este inconveniente. La emisión y recepción de los datos de los sensores (magnetómetro y acelerómetro) es constante, lo que permite que los movimientos realizados por el teléfono sean replicados instantáneamente en el mapa de la aplicación. Cabe señalar que durante el desarrollo se presentaron situaciones aisladas donde el ícono de representación del usuario tenga cambios muy bruscos o no parecidos al movimiento real realizado, esto presumiblemente por alguna interferencia en la señal obtenida por los sensores o una falta de calibración, y cuya repetición fue difícil de lograr. Además, para un mejor funcionamiento, se procedió con la calibración de la brújula gracias a la aplicación de Google Maps. En la Figura 26 se presentan la posición estándar de un usuario para que su orientación pueda ser reconocida, además, se presenta el resultado de la calibración mediante Google Maps.



Figura 26. Orientación de un usuario.

3.5. UTILIZACIÓN DE TECNOLOGÍA BLE

3.5.1. ADMINISTRACIÓN DE BALIZAS BLE

El manejo de la tecnología BLE en la aplicación se realizó a través de la API nativa de Android para BLE y el uso de la librería Android Beacon Library. La combinación de los dos consiste en utilizar la API de Android para BLE para el escaneo de dispositivos (ya sea para su descubrimiento, o en el posicionamiento interior) y la librería Android Beacon Library para manejar la información capturada (paquetes de anuncio) y poder identificar las balizas a través de su UUID. En la Figura 27 se puede apreciar un fragmento de código de la combinación de los dos apartados, al momento de la detección de una baliza BLE en la aplicación.


```

@Override
public void onScanResult(int callbackType, ScanResult result) {
    super.onScanResult(callbackType, result);
    for (BeaconParser parametro : beaconParserList) {
        Beacon beacon = parametro.fromScanData(result.getScanRecord().getBytes(), result.getRssi(), result.getDevice(), timestampMs: 10);
        if (beacon != null) {
            if (!dispositivosBluetooth.containsKey(beacon.getId1().toString())) {
                dispositivosBluetooth.put(beacon.getId1().toString(), result.getDevice());
                System.out.println("UUID: " + beacon.getId1().toString());
                System.out.println("MAC: " + result.getDevice().getAddress());
                String msg = "Datos Enviados = ";
                for (byte b : result.getScanRecord().getBytes())
                    msg += String.format("%02x ", b);
                System.out.println(msg);
                break;
            }
        }
    }
}

```

Figura 27. Función onScanResult para la detección de una baliza BLE.

Si bien, mediante el uso de onScanResult() de la API de Android, ya es posible obtener la dirección MAC del dispositivo y su RSSI, pero para realizar el posicionamiento interior, es necesario primeramente reconocer las balizas pertenecientes a un mapa interior en cuestión, pero los dispositivos BLE simulados cuentan con direcciones MAC dinámicas por lo cual no podían ser reconocidos por este medio, y segundo, también es necesario obtener el valor TxPower de la baliza encontrada.

Para la solución de estos inconvenientes se utiliza la librería Android Beacon Library. Esta librería permite transformar la información obtenida por onScanResult(), en un objeto Beacon, gracias a la utilización de las clases Beacon y BeaconParser, permitiendo capturar de una manera más sencilla tanto el UUID como el TxPower de la baliza encontrada. Esto permite tener los datos necesarios ya sea para que una baliza sea guardada en la aplicación para un posterior uso, o para su reconocimiento en el posicionamiento interior. Cabe señalar que es necesario definir encabezados para poder encontrar una baliza en específico, estos encabezados presentan una estructura propia de la librería, aunque para balizas AltBeacon y EddyStone ya se encuentran añadidos estos encabezados, para la detección de balizas iBeacon se añade un elemento más, para su posterior detección. En la Figura 28 se presentan los diferentes encabezados utilizados en la aplicación.

```

private String[] beaconsLayout = {"m:2-3=0215,i:4-19,i:20-21,i:22-23,p:24-24", //Encabezado iBeacon
    BeaconParser.EDDYSTONE_URL_LAYOUT,
    BeaconParser.EDDYSTONE_UID_LAYOUT,
    BeaconParser.EDDYSTONE_TLM_LAYOUT,
    BeaconParser.ALTBEACON_LAYOUT,
    BeaconParser.URI_BEACON_LAYOUT};

```

Figura 28. Diferentes encabezados para la detección de balizas BLE.

La Figura 29 muestra la detección de dos balizas en un escaneo de duración de 20 segundos (tiempo asignado para la detección de balizas y su posterior almacenamiento en el sistema), la primera baliza es de tipo AltBeacon (el dispositivo en el que se simuló no tiene soporte BLE) y la segunda baliza es de tipo iBeacon, además, se puede observar un pequeño fragmento de los datos enviados por las balizas al momento de su anuncio. Para la

realización de la aplicación, el uso de la librería Android Beacon Library y la API de Android para BLE permitió obtener todos los datos necesarios de las balizas para su administración y posterior uso.

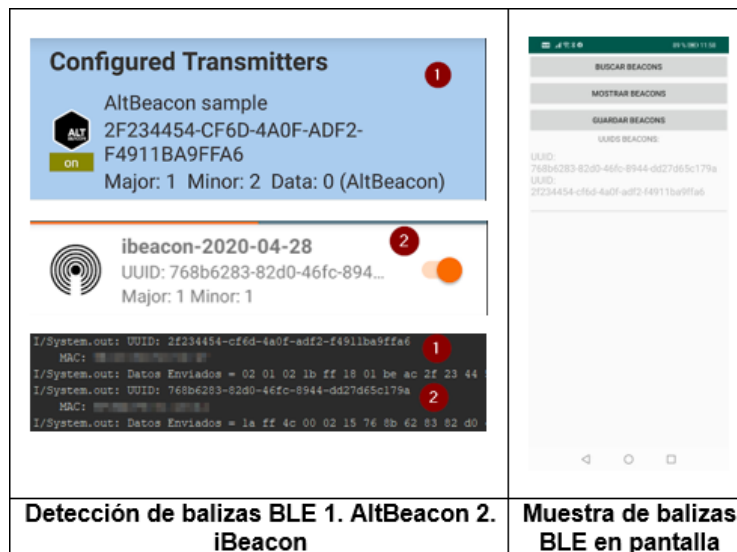


Figura 29. Detección de balizas BLE.

3.6. INDICACIONES SONORAS

3.6.1. DEFINICIÓN Y UTILIZACIÓN DE INDICACIONES SONORAS

Comúnmente se evalúan dos tipos de indicaciones sonoras: indicaciones por voz e indicaciones por tonos o pitidos. Las instrucciones por voz son instrucciones que mencionan direcciones de giro, ya sea giros rectos, o mencionando el ángulo específico que se requiere girar, además, informando cuánto se debe continuar si se sigue una ruta recta. Las instrucciones por tonos o pitidos presentan una mejor eficiencia que las indicaciones por voz, se han utilizado tonos de 440Hz de frecuencia, y por lo regular son utilizados para la realización de un giro en específico.

En cuanto respecta a técnicas o herramientas utilizadas, se observa que, la utilización de sonido estéreo puede ser una técnica de gran ayuda para la emisión de indicaciones sonoras, a su vez, para las indicaciones por voz, la utilización de herramientas tipo TTS puede ser de gran ayuda para este apartado en específico.

Para la realización del trabajo de investigación, en base a lo señalado con anterioridad, se definieron las siguientes agrupaciones de indicaciones sonoras: giros rectos, giros medios, avance e indicación de punto de interés. Estas agrupaciones son explicadas a continuación.

Los giros rectos, son indicaciones que se mencionan cuando es necesario realizar un giro mediante una indicación por voz, pero dicho giro es determinado por 90° o 180° aproximadamente. Estas indicaciones son mencionadas por TTS de Android, además dependiendo de la dirección de

giro, se utiliza un sonido estéreo. En la Tabla 5 se mencionan las indicaciones de esta agrupación.

Tabla 5. Indicaciones sonoras pertenecientes al grupo “giros rectos”.

Indicación	Situación
“Gire Izquierda”	Realización de un giro de 90° por la izquierda, la indicación se emite por el audífono izquierdo.
“Gire Derecha”	Realización de un giro de 90° por la derecha, la indicación se emite por el audífono derecho.
“Medio Giro por Izquierda”	Realización de un giro de 180° por la izquierda, la indicación se emite por el audífono izquierdo.
“Medio Giro por Derecha”	Realización de un giro de 180° por la derecha, la indicación se emite por el audífono derecho.

Los giros medios, son indicaciones que se mencionan cuando es necesario realizar un giro mediante la reproducción de una pista musical (arpeggio de piano). En un principio se consideró la utilización de un tono en una frecuencia de 440Hz, pero resultaba molesto su constante emisión, por lo cual se decidió utilizar un arpeggio de piano para la experimentación. Los giros considerados en esta agrupación van de $0^\circ < \alpha < 90^\circ$ y de $90^\circ < \alpha < 180^\circ$ aproximadamente. Dependiendo de la dirección del giro, se utiliza sonido estéreo para la emisión de la indicación. En la Tabla 6 se mencionan las indicaciones pertenecientes a esta agrupación.

Tabla 6. Indicaciones sonoras pertenecientes al grupo “giros medios”.

Indicación	Situación
Arpeggio de piano (Pista A) (Giro por izquierda)	Realización de un giro de $0^\circ < \alpha < 90^\circ$ o de $90^\circ < \alpha < 180^\circ$ dependiendo de la situación, por la izquierda, esta pista únicamente se emite cuando se requiere realizar un giro por la izquierda, la indicación se emite por el audífono izquierdo.
Arpeggio de piano (Pista B) (Giro por derecha)	Realización de un giro de $0^\circ < \alpha < 90^\circ$ o de $90^\circ < \alpha < 180^\circ$ dependiendo de la situación, por la derecha, esta pista únicamente se emite cuando se requiere realizar un giro por la derecha, la indicación se emite por el audífono derecho.

El grupo avance e indicación de punto de interés hace referencia a las indicaciones que se emiten mediante TTS de Android, cuando se requiere un avance recto e informando, si fuese el caso, de algún punto de interés que no es considerado un obstáculo como tal (Tabla 3). En un principio se consideró utilizar como métrica de distancia “x” metros, pero como cada celda de la grid-map es de un tamaño de 0.5 metros podría ser confuso la utilización de la

métrica señalada, por lo cual se la sustituyo con la frase “un paso”. La Tabla 7 muestra las indicaciones pertenecientes a esta agrupación.

Tabla 7. Indicaciones sonoras pertenecientes al grupo “avance e indicación de punto de interés”

Indicación	Situación
“Avance un paso”	Realización de un movimiento en línea recta (avance de un paso), la indicación es emitida tanto por el audífono izquierdo y derecho.
“Cuidado, tienes un (una) X a un paso”. Ejemplo X=Puerta	Realización de un movimiento en línea recta (avance de un paso) informando que se tiene un punto de interés definido, la indicación es emitida tanto por el audífono izquierdo y derecho.
“Cuidado, punto de interés a un paso”	Realización de un movimiento en línea recta (avance de un paso) informando que se tiene un punto de interés no definido, la indicación es emitida tanto por el audífono izquierdo y derecho.
“Ha llegado a su destino”	Fin de la navegación.

Las indicaciones son reproducidas cada cinco segundos. La velocidad de lectura del TTS de Android se estableció en 1.5. Cuando se produce un cambio de celda (posición) en el mapa, se reproduce una nueva indicación, en el caso de que se encontrara otra indicación en reproducción automáticamente la detiene y reproduce la nueva. Las indicaciones pertenecientes al grupo de “giros medios” se reproducen constantemente hasta que el usuario llegue a la dirección deseada.

En las pruebas realizadas mediante una simulación de posicionamiento ideal, Se establecieron tres distintos puntos de inicio, que terminan en un mismo punto final (Figura 32). Las indicaciones sonoras presentaron un comportamiento variado. Las indicaciones pertenecientes a “giros medios” presentan una reproducción rápida ofreciendo la información del giro correspondiente, aunque en algunas ocasiones se pueden ver opacadas por las indicaciones del grupo “giros medios”. Las indicaciones pertenecientes a “giros medios” presentan una buena interacción al momento de indicar un giro, también pueden ser de ayuda para la corrección de dirección, sin que esta implique un giro muy anunciado, pero esto puede ser contraproducente ya que la reproducción del sonido puede llegar a ser muy corta. Las indicaciones de “avance e indicación de punto de interés” se reproducen de forma constante, además, pueden ser las que más veces sean anunciadas en la navegación, especialmente la indicación “avance un paso”, las indicaciones que informan sobre algún punto de interés especificado o no, pueden ser de gran ayuda al momento de realizar una navegación.

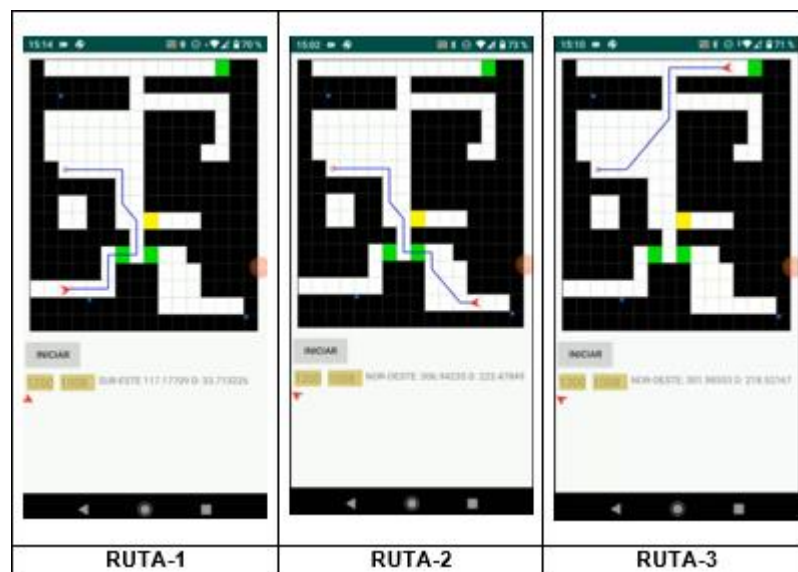


Figura 30. Rutas realizadas para la prueba de las indicaciones sonoras.

En cuanto respecta a las configuración o técnicas empleadas para la emisión de las indicaciones sonoras. La utilización del TTS funcionó de manera correcta permitiendo generar las indicaciones requeridas, la velocidad de lectura también presento un buen resultado permitiendo que las frases sean reproducidas a una velocidad medianamente alta pero aun siendo entendibles.

La utilización del sonido estéreo permite una mayor facilidad para distinguir las indicaciones sonoras, además, resulta muy intuitivo al momento del direccionamiento, pero para lograr distinguir el sonido estéreo es necesario la utilización de audífonos lo que causa una disminución del sonido ambiente que se presente en el lugar donde se encuentre el usuario.

La definición de grupos para dividir las diferentes indicaciones sonoras, en base a las agrupaciones realizadas por (Kacorri et al., 2018) para realizar un mejor análisis de rutas, ayudo a que la navegación sea comprendida de una mejor manera, permitiendo abarcar la mayoría de situaciones que puedan presentarse.

Si bien las indicaciones sonoras de voz emitidas con la utilización de TTS utilizadas en los trabajos de: (Cecílio et al., 2015), (Meliones & Sampson, 2018), (Sato et al., 2017), (Oh et al., 2017) presentan buenos resultados. En la aplicación desarrollada únicamente las indicaciones de aviso de algún punto de interés presentan un resultado aceptable, la indicación de “avance un paso” tiende a ser repetitiva, y las indicaciones “giros rectos” casi no fueron mencionadas durante la navegación. Además, como apartado complementario, el uso de TTS de Android fue de gran ayuda para poder generar las indicaciones de voz en la aplicación desarrollada.

Tomando como base los trabajos de: (Bai et al., 2017), (Cheraghi et al., 2017), (Guerrón Paredes et al., 2018) donde utilizan, entre otros tipos de indicaciones, indicaciones por pitidos, y las cuales presentan buenos resultados. En la aplicación desarrollada, las indicaciones “giros medios” son más intuitivas al momento de su seguimiento, por lo cual presentan un mejor funcionamiento que las otras dos categorías presentes en la aplicación.

(Bai et al., 2017), (Guerrón Paredes et al., 2018) resaltan el uso de pitidos como una indicación sonora eficiente en comparación con indicaciones sonoras de voz, a su vez, utilizando en las indicaciones por pitidos, sonido estéreo (Bai et al., 2017) y sonido 3D (Guerrón Paredes et al., 2018). En la aplicación desarrollada la categoría “giros medios” que toma como base el enfoque de indicaciones por pitidos, aunque con la utilización de arpeggios de piano, con la utilización de sonido estéreo es la que mejor resultado presento.

3.7. POSICIONAMIENTO INTERIOR

3.7.1. CALIBRACIÓN DE BALIZAS

Una de las cosas primordiales para poder realizar un posicionamiento interior, consiste en la calibración de las balizas, más específicamente, el valor de TxPower de la baliza, que consiste en el valor RSSI entre la baliza y un dispositivo central a una distancia determinada (1 metro), esto se realiza ya que el valor TxPower se utiliza para poder calcular la distancia entre la baliza y el dispositivo central y posteriormente utilizar ese valor en la trilateración. La Figura 31 muestra un ejemplo de cómo se calibraron las balizas para el posicionamiento interior.



Figura 31. Calibración de una baliza a un metro de distancia.

Con la ayuda de la aplicación Locate, se pudo hacer la calibración de los tres dispositivos simuladores de balizas BLE (Tabla 8), lastimosamente la baliza Beacon-1 no pudo ser calibrada debido a que la aplicación que funciona de simulador de baliza no permite realizar cambios, por lo cual utiliza un valor por defecto.

Tabla 8. Balizas utilizadas durante el posicionamiento interior.

Nombre Baliza	UUID	TxPower
Beacon-0 (Xiaomi Mi A2)	76b6283-82d0-46fc-8944-dd27d65c179a	-54
Beacon-1 (Laptop HP)	20cae8a0-a9cf-11e3-a5e2-0800200c9a66	-59
Beacon-2 (iPhone 11 Pro)	4b9c3ee7-bec5-4649-b0e0-91fcdabd23797	-52

3.7.2. DISTANCIA ENTRE BALIZA Y TELÉFONO

La distancia se obtiene mediante la lectura del RSSI que emite la baliza, entre un valor más cercano a 0 significa que estamos más cerca de la baliza en cuestión. Pero el valor RSSI tiende a ser inconstante por lo que el umbral del filtro de paso bajo se estableció en 0.75. El valor del exponente de pérdida de ruta se estableció con un valor de 4. La Figura 32 muestra un fragmento de código que permite obtener la distancia anteriormente descrita.

```
private double Distancia(double rssi, double txPower) {
    double distancia;
    distancia = Math.pow(10, (-rssi + txPower) / (10 * constanteNDistancia));
    return distancia;
}
```

Figura 32. Código para calcular la distancia entre una baliza y un dispositivo central.

Con el valor del RSSI filtrado, el valor TxPower de la baliza en cuestión, y la definición del exponente de pérdida de ruta, ya es posible realizar el cálculo de la distancia de una baliza. La Figura 33 muestra un ejemplo de los datos que se utilizan para el cálculo de la trilateración mediante el método del centroide.

```
I/System.out: UUID: 0 : 768b6283-82d0-46fc-8944-dd27d65c179a
I/System.out: UUID: 1 : 20cae8a0-a9cf-11e3-a5e2-0800200c9a66
                UUID: 2 : 4b9c3ee7-bec5-4649-b0e0-91fcdabd23797
I/System.out: POS 0: 1.3095238095238095 - 1.44
                POS 1: 2.5 - 7.453333333333333
                POS 2: 7.777777777777778 - 7.866666666666666
                R0: -75.0 R1: -46.0 R2: -73.0
I/System.out: T0: -54.0 T1: -59.0 T2: -52.0
                N: 4.0
```

Figura 33. Datos para el cálculo de la trilateración mediante el método del centroide.

3.7.3. APLICACIÓN DE LA TRILATERACIÓN

La actualización de la posición se realiza en cada lectura de unas de las balizas que se encuentran enlazadas al mapa interior, esto permite una constante actualización, aunque es propensa a errores. Para la aplicación del método del centroide para la resolución de la trilateración, es necesario obtener las posiciones de las balizas (son guardadas cuando se crea el mapa interior), el RSSI ya filtrado y el valor TxPower de la baliza. Ya con todos los valores, como se muestra en el código de la Figura 34, es posible realizar el cálculo mediante el método del centroide.

```

for (int i = 0; i < posiciones.size(); i++) {
    posicionTemp = posiciones.get(i); //posicion beacon
    rssiTemp = rssi[i]; //rssi del beacon
    txpowerTemp = txpower[i]; //txPower del beacon
    distancia = Distancia(rssiTemp, txpowerTemp); //distancia

    //Numeradores
    tempNumeradorX = (posicionTemp[0]) / (distancia); //X
    tempNumeradorY = (posicionTemp[1]) / (distancia); //Y

    numeradorX += tempNumeradorX; //X Sumatoria
    numeradorY += tempNumeradorY; //Y Sumatoria

    //Denominadores
    denominador += (1 / (distancia)); //Sumatoria (IGUAL PARA X y Y)
}

ubicacionDeterminada[0] = numeradorX / denominador; //Posicion estimada X
ubicacionDeterminada[1] = numeradorY / denominador; //Posicion estimada Y

return ubicacionDeterminada;

```

Figura 34. Código para calcular la trilateración mediante el método del centroide.

Se procedieron con tres distintas pruebas, todas siguiendo el mismo recorrido y en el mismo mapa interior. En la primera prueba realizada con las balizas descritas en la Tabla 8, se procedió con dejar todas las posiciones resultado del método del centroide, para así detectar la ruta realizada (Figura 35), si bien en algunos puntos se puede notar que realmente siguió el camino recorrido en la realidad, pero en otros casos marca puntos o zonas que no fueron transitadas durante la prueba.



Figura 35. Posicionamiento interior priorizando el seguimiento de ruta.

En una segunda prueba realizada se procedía con la actualización de la posición en el mapa, además, ya se implementaba el ícono de referencia al usuario en el mapa (Figura 36). Si bien presento un comportamiento parecido al anterior el seguimiento se pudo apreciar de una mejor manera, sin embargo, la posición era muy cambiante en plazos cortos de tiempo.



Figura 36. Posicionamiento interior con actualización de posición.

En la tercera prueba realizada se ejecutó de igual manera que la segunda prueba, pero esta vez se procedió con centrar la posición del usuario a una celda en la cual se marcaba su posición según el método del centroide (Figura 37), esto para evitar las constantes actualizaciones de posición. De igual manera que en las anteriores pruebas el seguimiento en algunos sectores era correcto, pero en otras ocasiones no se marcaba la posición correcta.

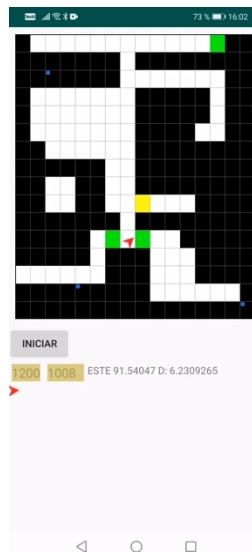


Figura 37. Posicionamiento interior con actualización de posición fijada a su centro más próximo.

En la misma línea del trabajo realizado por (Cabrera-Goyes & Ordóñez-Camacho, 2018) donde el RSSI es un factor determinante en el posicionamiento. En la aplicación desarrollada, el posicionamiento interior presenta un mejor resultado cuando la posición del usuario está más cerca de una baliza (valores RSSI más exactos), los valores de posición tienden a variar en menor rango.

La precisión de la posición es un tema que es objeto de discusión, (Nguyen et al., 2017) obtienen un rango de precisión entre 0.2 a 0.35 m con cuatro balizas BLE en un espacio de 5x5 m donde se presentaban obstáculos pero los dispositivos se podían ver frente a frente, (Mussina & Aubakirov, 2018) obtienen un rango de precisión entre 1 y 1.5 m con siete balizas BLE en un espacio de 5.40x8 m sin obstáculos, (Röbesaat et al., 2017) obtienen un rango de precisión menor a un metro con ocho balizas BLE en un espacio de 15x3.6 m, (Subedi et al., 2016) obtiene errores entre 1.58 y 2.45 m con catorce balizas BLE en un corredor de 2.5 de ancho, (W. Wang & Wang, 2015) en un ambiente simulado, obtienen un error máximo de 3.227 m, con tres puntos de referencia, utilizando el método del centroide en un ambiente de 20 x 10 m. En la aplicación desarrollada con tres balizas simuladas en distintos dispositivos (laptop, teléfono Android, teléfono iOS) en un espacio interior de 8x8 m con obstáculos, mediante la observación especialmente en la primera y segunda prueba, tomando en cuenta que cada celda perteneciente a la grid-map tiene un tamaño de 50x50 cm, la posición suele variar entre 2 y 3 celdas con respecto a la posición real del dispositivo central (usuario), aunque en algunos casos se podía observar que la posición marcada en pantalla iba más allá del número de celdas marcadas con anterioridad. Cabe señalar que la señalización de la posición en pantalla no era constante y solía variar con bastante frecuencia, por lo que se implementó lo señalado en la tercera prueba, si bien permitía unos cambios menos bruscos de posición, aun presentaba este inconveniente.

4. CONCLUSIONES Y RECOMENDACIONES

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

En la aplicación desarrollada se manejaron dos distintos enfoques de indicaciones sonoras orientadas a personas invidentes, estos enfoques fueron indicaciones basadas en voz (mediante la utilización TTS de Android), e indicaciones basadas en pitidos (mediante la utilización de arpegios de piano por motivos de experimentación), ambos reproducidos mediante el uso de un sonido estéreo. Estos dos enfoques presentan ventajas y desventajas, el primer enfoque puede ser muy informativo y descriptivo del entorno por el cual se navega, pero su emisión tiende a ser repetitiva, en situaciones que no son fundamentales de mencionar. El segundo enfoque es más intuitivo para el direccionamiento y corrección de dirección, pero no puede ofrecer información sobre puntos de interés que se encuentren en el camino. En base a estos dos enfoques, el que mejor resultado presento es el segundo enfoque, el cual permite una mejor navegación. El sonido estéreo permite una mejor interacción con las indicaciones sonoras, pero para detectar su emisión es necesario la utilización de audífonos, lo cual disminuye el sonido exterior que pueda existir. La interfaz gráfica desarrollada para la aplicación facilito la administración de mapas, y la administración de balizas. Además, permitió la realización de las pruebas de los diferentes apartados desarrollados en la aplicación, permitiendo mostrar en pantalla tanto, el mapa, la ruta, como el posicionamiento y dirección del usuario.

La definición de un mapa que pueda representar el ambiente interior se lo desarrolló a través de una grid-map que presenta diferentes categorías, con la utilización de colores, que permitan diferenciar sus celdas, y el cual presenta de manera correcta el ambiente interior pero no llega a ser preciso. El camino por el cual se va a transitar, evitando los obstáculos del ambiente interior, se realizó mediante la utilización del algoritmo A*, el cual presenta un buen funcionamiento. La orientación del usuario se detecta a través del magnetómetro con el acelerómetro, si bien presentan un funcionamiento correcto, sus resultados, en algunos casos, son influenciados por factores externos a la aplicación. El posicionamiento interior se realizó mediante la trilateración por el método del centroide, el cual brinda una posición estimada, pero no exacta ni constante. Todos estos apartados, en base a la consulta bibliográfica y sin tomar en cuenta las indicaciones sonoras para personas invidentes, fueron los considerados para el desarrollo de la aplicación.

La administración de balizas en la aplicación, mediante la utilización de la librería Android Beacon Library, permitió su detección, distinción y almacenamiento en la aplicación, por lo que su utilización puede ser muy utilitaria, ya que la información obtenida mediante la API nativa de BLE de Android entrega datos, en algunas ocasiones, sin procesar, además, que no

hay un estándar del protocolo beacon. Las balizas BLE utilizadas en la aplicación, mediante simulación en distintos dispositivos, presentaron un comportamiento variado, la calibración de su valor TxPower es necesario realizarlo en el ambiente interior donde van a ser utilizadas ya que este suele variar incluso en las mismas condiciones, además, se detectó un mejor posicionamiento, cuando se encontraba, el usuario, cerca de una baliza, por lo que el número de balizas puede ser determinante para obtener una mejor precisión y consistencia en la ubicación interior, aunque esto dependerá del ambiente interior.

4.2. RECOMENDACIONES

La realización de pruebas con personas invidentes sobre qué indicaciones sonoras son percibidas de mejor manera resulta sustancial para tener una idea más clara de su funcionamiento. La evaluación de distintos sonidos que sean percibidos de mejor manera por personas invidentes es un tema que debe ser tratado en el caso de que se implemente un enfoque auditivo como indicaciones para la navegación, aunque sin descartar indicaciones que no necesariamente sean auditivas. La utilización de audífonos, para el sonido estéreo es fundamental, por lo cual, una evaluación de estos permitiría determinar si su utilización puede ser factible o no. Para el manejo de una aplicación móvil destinada a personas invidentes requiere el análisis de distintas formas de interacción humano-máquina que eviten o minimicen el uso de la pantalla táctil de un teléfono móvil.

Si bien los distintos apartados contemplados al momento del desarrollo de la aplicación ayudaron a su funcionamiento, para la representación de un ambiente interior se puede evaluar el uso de APIs de diseño de mapas interior que permitan una representación más fidedigna del mismo; a su vez, los obstáculos que estos presentan pueden ser detectados mediante dispositivos o sensores adicionales destinados a este fin, lo cual, añadiría un enfoque más dinámico a este apartado. Si bien la navegación punto a punto es un tema para tratar, el hecho de una navegación libre, sin destino fijo, deberá ser evaluada para que la navegación en un ambiente interior sea mucho más completa. El reconocimiento de la orientación de un usuario mucho más exacta, sin interferencia de agentes externos, mediante la utilización de sensores o dispositivos, permitirá mejorar la obtención de datos que sean utilizados para este fin. La evaluación de distintas técnicas de posicionamiento como triangulación, dead reckoning, entre otras. O a su vez la fusión de estas técnicas permitirá obtener una posición interior más exacta, y por ende una mejor navegación interior.

La utilización de balizas BLE reales para la realización de pruebas resulta algo requerido para la evaluación de un sistema de posicionamiento interior que utilice la tecnología BLE como base. El análisis de tecnologías como WiFi, o algún otro dispositivo que emita alguna señal inalámbrica, además, la fusión

de dos o más tecnologías que emitan señales inalámbricas permitirá tener un mejor contexto de su utilización en ambientes interiores.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

- Abdullah, M. N., & Korial, A. E. (2015). Planning the Path and Avoidance Obstacles for Visually Impaired/Blind People. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 17(6), 147–151.
- Abu Doush, I., Alshatnawi, S., Al-Tamimi, A.-K., Alhasan, B., & Hamasha, S. (2016). ISAB: Integrated Indoor Navigation System for the Blind. *Interacting with Computers*, 29(2), 182–202.
- Adey, P. (2017). *Mobility* (2nd ed.). New York: Taylor & Francis Group.
- Aftab, M. U. bin. (2017). *Building Bluetooth Low Energy Systems*. Birmingham: Packt Publishing Ltd.
- Asmara, R. A., Al Huda, F., Andoko, B. S., & Handayani, A. N. (2017). Optimized walking straight guidance system for visually impaired person that use Android smartphone. *2017 International Conference on Sustainable Information Engineering and Technology (SIET)*, 332–336.
- Bai, J., Lian, S., Liu, Z., Wang, K., & Liu, D. (2017). Smart Guiding Glasses for Visually Impaired People in Indoor Environment. *IEEE Transactions on Consumer Electronics*, 63(3), 258–266.
- Bellavista, P., Corradi, A., Ferrari, M., Foschini, L., Maiorano Picone, P. C., & Montanari, R. (2019). Performance evaluation of Frameworks for iBeacon-based Localization using Bluetooth Low Energy. *SummerSim '19: Proceedings of the 2019 Summer Simulation Conference*, 1–7.
- Bhargav, M. (2017). *IoT Projects with Bluetooth Low Energy*. Birmingham: Packt Publishing Ltd.
- Burd, B. (2016). *Java Programming for Android Developers for Dummies* (2nd ed.). New York: John Wiley & Sons, Incorporated.
- Cabrera-Goyes, E., & Ordóñez-Camacho, D. (2018). Posicionamiento en espacios interiores con Android, Bluetooth y RSSI. *Enfoque UTE*, 9(1), 118–126.
- Cecílio, J., Duarte, K., & Furtado, P. (2015). BlindeDroid: An Information Tracking System for Real-time Guiding of Blind People. *The 6th International Conference on Ambient Systems, Networks and Technologies (ANT 2015)*, 113–120.
- Chen, N.-Y., Shaw, J., & Lin, H.-I. (2017). Exploration method improvements of autonomous robot for a 2-D environment navigation. *Journal of Marine Science and Technology*, 25(1), 34–42.
- Cheraghi, S. A., Namboodiri, V., & Walker, L. (2017). GuideBeacon: Beacon-Based Indoor Wayfinding for the Blind, Visually Impaired, and Disoriented. *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 121–130.
- Das, A. K., Pathak, P. H., Chuah, C.-N., & Mohapatra, P. (2016). Uncovering Privacy Leakage in BLE Network Traffic of Wearable Fitness Trackers.

HotMobile '16: Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications, 99–104.

- DiMarzio, J. (2016). *Beginning Android Programming with Android Studio* (Vol. 4). New York: John Wiley & Sons, Incorporated.
- Eraghi, N. O., López-Colino, F., de Castro, A., & Garrido, J. (2015). NafisNav: an Indoor Navigation Algorithm for Embedded Systems and based on Grid Maps. *2015 IEEE International Conference on Industrial Technology (ICIT)*, 345–350.
- Ferreira, A. G., Fernandes, D., Catarino, A. P., & Monteiro, J. L. (2017). Localization and Positioning Systems for Emergency Responders: A Survey. *IEEE Communications Surveys & Tutorials*, 19(4), 2836–2870.
- Gao, J., Li, L., Bissyand, T. F., & Klein, J. (2019). On the Evolution of Mobile App Complexity. *24th International Conference on Engineering of Complex Computer Systems (ICECCS)*, 200–209.
- Google. (2019a, December 27). Platform Architecture | Android Developers. Retrieved April 18, 2020, from <https://developer.android.com/guide/platform>
- Google. (2019b, December 27). Sensors Overview | Android Developers. Retrieved July 9, 2020, from https://developer.android.com/guide/topics/sensors/sensors_overview
- Google. (2020a, June 16). Bluetooth low energy overview | Android Developers. Retrieved July 10, 2020, from <https://developer.android.com/guide/topics/connectivity/bluetooth-le>
- Google. (2020b, June 22). Position sensors | Android Developers. Retrieved July 9, 2020, from https://developer.android.com/guide/topics/sensors/sensors_position#sensors-pos-orient
- Guerrón Paredes, N. E., Cobo, A., Martín, C., & Serrano, J. J. (2018). Methodology for Building Virtual Reality Mobile Applications for Blind People on Advanced Visits to Unknown Interior Spaces. *International Association for Development of the Information Society (IADIS)-14th International Conference on Mobile Learning*, 3–14.
- Hernández-Rojas, D. L., Fernández-Caramés, T. M., Fraga-Lamas, P., & Escudero, C. J. (2018). Design and Practical Evaluation of a Family of Lightweight Protocols for Heterogeneous Sensing through BLE Beacons in IoT Telemetry Applications. *Sensors*, 18(1), 57.
- Jameel, M. I., & Dungen, J. (2015). Low-power wireless advertising software library for distributed M2M and contextual IoT. *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 597–602.
- Kacorri, H., Ohn-Bar, E., Kitani, K. M., & Asakawa, C. (2018). Environmental Factors in Indoor Navigation Based on Real-World Trajectories of Blind Users. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–12.

- Kolodziej, K. W., & Hjelm, J. (2017). *Local Positioning Systems: LBS Applications and Services* (ilustrada). Boca Ratón: CRC Press.
- Kriz, P., Maly, F., & Kozel, T. (2016). Improving Indoor Localization Using Bluetooth Low Energy Beacons. *Mobile Information Systems, 2016*, 1–11.
- Mahalik, H., Bommisetty, S., Skulkin, O., & Tamma, R. (2018). *Practical Mobile Forensics, A Hands-On Guide to Mastering Mobile Forensics for the IOS, Android, and the Windows Phone Platforms* (3rd ed.). Birmingham: Packt Publishing, Limited.
- Mathivet, V. (2018). *Inteligencia Artificial para desarrolladores Conceptos e implementación en C#* (2nd ed.). Barcelona: Ediciones ENI.
- Meliones, A., & Sampson, D. (2018). Blind MuseumTourer: A System for Self-Guided Tours in Museums and Blind Indoor Navigation. *Technologies, 6*(1), 4.
- Mendoza-Silva, G. M., Torres-Sospedra, J., & Huerta, J. (2019). A Meta-Review of Indoor Positioning Systems. *Sensors, 19*(20), 4507.
- Mussina, A., & Aubakirov, S. (2018). RSSI Based Bluetooth Low Energy Indoor Positioning. *2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT)*, 1–4.
- Needham, M., & Hodler, A. E. (2019). *Graph Algorithms: Practical Examples in Apache Spark and Neo4j*. Sebastopol: O'Reilly Media, Inc.
- Nguyen, Q. H., Johnson, P., Nguyen, T. T., & Randles, M. (2017). Optimized Indoor Positioning for static mode smart devices using BLE. *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 1–6.
- Oh, Y., Kao, W.-L., & Min, B.-C. (2017). Indoor Navigation Aid System Using No Positioning Technique for Visually Impaired People. *International Conference on Human-Computer Interaction, 714*, 390–397.
- Pani, S., & Mishra, J. (2016). A Novel Approach for Mobile Native App Development Using Ontological Design. *International Journal of Software Engineering and Its Applications, 10*(9), 105–124.
- Qian, S., Ma, Y., & Hong, D. (2019). An Improved A* Path Planning Algorithm for Indoor Intelligent Robot. *Proceedings of the 2019 International Conference on Robotics Systems and Vehicle Technology*, 144–149.
- Radius Networks. (2019). Android Beacon Library. Retrieved July 20, 2020, from <https://altbeacon.github.io/android-beacon-library/index.html>
- Röbesaat, J., Zhang, P., Abdelaal, M., & Theel, O. (2017). An Improved BLE Indoor Localization with Kalman-Based Fusion: An Experimental Study. *Sensors, 17*(5), 951.
- Rosenthal, J., & Reynolds, M. S. (2019). A 158 pJ/bit 1.0 Mbps Bluetooth Low Energy (BLE) Compatible Backscatter Communication System for Wireless Sensing. *2019 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet)*, 1–3.

- Rubio González, M. Á., Gómez Palomo, S. R., Letón Molina, E., Rodrigo San Juan, C., & Chaos García, D. (2017). *Introducción a la Informática Básica*. Madrid: Editorial UNED.
- Satan, A. (2018). Bluetooth-based Indoor Navigation Mobile System. *2018 19th International Carpathian Control Conference (ICCC)*, 332–337.
- Sato, D., Oh, U., Naito, K., Takagi, H., Kitani, K., & Asakawa, C. (2017). NavCog3: An Evaluation of a Smartphone-Based Blind Indoor Navigation Assistant with Semantic Features in a Large-Scale Environment. *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, 270–279.
- Stanley, M., & Lee, J. (2018). *Sensor Analysis for the Internet of Things*. Morgan & Claypool Publishers.
- Subedi, S., Kwon, G.-R., Shin, S., Hwang, S., & Pyun, J.-Y. (2016). Beacon Based Indoor Positioning System Using Weighted Centroid Localization Approach. *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, 1016–1019.
- Tao, Y., Ding, L., & Ganz, A. (2017). Indoor Navigation Validation Framework for Visually Impaired Users. *IEEE Access*, 5, 21763–21773.
- Timón, Á., & de León, M. (2019). *La engañosa sencillez de los triángulos: De la fórmula de Herón a la criptografía*. Madrid: Los Libros De La Catarata.
- Wang, W., & Wang, X. (2015). Research on robot indoor localization method based on wireless sensor network. *International Conference on Advances in Mechanical Engineering and Industrial Informatics (AMEII 2015)*.
- Wang, Z., Yang, Z., & Dong, T. (2017). A Review of Wearable Technologies for Elderly Care that Can Accurately Track Indoor Position, Recognize Physical Activities and Monitor Vital Signs in Real Time. *Sensors*, 17(2), 341.
- Zhang, Y., Weng, J., Dey, R., Jin, Y., Lin, Z., & Fu, X. (2019). *On the (In)security of Bluetooth Low Energy One-Way Secure Connections Only Mode*. Retrieved from <https://arxiv.org/abs/1908.10497>

ANEXOS

ANEXOS

ANEXO 1

CÓDIGO FUENTE DE LA APLICACIÓN

https://github.com/gabrielconde97/Conde_Gabriel_Tesis

ANEXO 2

MANUAL DE USUARIO DE LA APLICACIÓN

MANUAL DE USUARIO – APLICACIÓN ANDROID DE

INDICACIONES SONORAS Y POSICIONAMIENTO

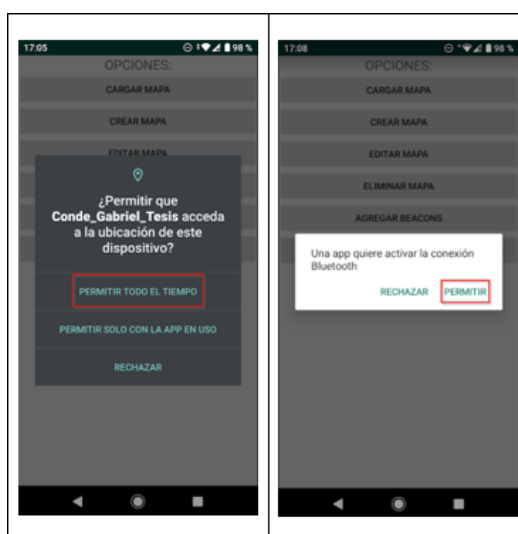
INTERIOR

1. OBJETIVO

Este manual de usuario muestra la forma de utilizar la aplicación Android de indicaciones sonoras y posicionamiento interior.

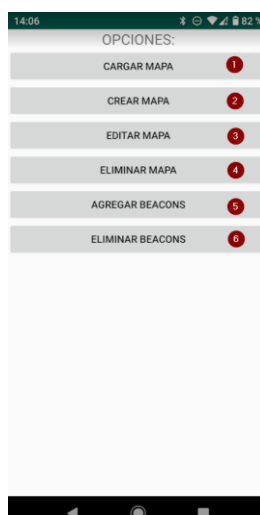
2. ASPECTOS GENERALES DE LA APLICACIÓN

La aplicación únicamente está optimizada para funcionar en dispositivos móviles (teléfonos) con una versión de Android 5.0 (Lollipop) o superior. La aplicación, para poder funcionar, requiere la otorgación del permiso de ubicación, ya que sin este no es posible encontrar balizas BLE. Además, es necesario que active el Bluetooth del teléfono para poder realizar las actividades que se encuentran en la aplicación.



3. PANTALLA PRINCIPAL

La pantalla principal de la aplicación consta de 6 distintos botones que direccionan a distintos apartados de la aplicación. Estos botones son: 1. "Cargar Mapa", 2. "Crear Mapa", 3. "Editar Mapa", 4. "Eliminar Mapa", 5. "Agregar Beacons", 6. "Eliminar Beacons".

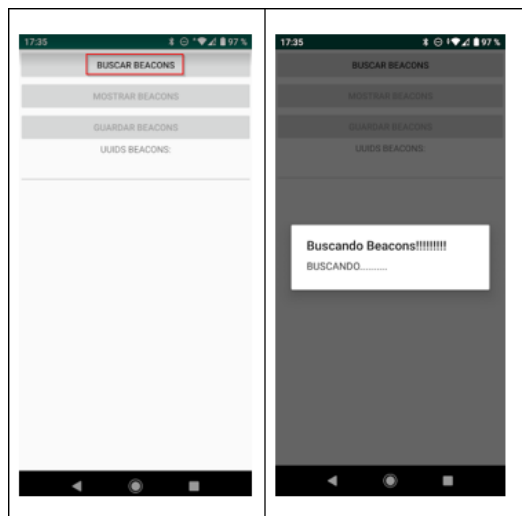


4. MANEJO DE BALIZAS BLE

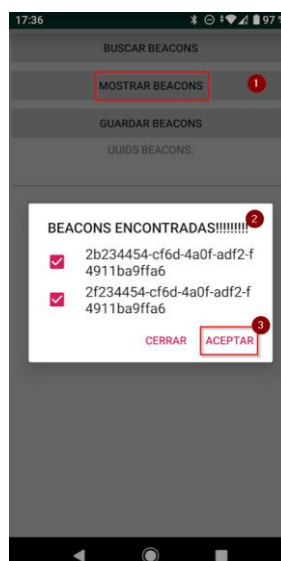
4.1. AGREGAR BALIZAS BLE

Primeramente, es necesario agregar las diferentes balizas BLE al sistema, esto permitirá agregarlas posteriormente cuando se cree un mapa de un ambiente interior en la aplicación. Este apartado corresponde al botón “Agregar Beacons” de la página principal.

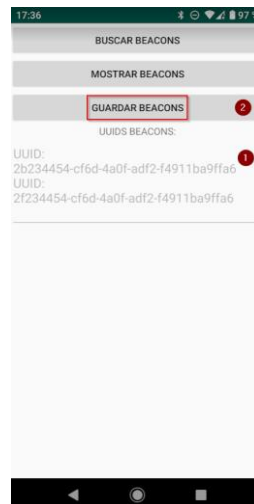
Una vez que ya nos encontremos en la página de agregar balizas, procedemos a dar clic en el botón “Buscar Beacons”, una vez dado clic, la aplicación empezara un escaneo de balizas BLE que dura 20 segundos.



Cuando ya finaliza el escaneo de balizas BLE se habilitarán los botones “Mostrar Beacons” y “Guardar Beacons”. Al hacer clic en el botón “Mostrar Beacons” se muestra una lista de las balizas BLE nuevas encontradas durante el escaneo, aquí podremos seleccionar las balizas BLE que se desee agregar a la aplicación, para confirmar nuestra selección de balizas BLE, lo hacemos dando clic en “Aceptar”.



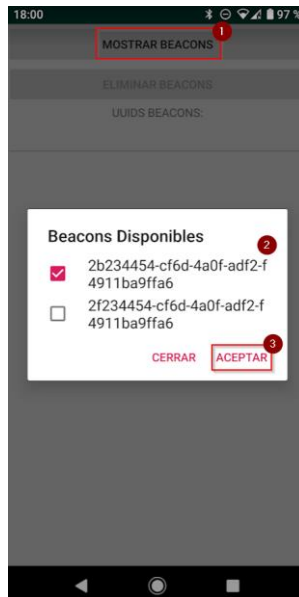
Después de confirmar nuestra selección de balizas BLE, estas se muestran en la parte inferior al grupo de botones presentes en pantalla, estas serán las balizas BLE que se guardarán en la aplicación. Para guardar las balizas BLE únicamente hay que hacer clic en el botón “Guardar Beacons”. Al finalizar de guardar las balizas BLE regresaremos a la página principal de la aplicación.



Si una baliza BLE en específico no se encuentra durante el escaneo, es posible que ya se encuentre almacenada en el sistema o a su vez la baliza BLE no está anunciándose correctamente.

4.2. ELIMINAR BALIZAS BLE

Este apartado corresponde al botón “Eliminar Beacons” de la página principal. Para poder eliminar una baliza BLE, es necesario que está no se encuentre asociada a un mapa dentro de la aplicación por lo cual, si la baliza BLE que se desea eliminar no aparece en la lista, seguramente es debido a esta situación. Para poder observar las balizas BLE que pueden ser eliminados, se debe hacer clic en el botón “Mostrar Beacons”, posteriormente se desplegara una lista con todas las balizas BLE que pueden ser eliminadas. A continuación, procedemos a seleccionar las balizas BLE que deseamos eliminar, y para confirmar nuestra selección procedemos hacer clic en “Aceptar”.



Cuando ya confirmemos nuestra selección de balizas BLE que deseamos eliminar, se habilitara el botón “Eliminar Beacons”, y la lista de balizas BLE que deseamos eliminar se muestran al final del grupo de botones pertenecientes a esta pantalla. Para eliminar las balizas BLE seleccionadas debemos hacer clic en el botón “Eliminar Beacons”, al finalizar nos redireccionara automáticamente a la pantalla principal de la aplicación.

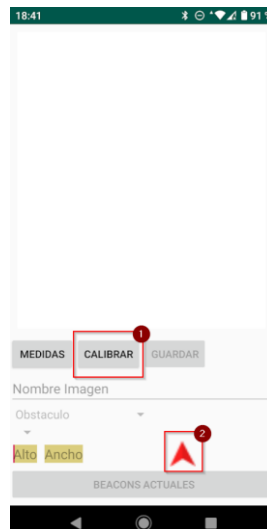


5. MANEJO DE MAPAS INTERIORES

5.1. CREAR MAPA INTERIOR

Este apartado corresponde al botón “Crear Mapa” de la página principal. Una vez que ya estemos en la página correspondiente a este apartado. El primer paso que debemos hacer es calibrar la dirección de representación

gráfica, para esto el ícono en forma de flecha presente en la pantalla nos ayudara a este proceso, este ícono en un principio funciona como una brújula, pero cuando ya hemos decidido en qué dirección vamos a graficar, siempre apuntara hacia arriba cuando estemos en la dirección que calibramos con anterioridad. Para calibrar debemos hacer clic en el botón “Calibrar”.

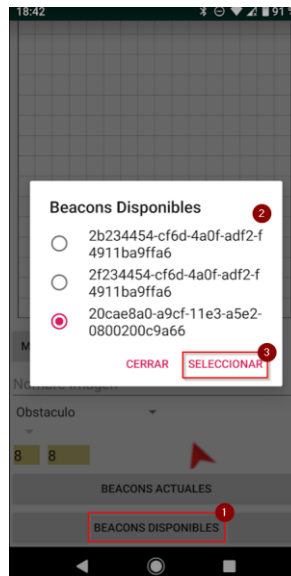


Después de calibrar la dirección de calibración, se habilitará el botón “Medidas” y los campos de texto “Alto” y “Ancho”. En los campos de texto procedemos a poner las medias del ambiente interior para poder generar la grid-map que lo represente, posteriormente, debemos hacer clic en el botón “Medidas” para poder confirmar y generar la grid-map del mapa interior en base a sus medidas anteriormente ingresadas.

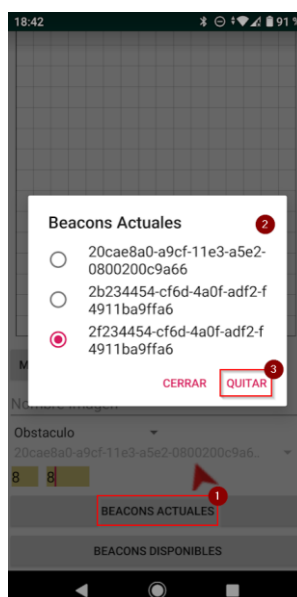


Una vez generada la grid-map del ambiente interior, ya estarán disponibles todas las opciones para poder crear el mapa, como recomendación, es preferible seleccionar las balizas BLE que estarán en el mapa, estas son necesarias que sean agregadas una a una. Para

agregar una baliza BLE al mapa debemos dar clic en el botón “Beacons Disponibles” posteriormente se desplegará una lista con las balizas disponibles para ser agregadas al mapa, seleccionamos la que deseamos agregar, después, damos clic en “Seleccionar”, podremos observar que la baliza BLE ya se encuentra disponible para poder marcar su ubicación en el mapa.



Si deseamos retirar una baliza BLE del mapa, debemos hacer clic en el botón “Beacons Actuales”, a continuación, se desplegará una lista con las balizas que están actualmente en el mapa, posteriormente seleccionamos la baliza BLE que deseamos retirar, y para finalizar procedemos hacer clic en el botón “Quitar”. Si la baliza BLE ya se encuentra plasmada en el mapa en una ubicación en específico, esta ubicación se borrará automáticamente del mapa.



Para proceder a dibujar los elementos pertenecientes al ambiente interior en la grid-map, debemos fijarnos en la categoría presente en la parte inferior al campo de texto “Nombre Imagen”. Para poder plasmar en la grid-map algún elemento en concreto debemos dar un touch en la celda de la grid-map donde queremos representarlo, y si queremos eliminarlo, de igual forma, debemos hacer touch en donde se encuentra el elemento ya existente en el mapa. Cuando seleccionemos la categoría “Beacon” se activará el segundo spinner donde podremos seleccionar la baliza BLE que deseamos mostrar en el mapa.

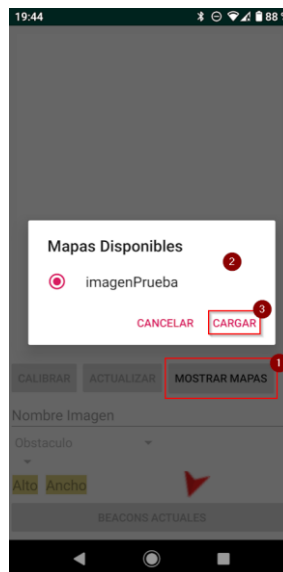


Cuando ya hemos finalizado de representar el ambiente interior en la grid-map, procedemos a darle un nombre en el campo de texto “Nombre Imagen” para posteriormente guardar el mapa interior haciendo clic en el botón “Guardar”, Cuando ya se haya guardado nos redireccionara automáticamente a la página principal de la aplicación.



5.2. ACTUALIZAR MAPA INTERIOR

Este apartado corresponde al botón “Editar Mapa” de la página principal de la aplicación. Cuando ya nos encontremos en la página correspondiente a este apartado, lo primero que debemos hacer es seleccionar el mapa que deseamos editar, para esto debemos dar clic en el botón “Mostrar Mapas” a continuación se desplegara una lista con todos los mapas guardados en la aplicación, procedemos a seleccionar el mapa que deseamos editar, y posteriormente debemos hacer clic en el botón “Cargar”.

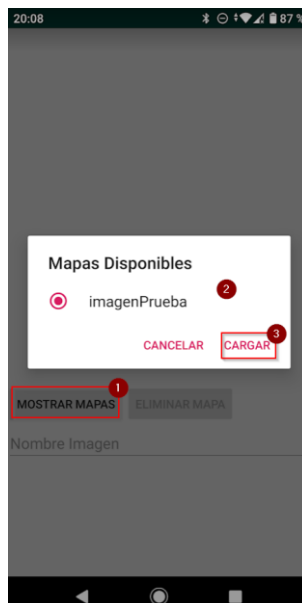


Cuando ya hemos seleccionado el mapa que deseamos editar, este se mostrara en pantalla, la forma de interacción es igual que al momento de crear un mapa, únicamente que aquí no podemos cambiar las medidas del mapa ya que esto afectaría y cambiaría la grid-map del ambiente interior y por ende la posición de los elementos que conforman el mismo. Cuando ya hemos finalizado de editar el mapa, procedemos hacer clic en el botón “Actualizar”, cuando ya se haya actualizado el mapa, nos redireccionara automáticamente a la página principal de la aplicación.



5.3. ELIMINAR MAPA INTERIOR

Este apartado pertenece al botón “Eliminar Mapa” presente en la página principal de la aplicación. Primeramente, debemos seleccionar el mapa que deseamos eliminar, para esto, procedemos a dar clic en el botón “Mostrar Mapas” a continuación se desplegará una lista con los mapas disponibles, posteriormente procedemos a seleccionar el mapa que deseamos eliminar, para confirmar la selección debemos hacer clic en el botón “Cargar”.



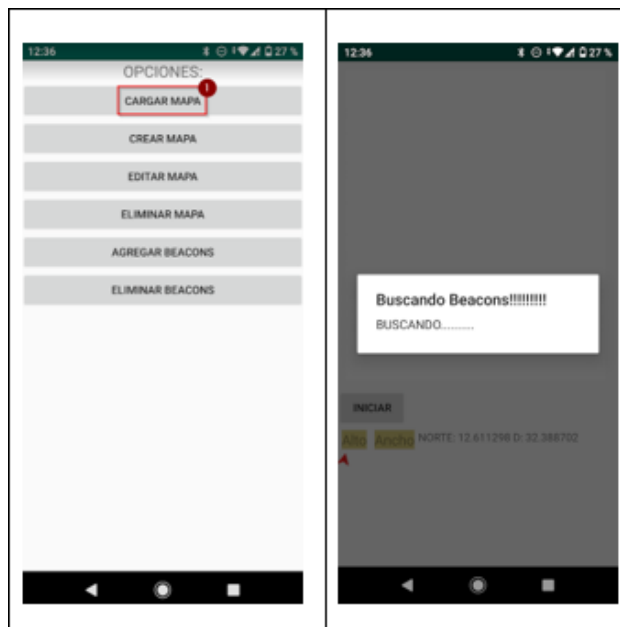
Después de seleccionar el mapa su imagen y su nombre serán mostrados en pantalla, para eliminar el mapa seleccionado debemos dar clic en el botón “Eliminar Mapa”, al finalizar de eliminar automáticamente nos redireccionara a la página principal de la aplicación.



6. INDICACIONES SONORAS Y POSICIONAMIENTO INTERIOR

6.1. INDICACIONES SONORAS

Este apartado pertenece al botón “Cargar Mapa” ubicado en la página principal de la aplicación. Cuando hemos dado clic en el botón “Cargar Mapa” automáticamente empezará un escaneo de 20 segundos en el cual buscará la baliza BLE más cerca al teléfono y cargará el mapa asociado a esa baliza BLE en específico.



Cuando la baliza BLE haya sido encontrada y haya sido asociada a un mapa en específico, dicho mapa será cargado en pantalla. Todo el manejo del mapa, tanto para marcar nuestra posición como para fijar un destino,

se realiza de manera táctil. No es posible marcar una posición sin marcar primero un destino.



Para seleccionar un destino, es necesario mantener presionado durante mínimo dos segundos en la celda en la cual queremos que sea el destino de la navegación, no se puede marcar una celda de color negro. El destino estará representado con un pequeño círculo de contorno rojo. El destino puede ser cambiado en cualquier momento siguiendo el mismo procedimiento.

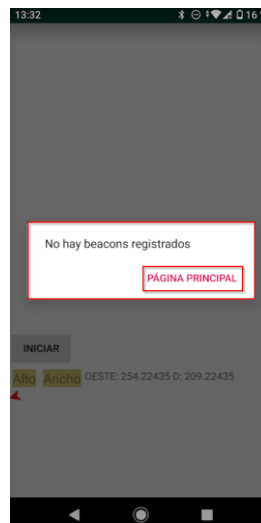


Posteriormente, para seleccionar nuestra posición ideal, debemos hacer un pequeño touch en la celda donde queremos que aparezca el ícono de color rojo que asemeja nuestra posición en un ambiente interior. A partir de seleccionar nuestra posición, ya aparecerá la ruta entre nuestra posición actual y el destino, además de las indicaciones sonoras. De igual forma solo está permitido en las celdas que no son de color negro. Esta puede ser cambiada en cualquier momento. Si deseamos iniciar el posicionamiento interior, podemos hacerlo dando clic en el botón “Iniciar”

aunque esto borrará la ruta actual, pero dando un touch en la pantalla de nuevo detendrá el posicionamiento interior y mostrará la ruta en el mapa.

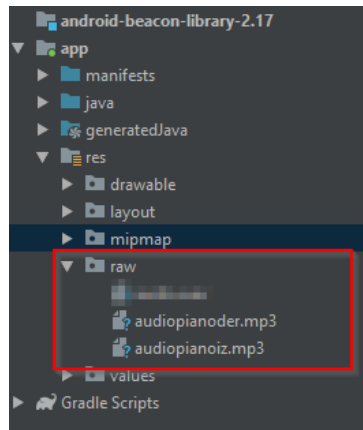


Si por algún motivo no fue cargado el mapa correctamente o no se encontró una baliza BLE en el escaneo realizado, se mostrará un mensaje y botón “Página Principal” que permite regresar a la página principal para realizar un nuevo escaneo.



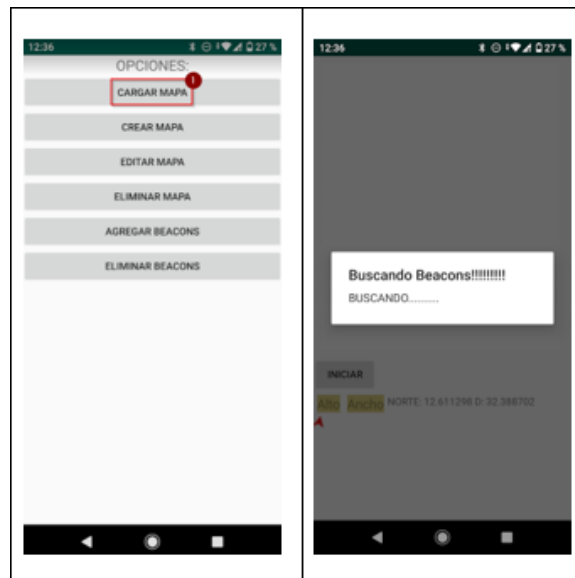
6.1.1. MODIFICACIÓN DE INDICACIONES SONORAS

Para poder cambiar el tono correspondiente de giro, se necesitan de dos diferentes tonos, uno para la izquierda y otro para la derecha. Estos dos archivos se encuentran en la carpeta “res”, posteriormente en la carpeta “raw”. Se recomienda que los dos archivos mantengan el mismo nombre y tipo de archivo: “audiopianoder.mp3” para el audio de la derecha, y “audiopianoiz.mp3” para el audio de la izquierda.

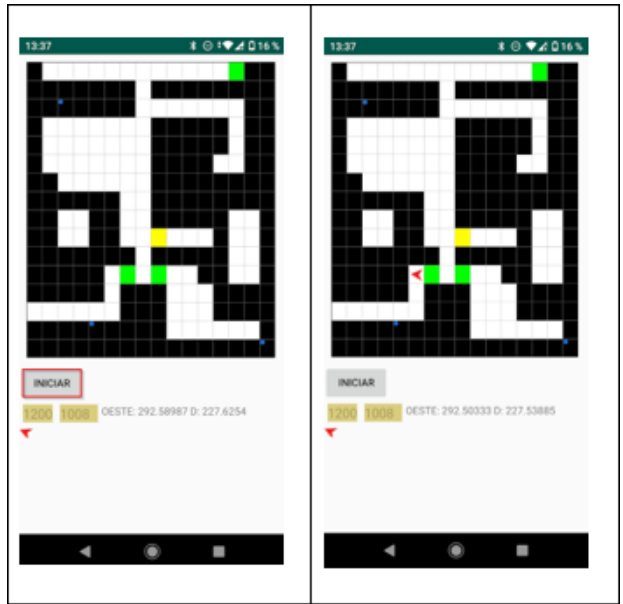


6.2. POSICIONAMIENTO INTERIOR

De igual forma que en el apartado de indicaciones sonoras, el posicionamiento interior corresponde al botón “Cargar Mapa” de la página principal de la aplicación. Cuando hemos dado clic en el botón “Cargar Mapa” automáticamente empezará un escaneo de 20 segundos en el cual buscará la baliza BLE más cerca al teléfono y cargará el mapa asociado a esa baliza BLE en específico.



Cuando el mapa ya ha sido cargado, para iniciar el posicionamiento interior, procedemos a dar clic en el botón “Iniciar” ubicado en la parte inferior del mapa. Como recomendación, debe asegurarse de que las balizas BLE están ubicadas en la posición correcta, y que su anunciación sea de forma correcta. En el caso de que esté en pantalla una ruta de navegación esta será borrada para poder mostrar la posición interior, aunque podemos volver a ella dando un touch en el mapa interior.



Si por algún motivo no fue cargado el mapa correctamente o no se encontró una baliza BLE en el escaneo realizado, se mostrará un mensaje y botón “Página Principal” que permite regresar a la página principal para realizar un nuevo escaneo.

