

2022-08

# Sistema de posicionamiento en interiores basado en la utilización de tecnología Beacon

Rivas Muñoz-Ferrada, María Fernanda

---

<https://hdl.handle.net/11673/54033>

*Repositorio Digital USM, UNIVERSIDAD TECNICA FEDERICO SANTA MARIA*

**UNIVERSIDAD TÉCNICA FEDERICO SANTA  
MARÍA  
DEPARTAMENTO DE ELECTRÓNICA  
VALPARAÍSO - CHILE**



**“SISTEMA DE POSICIONAMIENTO EN  
INTERIORES BASADO EN LA  
UTILIZACIÓN DE TECNOLOGÍA BEACON”**

**MARÍA FERNANDA RIVAS MUÑOZ-FERRADA  
JUAN IGNACIO SÁNCHEZ ALARCÓN**

**MEMORIA DE TITULACIÓN PARA OPTAR AL  
TÍTULO DE INGENIERO CIVIL TELEMÁTICO**

**Profesor Guía: Nicolás Jara Carvallo  
Profesor correferente: José Manuel Martínez**

**Agosto - 2022**

# Índice

<b>1. AGRADECIMIENTOS</b>	<b>7</b>
1.1. Agradecimientos María Fernanda . . . . .	7
1.2. Agradecimientos Juan Ignacio . . . . .	8
<b>2. CAPÍTULO 1: INTRODUCCIÓN</b>	<b>9</b>
2.1. Introducción . . . . .	9
2.2. Problemas a resolver . . . . .	12
2.3. Acercamiento a la solución . . . . .	14
2.4. Estado del Arte . . . . .	16
<b>3. CAPÍTULO 2: DISEÑO DE LA SOLUCIÓN</b>	<b>18</b>
3.1. Objetivos . . . . .	18
3.1.1. Objetivo General . . . . .	18
3.1.2. Objetivos Específicos . . . . .	18
3.2. Requisitos Funcionales y No Funcionales . . . . .	18
3.3. Requisitos de Interfaz . . . . .	20
3.4. Requisitos de Ambiente . . . . .	23
3.4.1. Hardware de Desarrollo . . . . .	23
3.4.2. Software de Desarrollo . . . . .	23
3.5. Entorno de soporte y desarrollo . . . . .	23
3.6. Hitos principales del proyecto . . . . .	23
3.7. Estructura del sistema . . . . .	25
3.7.1. Contexto del sistema . . . . .	25
3.7.2. Esquema general del sistema . . . . .	26
3.7.3. Descripción Componentes . . . . .	26
3.8. Diseño de interfaces . . . . .	34
3.8.1. Modelo de navegación . . . . .	34
3.8.2. Diseño preliminar de las vistas . . . . .	34
<b>4. CAPÍTULO 3: RESULTADOS</b>	<b>40</b>
4.1. Datos obtenidos : beacons . . . . .	40

4.1.1.	Tratamiento de datos obtenidos: distancia . . . . .	40
4.1.2.	Implementar triangulación: posición usuario . . . . .	43
4.2.	Funciones principales de la aplicación . . . . .	44
4.3.	Verificación . . . . .	47
4.4.	Caso de uso . . . . .	47
4.5.	Simulación del prototipo . . . . .	53
4.5.1.	Posicionamiento de los beacons . . . . .	53
4.5.2.	Señales recibidas y tratadas . . . . .	56
4.5.3.	Distancia y trilateración . . . . .	57
4.5.4.	Creación de nodos en el plano . . . . .	57
4.5.5.	Creación de algoritmos de rutas más cortas: dijkstra y astar (A*) . . . . .	58
4.5.6.	Eventos de tiempo . . . . .	60
4.6.	Análisis crítico y evaluación de riesgos . . . . .	60
4.6.1.	Análisis crítico . . . . .	60
4.6.2.	Riesgos . . . . .	61
<b>5.</b>	<b>CAPÍTULO 4: CONCLUSIONES</b>	<b>62</b>

## Índice de figuras

1.	Diagrama de interacción aplicación GUIA.ME. . . . .	22
2.	Diagrama de contexto Desafío 7 . . . . .	25
3.	Esquema general del sistema . . . . .	26
4.	I10 indoor location beacon. . . . .	27
5.	Esquema de triangulación. . . . .	29
6.	Nodos edificio C. . . . .	32
7.	Modelo de navegación. . . . .	34
8.	Izquierda: Inicio de la aplicación, Derecha: Menú de la aplicación. . . . .	35
9.	Izquierda: Selección de destino, Derecha: Ruta tentativa. . . . .	36
10.	Izquierda: Seguimiento del usuario, Derecha: Aviso de llega a destino. . . . .	37
11.	Izquierda: Mapa alternativo, Derecha: Confirmación de la sala. . . . .	38
12.	Izquierda: Indicaciones, Derecha: Llegada a destino. . . . .	39
13.	Beacon con UIDD1: AC233F8A8804A . . . . .	41
14.	Beacon con UIDD2: AC233F8A88044 . . . . .	41
15.	Beacon con UIDD3: AC233F8A88045 . . . . .	42
16.	Valores en consola . . . . .	42
17.	Tres beacons y su respectiva distancia al usuario. . . . .	43
18.	Vistas funcionales de la aplicación . . . . .	44
19.	Funciones del mapa principal . . . . .	45
20.	Funciones del mapa alternativo . . . . .	46
21.	Funciones de vista de seguimiento . . . . .	47
22.	Selección de destino. . . . .	48
23.	Ruta tentativa después de seleccionar destino . . . . .	48
24.	Botón que inicia el seguimiento del usuario. . . . .	49
25.	Vista de seguimiento al usuario. . . . .	50
26.	Ventana emergente que se muestra al usuario que llego a su destino. . . . .	50
27.	Izquierda: mapa principal, Derecha: mapa alternativo. . . . .	51
28.	Izquierda: seleccionar configuraciones, Derecha: configuraciones. . . . .	51
29.	Izquierda: mapa alternativo, Derecha: mapa principal. . . . .	52
30.	Izquierda: mapa alternativo, Derecha: menú configuraciones. . . . .	53

31.	Posicionamiento de los beacons en el plano. . . . .	55
32.	Posicionamiento línea vista de los beacons y el usuario en el plano. . . . .	55
33.	Izquierda son los datos de la señal y derecha la conversión a distancia . . . . .	56
34.	Nodos .csv del edificio C. . . . .	57
35.	Ruta tentativa después de seleccionar destino . . . . .	58
36.	Parte del 3er pasillo del plano del edificio C, 2do piso. . . . .	59
37.	Ruta tentativa después de seleccionar destino . . . . .	60

## Índice de tablas

1.	Eventos externos. . . . .	20
2.	Respuesta del sistema . . . . .	21

# 1. AGRADECIMIENTOS

## 1.1. Agradecimientos María Fernanda

Cerrando este ciclo universitario, agradezco a todas las personas que fueron parte de el y me apoyaron para terminarlo, en especial a mi madre que me sacó adelante y confiaba en mis capacidades.

A esos amigos que uno se encuentra en este camino como lo son Kathy, Cata y Pelón; agradezco al destino por ponerlos a mi lado. Además, a mi compañero de memoria Juanito, con quien nunca pensé que llegaríamos tan lejos juntos, a pesar de lo poco que nos conocíamos la conexión fue grande. Gracias por aguantarme we. Estoy muy orgullosa de vos.

También agradecer a esos amigos que ya estaban cuando empecé esta etapa, como Conchita que es de los primeros amigos que hice cuando llegue a este país y ha evolucionado nuestra amistad gracias a nuestro recorrido juntos; y en algún momento a mi amiga que esta lejos, Nelly.

Finalmente, a mi pareja Milo (mi chichi), quien me dio a conocer esta hermosa carrera y gracias a él estoy terminando esta etapa. Agradecer infinitamente a la vida por a todas las experiencias junto a él. Gracias por tu apoyo incondicional en momentos felices o difíciles tanto académicos como personales. Espero que la vida nos regale más tiempo juntos para seguir abriendo y cerrando ciclos. Te amo.

Y gracias mi gordito bello, mi gatito, lo amo tanto.

Por último a mi profesor guía Njara, me encanta la dinámica que tiene para enseñar y su compromiso con el alumnado.

## 1.2. Agradecimientos Juan Ignacio

Primero que todo, agradezco a mi familia ya que siempre han estado allí a pesar de mis colapsos y siempre me han brindado todo su apoyo, me han dado la mejor educación que pudieron y la persona que soy hoy es en gran medida gracias a ellos. Agradezco a mis tías, mis primos y a mis abuelitas que también me apoyaron en todo lo que pudieron y hacían de mis visitas a La Serena muy cálidas y acogedoras.

Mafesita gracias por soportarme todos estos años ya que a pesar de que nos conocimos como 3 veces en distintos ramos y las cosas no cuajaban, terminó siendo una amistad super buena y la valoro mucho, estoy seguro que te va a ir bien en todo lo que te propongas mijita :3

Agradezco a toda la gente que conocí en la pensión del Leo y con los que viví varios años, no conocía a nadie en la región y me ayudaron bastante a crecer y socializar, siempre los recordaré con cariño.

Le agradezco a la Javiera por seguir siendo mi amiga a pesar de que hemos tenido muchos altos y bajos. Me has ayudado a madurar un montón y sin ti mi vida no sería igual.

Agradezco a todos esos Mets (En especial a Hernán) con los que compartí un par de años he hicieron de mi paso por Metalurgia mucho más agradable.

Gracias a la comunidad Telemática por mostrarme lo maravillosa que es esta carrera y en especial a Conchita, Milito y Mafesita (otra vez) por ser un gran apoyo estos años y brindarme su amistad, la cual espero dure muchos años más.

Y por último agradezco a mi profesor guía y lo entretenidas que fueron todas sus clases y lo comprometido que está con los estudiantes.

## 2. CAPÍTULO 1: INTRODUCCIÓN

### 2.1. Introducción

Existen muchas construcciones y espacios de gran extensión, y estos van en aumento a medida que pasa el tiempo. Cuando las personas utilizan estos lugares, suelen necesitar herramientas de orientación tales como mapas físicos o digitales que le permiten llegar a sus destinos con mayor facilidad. Sin embargo, si no se cuenta con estas herramientas, los sentidos, y en especial la vista, permiten de manera innata dimensionar el espacio y ubicar sectores, pero si la vista está deteriorada, será más difícil poder ubicarse en un recinto.

Actualmente, una de las herramientas para estos fines es Google Maps, que fue estrenada en el año 2005 y hoy en día figura como una de las herramientas de mapas más utilizada. Esta aplicación dispone de los mapas de más de 220 países, además de guías por voz para llegar a cualquier lugar en automóvil, bicicleta o caminando. Posee actualizaciones de tráfico de más de 15.000 ciudades y, sobre todo, dispone de Street View. Además de lo anterior, en el año 2013 surge Waze: una aplicación social de tránsito automotor en tiempo real y navegación asistida por GPS desarrollada por Waze Mobile.

Estos servicios de guía pueden ubicar a todas las personas en el mundo, e incluso pueden guiar dentro de recintos cerrados debido a sistemas de fotografías que permiten visibilizar el interior y generar rutas.

En los últimos años, los avances tecnológicos han propiciado el aumento en las capacidades de cómputo, disminución del tamaño de los dispositivos y reducción de costos. Estos factores han permitido el surgimiento de nuevos paradigmas computacionales como el Internet de las Cosas (IoT, por sus siglas en inglés Internet of Things), que se puede describir como la conexión de objetos cotidianos a Internet en donde los dispositivos están conectados de una manera inteligente, permitiendo nuevas formas de comunicación entre cosas y personas (Bhide, 2014)[1]. IoT puede aplicarse en varias áreas tales como en el hogar, ciudades, automotores y salud, entre otros, pues la cantidad de usos depende de la creatividad e ingenio de los desarrolladores. En este sentido, es factible pensar en la integración de IoT puede ayudar a las personas con discapacidad visual al momento de desplazarse en espacios cerrados.

De acuerdo con cifras oficiales de la Organización Mundial de la Salud (OMS, 2017), en el mundo existen aproximadamente 253 millones de personas con discapacidad visual, de las cuales 36 millones son ciegas y 217 presentan baja visión. A pesar de que existen diversas herramientas tecnológicas, “Los sistemas de ayuda para personas con discapacidad visual siguen siendo los métodos tradicionales, como el perro guía o el bastón blanco”(Paniagua, 2016)[2]. Por tal motivo, IoT representa un área de oportunidad y un desafío tecnológico en la creación de tecnologías accesibles para personas con discapacidad visual.

En relación a lo anterior, las IoT pueden ser integradas a través de los beacons, que son dispositivos pequeños que incorporan un sensor Bluetooth de bajo consumo que es capaz de emitir una señal de radio de hasta 70mts hacia un dispositivo rastreador (Bacchetta, 2017)[3]. Los beacons están utilizándose en diversas organizaciones en todo el mundo debido a su bajo costo, por ejemplo, en aeropuertos, museos, festivales y bancos, entre otros (Sánchez, 2016)[4]. Actualmente, uno de los usos más populares se enfoca en el marketing de proximidad, es decir, sorprender a los clientes de un local comercial con descuentos, ofertas y novedades, entre otros. Además, existen otros contextos en donde se puede aplicar esta tecnología, por ejemplo, en ubicación en interiores, check-in automático y pagos sin contacto (Bacchetta, 2017)[3].

En este sentido, los beacons pueden ser utilizados para ayudar en el desplazamiento de personas con discapacidad visual en espacios cerrados debido a que su principal ventaja es la gran precisión que tienen en edificios, lo cual se conoce como geolocalización indoor o de interiores (Bacchetta, 2017)[3]. Es así como los beacons permiten conocer la ubicación exacta del dispositivo del usuario, independientemente si se encuentra en un espacio cerrado o en el exterior. Por otra parte, los beacons tienen la ventaja de comunicarse con tecnologías que ya están lo suficientemente incorporadas en nuestras vidas, como los dispositivos móviles que en conjunto podrían brindar una solución factible en el desarrollo de tecnologías accesibles para personas con discapacidad visual, específicamente en espacios cerrados.

En la actualidad, los dispositivos móviles se utilizan para la comunicación, diversión y acceso a la información, sin embargo, cuentan con otras características que resultan de gran utilidad en el desarrollo de tecnologías accesibles, por ejemplo, la capacidad para emitir sonido y la integración de sensores como acelerómetro y magnetómetro que permiten conocer la dirección hacia la que se dirige un usuario, y estas características ofrecen grandes oportunidades de aprovechamiento en la creación de tecnologías accesibles.

Debido a lo anterior, en este documento se presenta el Sistema de Orientación para Personas en Espacios Cerrados mediante Tecnología beacons denominado Guia.me. Se incluye además un caso de estudio en el que se implementa el sistema en el Campus Casa Central Valparaíso de la Universidad Técnica Federico Santa María, específicamente en el Edificio C, para brindar información de utilidad, puntos de interés y sus obstáculos a los estudiantes, profesores o visitantes al momento de desplazarse de un punto a otro, inclusive si estas personas presentan algún grado de discapacidad visual.

## 2.2. Problemas a resolver

Actualmente, existen diversas tecnologías que permiten ayudar a las personas con discapacidad visual al momento de desplazarse de un lugar a otro. Una de estas tecnologías es el GPS que permite ubicar, con cierta precisión, a una persona a través de las señales emitidas por los satélites (Ortiz y Corredor, 2013)[5]. Sin embargo, esta tecnología se enfoca en cubrir espacios abiertos, y por consiguiente resulta obsoleta la implementación dentro de espacios cerrados como edificios, escuelas, centros comerciales, entre otros. Por tal motivo, surge la necesidad natural de disponer de nuevas soluciones tecnológicas que permitan ayudar a las personas con discapacidad visual a desplazarse dentro de espacios cerrados.

En efecto, hoy existen diversas herramientas tecnológicas que colaboran en el desplazamiento de personas con discapacidad visual dentro de espacios cerrados, por ejemplo, bastones y pulseras inteligentes que a través de sensores permiten detectar un obstáculo en el camino y avisar al usuario. Sin embargo, a pesar de la gran variedad de dispositivos tecnológicos que existen, Paniagua (2016)[2] afirma que los medios tradicionales como el bastón blanco y los perros guías siguen siendo el principal medio de orientación para las personas con discapacidad visual. En este sentido, es factible pensar en nuevas soluciones que permitan complementar los actuales medios utilizados para orientarse.

Como se mencionó anteriormente, las aplicaciones móviles con tecnología beacon solo han sido implementadas para ubicar distintos objetos, y casi no se ha considerado la función de contribuir en la creación de una guía, a tiempo real, dentro de una institución de gran extensión (lo más parecido es lo efectuado en las minas para el seguimiento de sus trabajadores, de manera tal, que se le puede entregar rutas seguras de escape).

Es así como el beacon es un dispositivo que se aplica para dar soporte en la orientación de interiores, por ejemplo, Nav-Cog (Ahmetovic, Gleason, Ruan, Kitani, Takagi y Asakawa, 2016)[6] y SmartGuia (Soares, 2014)[7] son sistemas que utilizan dispositivos móviles para brindar la información obtenida de cada beacon, dependiendo del contexto en que se encuentre. Sin embargo, ninguno de estos sistemas considera la orientación del dispositivo de la persona como elemento clave para brindar información de utilidad.

Para Valdez (2000)[8] el principal problema que tienen las personas con discapacidad visual es la orientación, lo que dificulta hacer sus actividades cotidianas. Por lo tanto, aparte de brindar

información del lugar en donde se encuentra, es necesario buscar soluciones tecnológicas que permitan conocer la orientación del usuario, y a través de esta guiarlos en su desplazamiento indicándoles la información que se encuentra a su alrededor, tal como obstáculos y proximidad hacia un punto (distancia), entre otros. El objetivo final será, entonces, mejorar la autonomía de las personas con y sin discapacidad visual al momento de desplazarse en espacios cerrados.

Otro problema detectado en la mayoría de las universidades es que no existe una herramienta inclusiva para informar a las personas respecto al camino a seguir dentro del recinto, y gran parte de los asistentes se guía mediante paneles posicionados en ciertos sectores de los campus, que al ser estáticos, no pueden ser consultados a mitad del trayecto. Además, no son lo suficientemente atractivos para una persona con problemas de visión y resultan imposibles de utilizar por personas ciegas. Por estos motivos se vuelve necesaria la creación de un servicio de asistencia por medio de voz, pues serviría para guiar a las personas no videntes y a cualquiera que desee que su recorrido sea asistido.

### **2.3. Acercamiento a la solución**

Para abordar esta problemática se desarrolla una aplicación que le permita a las personas desplazarse con facilidad y confianza por recintos tales como un hospital, centros comerciales, aeropuertos, parques de diversiones o una universidad, entre otros. Para ello, en estos lugares se debe distribuir previamente un sistema de red beacon determinado por un plano correspondiente a la instalación de interés. Asimismo, se puede mencionar que, dependiendo del enfoque que tenga cada una de estas instituciones, se desarrolla un modelo de negocio distinto debido a que se rescata un valor diferente en cada situación, dependiendo del rubro.

Ejemplo de lo anterior sería un centro comercial: si una persona ciega cuenta con la aplicación podrá llegar a su destino exitosamente y en menor tiempo que sin la utilización de la aplicación, por lo que se generará un incentivo para volver a comprar en las tiendas que se encuentren en ese recinto. Para ello es necesario poseer el sistema de guía y localización que incentive a regresar a ese centro comercial, lo que generaría mayores ventas, sumado a que el usuario tendría la posibilidad de ser informado sobre las ofertas de las tiendas que se encuentran en su recorrido, si así lo desea.

En el caso de un aeropuerto, este sistema daría mayor seguridad para el viajero al momento de abordar el vuelo, debido que le permitiría conocer la instalación que se visita a través de esta herramienta que permite optimizar el tiempo, disminuyendo así la posibilidad de desorientación y pérdida de vuelos. Y si una institución, como las mencionadas anteriormente, quiere desarrollar su propio sistema beacon, se podría ofrecer una herramienta para que las personas puedan desenvolverse en estos espacios sociales con confianza.

En el caso de las personas con discapacidad visual, el aumento de la seguridad en el desplazamiento sería una motivación para realizar actividades tales como ingresar a una universidad, lo que resolvería en parte la escasez de herramientas inclusivas en estas instituciones. Ello impulsaría a personas con visibilidad reducida o no videntes a realizar estudios superiores y desenvolverse en sociedad con oportunidades similares, pudiendo convertirse en un aporte para el país desempeñándose, como profesionales en un futuro cercano.

A partir de las problemáticas mencionadas anteriormente, en este proyecto de tesis se intentará dar solución desarrollando una aplicación que permita guiar a las personas por ciertos sectores de una universidad. Esto se realizará a través de la instalación de un sistema de red beacon que

considere un asistente de voz para personas invidentes y de visión reducida, con la finalidad de mejorar la experiencia universitaria de cada uno de los estudiantes.

Uno de las finalidades que se desea cumplir en este proyecto de tesis, es que logre ser sostenible en el tiempo, por lo que se espera trabajar en una propuesta de valor social, medioambiental y económica, con el fin de que genere un impacto positivo.

El tipo de utilización que se le dará a este producto destaca como un emprendimiento innovador, ya que contempla una nueva tecnología y se programa de tal manera que cumpla la función de llevar a la persona a su destino de forma óptima y segura, presentando una oportunidad de mejora para las universidades. Como es un producto nuevo en el mercado, se espera que en un futuro sea comercializado en gran cantidad de instalaciones a lo largo del país y posiblemente al extranjero.

Para conseguir este sistema se creará una aplicación móvil que permita a los usuarios identificar sus destinos utilizando su ubicación, además de incluir una guía que posea voz y permita moverse dentro de un recinto con gran extensión por medio de instrucciones.

Para comenzar se apuntará a la creación de un algoritmo de posicionamiento de la tecnología beacon, el cual entregará una propuesta de posicionamiento de esta tecnología, maximizando la cobertura del recinto. Para ello es necesario crear una matriz de posicionamiento en un plano de la universidad, la cual entregará la posición de los beacons después de ejecutar el algoritmo.

Posteriormente se crearán algoritmos de ubicación por medio de los beacons, con la finalidad de reconocer el posicionamiento de una persona entre 3 beacons, y así conocer la distancia a la que se encuentra el usuario, pudiendo obtener la ubicación.

A continuación se procederá a crear un algoritmo que permita la creación de rutas por medio de los nodos posicionados en sus respectivos puntos de interés. Luego, se creará un sistema por medio de voz, el cual debe ser capaz de entregar la información de la ruta por medio de un asistente virtual.

Finalmente, se espera integrar lo anteriormente mencionado en una aplicación que pueda mostrar el mapa de la universidad, además de presentar una ruta a seguir según el punto marcado por el usuario.

## 2.4. Estado del Arte

En la industria de la tecnología móvil existen muchas compañías que proponen diversos sistemas operativos (SO), y dentro de estas destaca SO Android de Google pues tiene mayor aceptación entre los desarrolladores de aplicaciones [11]. Este SO posee la ventaja de ser software libre, lo que permite a un desarrollador de software tener acceso completo al sistema para diseñar e implementar cualquier tipo de herramienta, por ejemplo, la aplicación que se describe en el artículo [8] cuyo nombre es TEUBICA, la cual proporciona ayuda a las personas con discapacidad visual (PcDV) para reconocer la ubicación o el lugar en dónde se encuentran. Existen trabajos similares al que se propone en el artículo, tales como el PYOMSystem [14], aplicación que utiliza teléfonos inteligentes, y que cuenta con tecnología Wi-Fi que permite identificar la posición y orientación de una persona ciega en un ambiente cerrado. Otros trabajos que involucran una tecnología que permita la movilidad de personas con discapacidad visual en los medios de transporte público son el OnTheBus [13] y el App&Town, los cuales funcionan en Barcelona y Alemania [12]. Estas aplicaciones actúan mediante una interfaz para PcDV, y son utilizados en el metro de estos países gracias a la impecable organización del transporte y al apoyo de tecnologías como el GPS que se encuentra incorporado en todo el sistema de locomoción.

De acuerdo con lo anterior, la utilización de tecnologías para ubicar personas y poder guiarlas se ha extendido año tras año, llegando a reducir sus tamaños y aumentar funcionalidades en aparatos tecnológicos como el beacon. A pesar de que en los últimos años la utilización de la tecnología beacon ha sido enfocada en localizar objetos, hoy en día ha evolucionado y puede ser utilizada en espacios abiertos y cerrados, puesto que se usan para mandar señales bluetooth a los teléfonos para realizar marketing y promocionar productos cercanos a algunos lugares por medio de mensajes o voz [4]. Algunas empresas como Accent utilizan beacons para hacer localización indoor [5], debido al alcance que estos poseen y al potencial que se puede extraer de la tecnología.

Para el desarrollo de proyectos en donde se implementan beacons para localización, ya sea de interiores o exteriores, se utiliza como base la monitorización de la intensidad de la señal de radio, el denominado indicador de intensidad de la señal recibida (RSSI) [1] en donde se puede indicar la cercanía o lejanía del objeto de interés que se está monitoreando.

La utilización de RSSI es de suma importancia debido a que por medio de esta señal se pueden

aplicar diversos métodos de localización en interiores, como por ejemplo en la triangulación, Fingerprinting ,Bluetooth-Based Localization, entre otros métodos [2]. Sumado a esto, hay diferentes tipos de protocolos según el beacon a utilizar, y estos van desde los desarrollados por la empresa Google Eddystone, y por la empresa Apple, ibeacon [3]. Del mismo modo, los proyectos se basan en la utilización de planos del sector que se desea cubrir para estimar una aproximación de la ubicación de los beacons [2], por lo que es necesario contar con el lugar ideal para poder hacer las pruebas.

Así mismo, al momento de utilizar los beacons para interiores es necesario contar con la tecnología actualizada y las herramientas necesarias para su correcto desarrollo, en especial para lograr aplicar técnicas para detectar a las personas y su movimiento dentro del recinto.

En resumen, dentro de los artículos revisados se presentan distintas soluciones para la detección de personas y posicionamiento de los beacons. Además, el proyecto por desarrollar consiste en tomar las técnicas existentes para guiar a personas detectándolas mediante sus teléfonos móviles para marcarles una ruta a seguir en un mapa que los llevará a sus respectivos destinos.

## 3. CAPÍTULO 2: DISEÑO DE LA SOLUCIÓN

### 3.1. Objetivos

#### 3.1.1. Objetivo General

Crear una aplicación para teléfono móvil que permita guiar a las personas hacia sus destinos dentro de un recinto de gran extensión, orientado para personas con alguna discapacidad.

#### 3.1.2. Objetivos Específicos

- **OE1:** Analizar el desafío para su correcto desarrollo.
- **OE2:** Metodología para convertir planos en matriz de coordenadas.
- **OE3:** Metodología para determinar las posiciones de los beacons.
- **OE4:** Metodología para determinar la ubicación del usuario.
- **OE5:** Desarrollar un método para crear rutas en el mapa.
- **OE6:** Metodología para realizar un seguimiento del usuario.
- **OE7:** Desarrollar un método que permita una guía por medio de voz.
- **OE8:** Implementación de los métodos previamente desarrollados.
- **OE9:** Realizar una evaluación de prefactibilidad del proyecto.

### 3.2. Requisitos Funcionales y No Funcionales

- **FUNCIONALES:**

1. **RF1** [ El sistema debe ser capaz de detectar la posición de cada persona]
2. **RF2** [ El sistema debe ser capaz de mostrar un mapa]
3. **RF3** [ El sistema debe ser capaz de mostrar rutas guías ]
4. **RF4**[ El sistema debe ser capaz de entregar instrucciones por voz]
5. **RF5** [ El sistema debe ser capaz de ser capaz de avisar cuando se llega al destino deseado]

6. **RF6** [ El sistema debe permitir seleccionar destino ]

■ **NO FUNCIONALES:**

1. **RNF1**: [ Escalabilidad: Poder agregarle más funcionalidades ]

2. **RNF2**: [ Detectar a tiempo real la posición de la persona y el movimiento de esta ]

3. **RNF3**: [ Seguridad: cifrado de identificadores de los beacons ]

### 3.3. Requisitos de Interfaz

Tabla 1: Eventos externos.

<b>Evento</b>	<b>Descripción</b>	<b>Iniciador</b>	<b>Parámetros</b>	<b>Respuesta</b>
<b>Abrir aplicación</b>	Se abre la aplicación.	Un usuario ingresa a la aplicación.	Ubicación usuario, id beacons, intensidad de señal beacons.	Mostrar mapa y ubicación dentro de este.
<b>Seleccionar destino</b>	Ingreso del destino en buscador.	El usuario ingresa el destino al que desea ir.	Id destino.	Se ingresa el destino en la barra de búsqueda.
<b>Mostrar ruta</b>	Se muestra la ruta a seguir por el usuario.	El usuario selecciona el destino.	Ubicación usuario , Id destino, Id beacons.	Se muestra ruta a seguir para llegar al destino.
<b>Seguimiento usuario</b>	Se realiza un seguimiento del usuario a medida que avanza en el mapa.	El usuario camina hacia su destino y sigue.	Tramos, Id destino, ubicación usuario.	Se muestra ruta que queda para llegar a su destino.
<b>Guía de voz</b>	Se realiza seguimiento del usuario y se entregan indicaciones por medio de voz.	El usuario sigue la ruta	Ubicación usuario , Id destino, Tramo.	Voz que entrega indicaciones de la ruta.
<b>Llegada a destino</b>	Se alerta al usuario que llegó a su destino.	El usuario llega a su destino.	Ubicación usuario , Id destino.	Se muestra una alerta indicando que llegó al destino.

Continúa en la próxima página

**Tabla 1 – Continuación de la pagina anterior.**

<b>Evento</b>	<b>Descripción</b>	<b>Iniciador</b>	<b>Parámetros</b>	<b>Respuesta</b>
<b>Mapa alternativo</b>	Mapa con instrucciones visuales y voz.	El usuario cambia el modo del mapa.	Ubicación usuario , Id destino.	Se muestran solo indicaciones y existe una guía por voz.

Tabla 2: Respuesta del sistema

<b>Respuesta</b>	<b>Descripción</b>	<b>Parámetros</b>
<b>Mostrar mapa</b>	Se mostrará un mapa de la universidad al momento de iniciar.	Ubicación usuario, Id beacon.
<b>Mostrar mapa alternativo</b>	Se mostrará un mapa alternativo.	Ubicación usuario, Id beacon.
<b>Mostrar ruta</b>	Al momento de seleccionar el destino deseado se mostrará una ruta para llegar.	Ubicación usuario, Id beacon, Tramos.
<b>Mensajes por voz</b>	Cuando el usuario se encuentre caminando para llegar al destino se irán entregando mensajes por medio de voz que le den instrucciones de a cuantos metros tiene que girar o seguir en línea recta.	Ubicación usuario,Tramos.

Se elaboró un diagrama de interacción que aclara de mejor manera el funcionamiento de la aplicación. En la Figura 1 se puede detallar el flujo que tiene la aplicación al momento de solicitar el mapa para luego seleccionar el destino y posteriormente crear la ruta óptima para el usuario. Seguido de esto, se le realiza un seguimiento y se activa el asistente de voz.

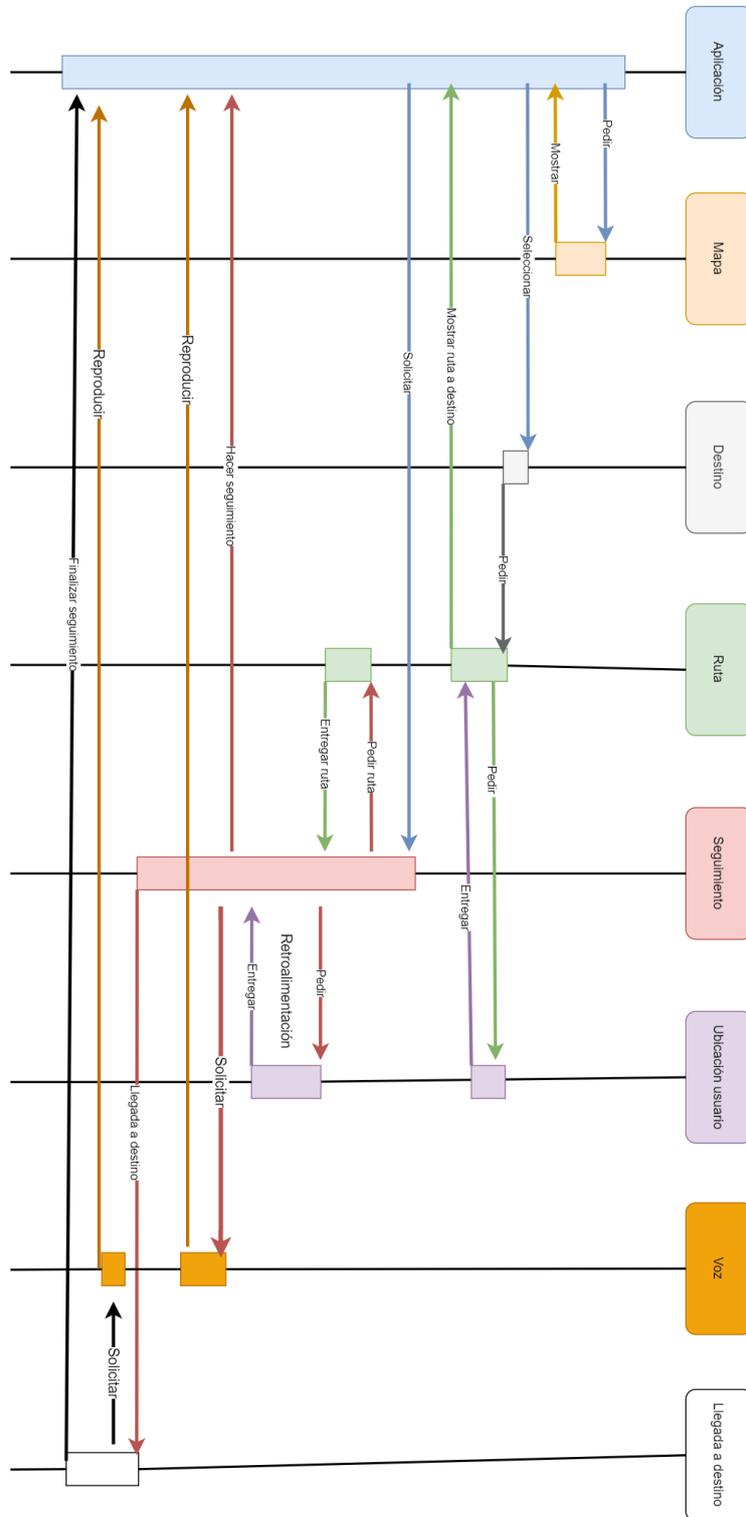


Figura 1: Diagrama de interacción aplicación GUIA.ME.

### 3.4. Requisitos de Ambiente

#### 3.4.1. Hardware de Desarrollo

- **Computador:** Con procesador capaz de soportar la lectura de datos.
- **Beacon:** Dispositivo bluetooth
- **Dispositivo móvil**

#### 3.4.2. Software de Desarrollo

- **Framework de desarrollo móvil Flutter**
- **Python**

### 3.5. Entorno de soporte y desarrollo

Para el desarrollo del proyecto se evaluaron algunos frameworks para resolver el problema del cliente, para ello, se utilizaron los siguientes:

- **Python:** Desarrollo de algunos algoritmos necesarios para el desarrollo del proyecto
- **Flutter:** Framework desarrollado por Google para la creación de aplicaciones multiplataforma, utilizando el lenguaje de programación "Dart", este framework permitirá desarrollar la interfaz gráfica de la aplicación, integrar los algoritmos elaborados en Python.

### 3.6. Hitos principales del proyecto

Los hitos principales para el desarrollo del proyecto son los siguientes:

- **Hito 1 Entendimiento del desafío:** interiorizar los requerimientos del cliente y comprender lo que el cliente espera de la aplicación.
- **Hito 2 Planos procesados:** planos procesados para poder trabajar la posición correcta de los beacons.
- **Hito 3 Obtención de algoritmo:** algoritmo que permite la posición tentativa de los beacons en un plano.

- **Hito 4 Identificar posición de la persona respecto a los beacons:** ubicación de la persona por medio de detección de intensidades de los beacons.
- **Hito 5 Listado de Instrucciones:** listado con instrucciones de como llegar al destino seleccionado por el camino más corto.
- **Hito 6 Grafo sobre mapa:** implementar un algoritmo que sea capaz de generar un grafo a partir de las instrucciones y las coordenadas del usuario.
- **Hito 7 Asistente guía por medio de voz:** creación de asistente por medio de voz para poder dar indicaciones.
- **Hito 8 Creación de una aplicación movil:** aplicación movil que permite ubicar a la persona en un mapa y ser guiada por los beacons.
- **Hito 9 Viabilidad de posicionar a la aplicación en el mercado nacional.:** conocimiento de factores del mercado que influyen en el proyecto para tomar una decisión sobre si es factible entrar en un mercado con esta aplicación.

### 3.7. Estructura del sistema

#### 3.7.1. Contexto del sistema

Para la elaboración del prototipo fue necesario incorporar las necesidades del cliente por medio de un diagrama de contexto mostrado en la siguiente Figura 2 el cual ayuda a entender los límites entre el sistema y el ambiente, mostrando entidades que interactúan entre si.

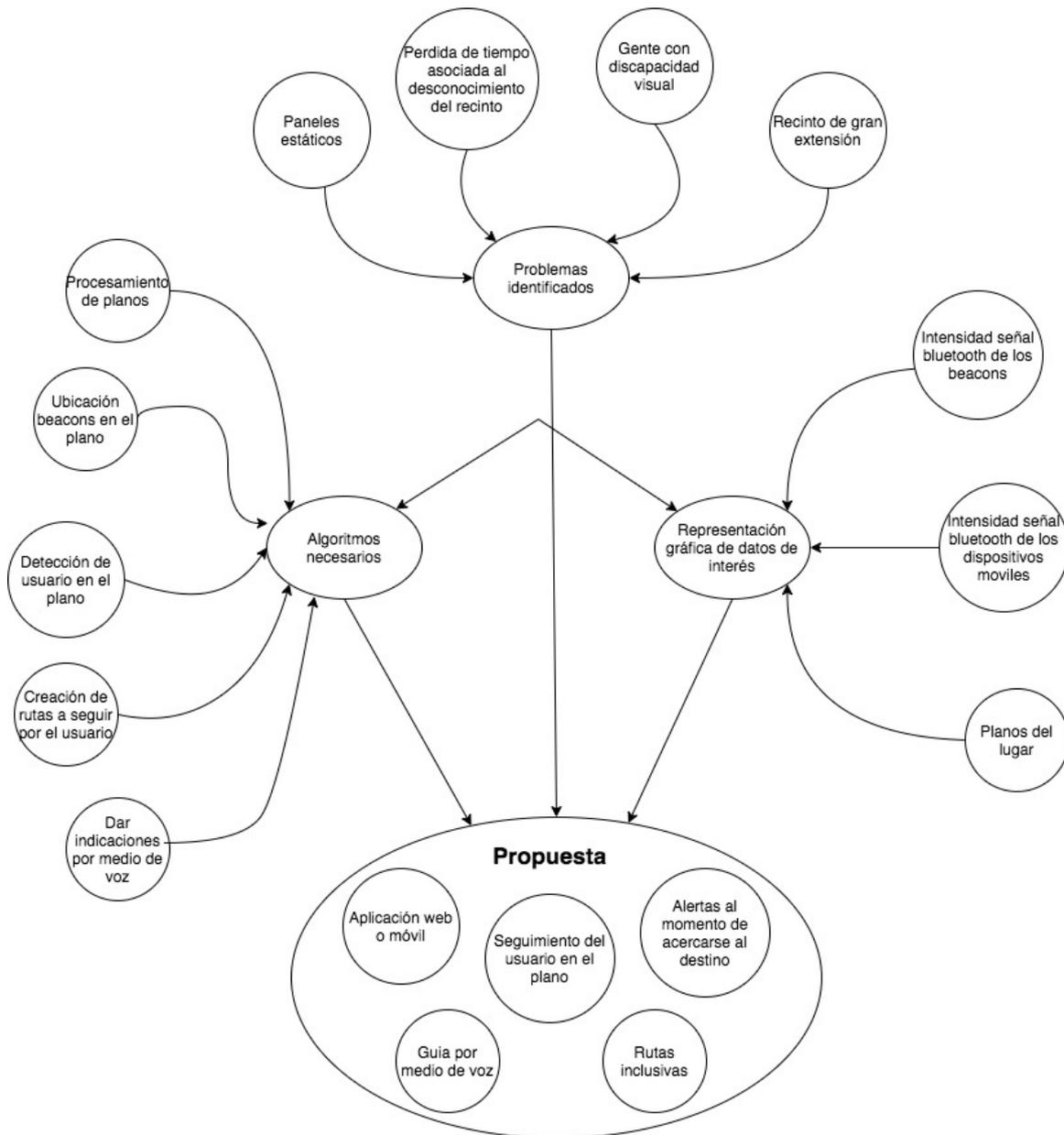


Figura 2: Diagrama de contexto Desafío 7

### 3.7.2. Esquema general del sistema

En la siguiente Figura 3 se muestra el esquema general del sistema y la interacción con los componentes.

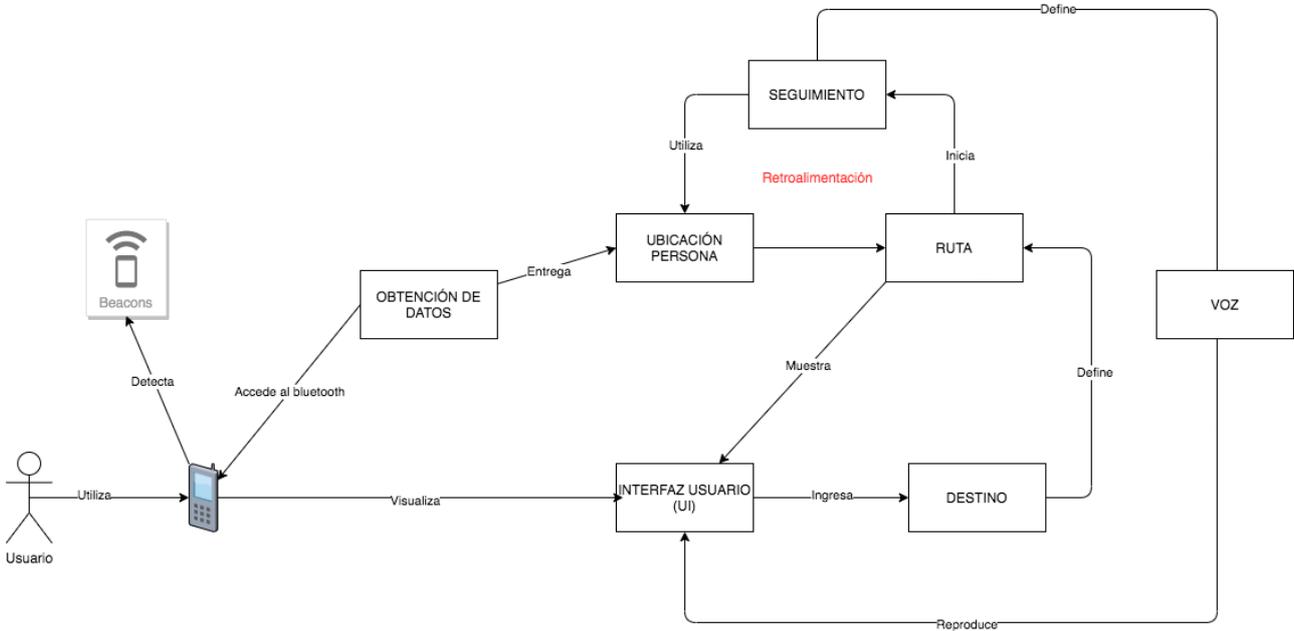


Figura 3: Esquema general del sistema

### 3.7.3. Descripción Componentes

Los componentes que describen el sistema son los siguientes:

**C1- Beacon:** tecnología utilizada para la emisión de señales RSSI.

Para empezar, los beacons que se adquirieron son dispositivos bluetooth de baja energía; estos utilizan el protocolo Eddystone que fue creado por Google. Los comprados fueron los i10 durable mostrada en la Figura 4 fabricados por MINEW [20].

La siguiente Figura muestra el dispositivo que se utilizó:



Figura 4: I10 indoor location beacon.

Eddystone tiene los siguientes marcos de trabajo:

- **Eddystone-UID:** código de identificación que permite a las aplicaciones recuperar información de los servidores. Estos se pueden utilizar para ubicación en interiores, la identificación de objetos físicos y para interactuar con las aplicaciones de cualquier forma que decida el desarrollador.
- **Eddystone-EID:** identificador rotatorio cifrado para aumentar la seguridad del protocolo, y además, actúa de manera similar a la trama UID.
- **Eddystone-TLM:** información sobre el beacon. Esto puede incluir el nivel de la batería, los datos del sensor u otra información relevante para los administradores de los beacons.
- **Eddystone-URL:** transmite una URL de 18 caracteres como máximo que redirige a un sitio web protegido mediante SSL.

Para efectos de este proyecto, solo se utiliza lo que es el UID y TLM, para obtener información relevante.

**C2- Obtención de datos:** módulo encargado de conectarse con el bluetooth del teléfono para listar las señales percibidas que correspondan al beacon y su intensidad de señal, estas son entregadas al módulo de Ubicación de persona. Estas señales son captadas sin necesidad de

conexión o emparejamiento previo con el beacon mientras el dispositivo se encuentre cercano del radio de cobertura de estos. Para este módulo, se escribió un algoritmo basado en python que es el encargado de la lectura de la señal del beacon utilizando el bluetooth del dispositivo. Los datos obtenidos del beacon son los siguientes:

- **RSSI:** intensidad de señal recibida por el dispositivo.
- **Potencia de Transmisión (Tx Power):** los beacons poseen una potencia transmisión de señal fija , lo que se han usado tienen potencia de transmisión fija de -24[dBm] puesta por el fabricante
- **UIDD:** identificador único del beacon, indica que beacon es el que se comunica.

**C3-Ubicación Usuario:** módulo que se encarga de entregar la ubicación del usuario en todo momento. Este recibe la información del teléfono con respecto a las señales recibidas de los beacons y los planos en su forma de matriz de coordenadas.

Para este módulo es necesario tener la distancia en que se encuentra el usuario de los beacons. Esto se obtiene con la información recibida del módulo anterior, en donde se tendrán los datos de los RSSI para ser integrados en la siguiente fórmula:

$$D = 10^{\left(\frac{T_x Power - RSSI}{10 * N}\right)} [m] \quad (1)$$

- **D:** distancia entre el beacon y el dispositivo en metros.
- **TxPower:** potencia de transmisión fija [dBm].
- **RSSI:** señal recibida por el dispositivo [dBm].
- **N:** pérdida del ambiente, donde en este caso toma el valor de 2 por ser espacio libre.

Esta fórmula nace a partir de la ecuación de Friis y la teoría electromagnética [17] [18].

Luego de estimar la distancia de los beacons hacia el usuario, es necesario poder localizar en que punto del mapa se encuentra este. Para poder estimarlo, se toman alrededor de 3 señales recibidas de beacons cerca y se realiza una triangulación que sirve para determinar la posición más acertada en el plano [19]. Esta ubicación obtenida se pasará al módulo de seguimiento. Para lograr la triangulación se realiza el siguiente procedimiento:

Se puede ver en la siguiente Figura 5 los beacons A, B y C a una distancia  $d_1, d_2$  y  $d_3$  (respectivamente); el usuario de la aplicación es (O). Con esta Figura se puede entender mejor las siguientes ecuaciones.

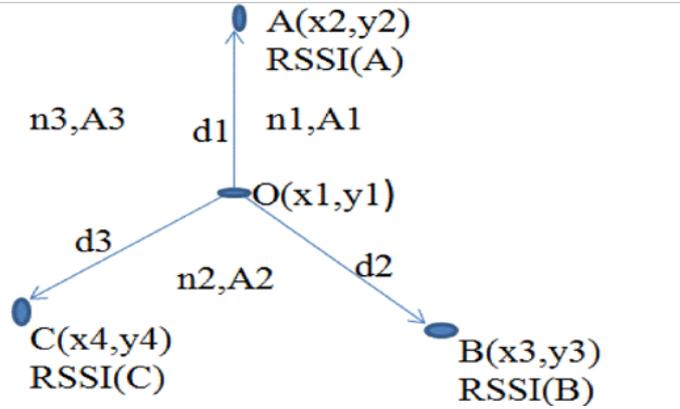


Figura 5: Esquema de triangulación.

Las distancias al usuario pueden describirse de la siguiente forma:

$$\begin{aligned} d_1^2 &= (x_1 - x_2)^2 + (y_1 - y_2)^2 [m] \\ d_2^2 &= (x_1 - x_3)^2 + (y_1 - y_3)^2 [m] \\ d_3^2 &= (x_1 - x_4)^2 + (y_1 - y_4)^2 [m] \end{aligned} \quad (2)$$

Reordenando este sistema, es posible obtener dos variables adicionales:

$$\begin{aligned} v_1 &= \frac{(d_2^2 - d_3^2) - (x_2^2 - x_3^2) - (y_2^2 - y_3^2)}{2} [m] \\ v_2 &= \frac{(d_2^2 - d_1^2) - (x_2^2 - x_1^2) - (y_2^2 - y_1^2)}{2} [m] \end{aligned} \quad (3)$$

Finalmente, las coordenadas del usuario se calculan de la siguiente forma:

$$y = \frac{v_2(x_3 - x_2) - v_1(x_1 - x_2)}{(y_1 - y_2)(x_3 - x_2) - (y_3 - y_2)(x_1 - x_2)} [m] \quad (4)$$

$$x = \frac{v_1 - y(y_3 - y_2)}{(x_3 - x_2)} [m] \quad (5)$$

Con las coordenadas obtenidas en 4 y 5 se puede ubicar al usuario en la interfaz gráfica, para esto se utilizan los planos procesados anteriormente en Python y el conjunto del sistema de coordenadas creado a partir de los planos del edificio.

**C4-Interfaz usuario:** módulo encargado de recibir parámetros del usuario y de mostrar al usuario la información proveniente de los módulos Destino, Ruta y Seguimiento.

Para el desarrollo de este módulo se utilizó el software de Flutter, el cual nos permite mostrar al usuario el mapa principal, el mapa alternativos, configuraciones y su seguimiento.

La vista primordial es la del mapa principal(Figura 30(a)). Esta vista se conforma por un widget que ejecuta las funciones principales que haciendo lo siguiente:

- **Desplegar mapa:** función que se encarga demostrar el mapa donde se encuentra el usuario, para el caso de este proyecto se muestra el mapa del edificio C, segundo piso de la Universidad Técnica Federico Santa María.
- **Pintar caminos:** función que se encarga de pintar el camino según las instrucciones entregadas.
- **Dijkstra:** algoritmo encargado de mostrar la ruta más corta tentativa.
- **A star:** algoritmo encargado de mostrar el camino correcto al momento de que el usuario se mueva por el mapa.
- **Trilateración:** función encargada de recibir los datos de las distancias entregadas por los beacons y calcular la posición aproximada del usuario.
- **Marcar el destino:** Función que recibe el destino donde quiere ir el usuario para poder marcarlo en el mapa.

Luego, la vista de seguimiento se compone también por un widget que ejecuta lo siguiente:

- **A star:** algoritmo encargado de mostrar el camino correcto al momento de que el usuario se mueva por el mapa.
- **Trilateración:** función encargada de recibir los datos de las distancias entregadas por los beacons y calcular la posición aproximada del usuario.
- **Pintar y borrar:** función que pinta la ruta que debe seguir el usuario y borra por donde el usuario estaba antes.
- **Guía de voz:** función encargada de dictar instrucciones al usuario al momento de ir caminando, como por ejemplo: "girar hacia la derecha".
- **Vibración:** función encargada de vibrar si el usuario se dirige por el camino equivocado.

Por último, la vista de mapa alternativo ejecuta lo siguiente:

- **A star:** algoritmo encargado de mostrar el camino correcto al momento de que el usuario se mueva por el mapa.
- **Trilateración:** función encargada de recibir los datos de las distancias entregadas por los beacons y calcular la posición aproximada del usuario.
- **Pintar y borrar:** función que pinta la ruta que debe seguir el usuario y borra por donde el usuario estaba antes.
- **Guía de voz:** función encargada de dictar instrucciones al usuario al momento de ir caminando, como por ejemplo: "girar hacia la derecha."<sup>en</sup> caso de ser necesario para llegar la destino.
- **Vibración:** función encargada de vibrar si el usuario se dirige por el camino equivocado.

**C5-Seguimiento:** módulo encargado de hacer un seguimiento al usuario utilizando información del módulo Ruta y Ubicación Usuario para poder llevar al usuario a su destino ingresado con anterioridad.

En primer lugar, se hace el despliegue de nodos que se muestra en la Figura 6, en esta se puede apreciar la posición tentativa de los nodos en el edificio C de la universidad. Luego, para realizar el seguimiento a tiempo real es necesario utilizar un algoritmo que vaya descubriendo el camino al momento de ir avanzando, para esto, se utilizó el algoritmo A\* (A estrella), en donde usa la combinación entre el algoritmo UCS (ordena por costo ya recorrido) y Greedy (ordena por lo que falta por recorrer). Con estos, el algoritmo A\* estima el camino total utilizando la siguiente fórmula:

$$F(s_k) = G(s_k) + H(s_k) \tag{6}$$

En donde:

- **F:** ruta máxima
- **G:** función que estima los costos de las movidas realizadas
- **H:** función que estima el costo estimado del mejor camino

A\* es optimista pues considera que el costo para resolver un problema siempre es inferior. Además, utiliza una heurística de apoyo llamada "distancia de manhattan", esta no solo elige el vecino con el menor costo de distanciamiento para moverse, si no que le dice cual de los vecinos esta más cerca del nodo de destino para indicarle a A\* por donde seguir.

Para el proyecto, A\* tendrá una cola de prioridad que elige los posibles nodos que son explorables que están cerca del nodo destino.

**C6-Ruta:** módulo encargado de entregar la ruta que el usuario debe seguir al módulo de Interfaz de Usuario y al de Seguimiento.

Para realizar la ruta es necesario establecer una red en cada mapa a trabajar para poder guiarlos.

En la figura 6 se muestran los nodos que se desplegaran a manera de simulación y pruebas.



Figura 6: Nodos edificio C.

Si el usuario solo desea ver la ruta tentativa se utiliza Dijkstra, este muestra el camino más corto sin realizar el seguimiento.

**C7-Destino:** módulo donde el usuario selecciona el destino al que quiere ir. El destino seleccionado es entregado al módulo de Ruta y a su vez es mostrado en el módulo de Interfaz Usuario.

Para seleccionar destino se despliegan opciones donde el usuario puede moverse dentro del mapa.

Debido a que el proyecto es un prototipo, solo se mostrarán algunas salas del edificio C del segundo piso. Estas salas son las que se encuentran ubicadas en los nodos rojos vistos en la Figura 6.

**C8-Voz:** módulo encargado de enviar las instrucciones claves por medio de voz al Módulo de Interfaz de Usuario.

La función que se escribió para este módulo, consiste en tomar un diccionario con las siguientes instrucciones:

- **Gire a la derecha**
- **Gire a la izquierda**
- **Dar media vuelta**
- **Camine hacia adelante**
- **Haz llegado a tu destino**

Este diccionario es leído por la función de flutter "TEXT TO SPEECH".

Se ejecuta el código según la ubicación del usuario al momento de solicitar el seguimiento y caminar.

### 3.8. Diseño de interfaces

#### 3.8.1. Modelo de navegación

En la Figura 7 se puede apreciar el modelo de navegación de la aplicación, este muestra las posibles interacciones y flujos que puede tener la aplicación.

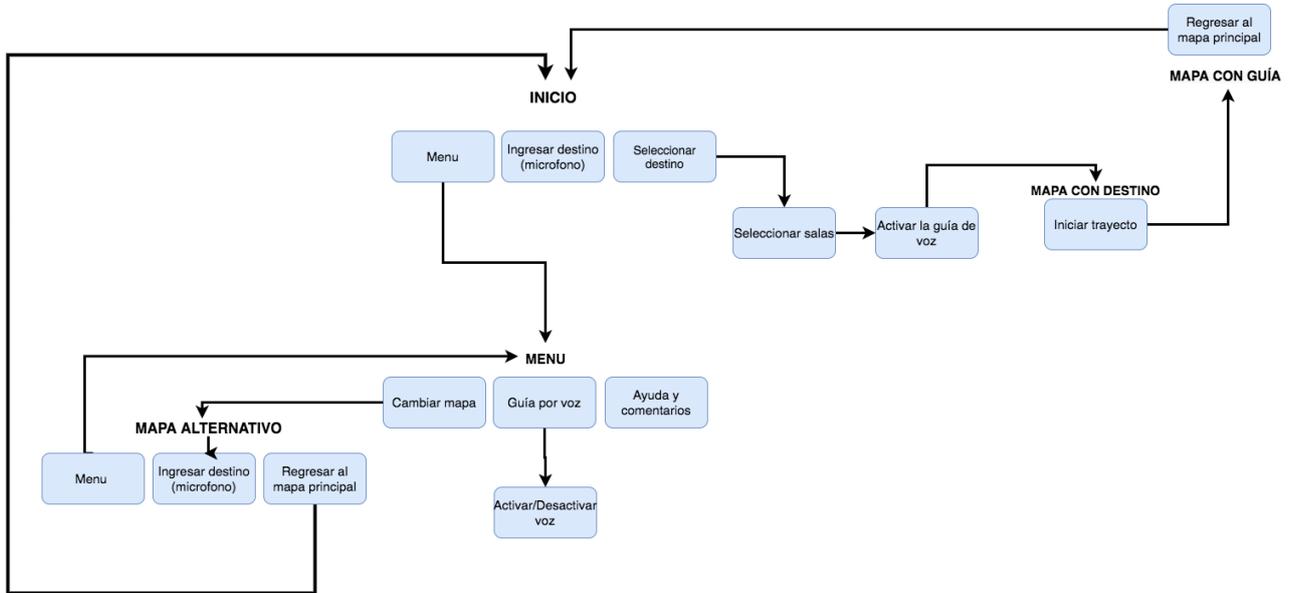


Figura 7: Modelo de navegación.

#### 3.8.2. Diseño preliminar de las vistas

Para el diseño preliminar del software que se estableció, fue necesario investigar y esclarecer las necesidades del cliente y los usuarios. Luego, ya establecidos los requisitos que se necesitaban para resolver el problema del cliente, se procedió a realizar una primera propuesta de la interfaz de usuario. A continuación se presentan el primer diseño requerido por el cliente.

Las siguientes Figuras 8(a) y 8(b) muestran el inicio del mapa principal y el menú de la aplicación dentro del mapa principal.



Figura 8: Izquierda: Inicio de la aplicación, Derecha: Menú de la aplicación.

En las siguientes Figuras 9(a) y 9(b) se puede apreciar lo que es la selección del destino y la ruta tentativa a seguir.



Figura 9: Izquierda: Selección de destino, Derecha: Ruta tentativa.

Luego, en estas Figuras 10(a) y 10(b), se muestra lo que es el inicio del seguimiento y la llegada al destino del usuario



Figura 10: Izquierda: Seguimiento del usuario, Derecha: Aviso de llega a destino.

En las Figuras 11(a) y 11(b) se puede apreciar la vista del mapa alternativo y confirmación.



Figura 11: Izquierda: Mapa alternativo, Derecha: Confirmación de la sala.

Finalmente, se muestra en las siguientes Figuras 12(a) y 12(b) las instrucciones que se presentaran en la vista del mapa alternativo.



Figura 12: Izquierda: Indicaciones, Derecha: Llegada a destino.

## 4. CAPÍTULO 3: RESULTADOS

### 4.1. Datos obtenidos : beacons

El dato que se obtiene es la potencia de transmisión que el beacon emite dentro de su radio de acción. Estos son de suma importancia para deducir la ubicación del usuario en el plano donde se encuentra.

La ubicación del usuario es el resultado de 2 etapas:

- **Tratamiento de los datos obtenidos: distancia**
- **Implementar triangulación: posición usuario**

#### 4.1.1. Tratamiento de datos obtenidos: distancia

En primer lugar, el beacon entrega diferentes datos, entre estos están los mencionados con anterioridad:

- **Potencia de Transmisión (Tx Power)**
- **UIDD.**

Para efectos de esta memoria, se utilizó el lenguaje de programación Python que ayuda a obtener la potencia de transmisión por medio de un código que se encarga de abrir los puertos necesarios y hacer una lectura de la señal que emiten los beacons. Luego de hacer la lectura y obtener la potencia de transmisión se introduce en otro algoritmo que calcula la distancia utilizando la ecuación (1 para poder estimar los metros a los que se encuentra el beacon en ese momento.

El obtener la distancia del usuario al beacon es de suma utilidad debido a que con este se puede calcular la posición aproximada donde se encuentra el usuario en el plano.

Las UIDD de los beacons fueron:

- **UIDD 1:** AC233F8A804A
- **UIDD 2:** AC233F8A8044
- **UIDD 3:** AC233F8A8045

A continuación se presentan los beacons utilizados para las mediciones con sus respectivos UIDD:



Figura 13: Beacon con UIDD1: AC233F8A8804A



Figura 14: Beacon con UIDD2: AC233F8A88044



Figura 15: Beacon con UIDD3: AC233F8A88045

En la siguiente imagen se puede apreciar los valores entregados de cada beacon al momento de realizar el escaneo con Python. Este entrega valores importantes como el UIDD en formato ac:23:3f:8a:80:XX, Tx Power [dBm]. En el proceso de escaneo, se incorporó el algoritmo de distancia [m] ingresando Tx Power a la ecuación 1, eso se puede evidenciar en la primera línea mostrada que hace referencia a : **la distancia del beacon ac:23:3f:8a:80:XX es: x[m]** Además se puede apreciar, que aparece el URL mencionado con anterioridad, dirigido al sitio web del fabricante, Minew.

```
<ac:23:3f:8a:80:45, -67> EddystoneURLFrame<tx_power: -24, url: https://www.minew.com/>
la distancia del beacon ac:23:3f:8a:80:45 es : 2.4633797972265947
<ac:23:3f:8a:80:45, -73> EddystoneURLFrame<tx_power: -24, url: https://www.minew.com/>
'}
la distancia del beacon ac:23:3f:8a:80:45 es : 4.771839267923697
<ac:23:3f:8a:80:4a, -75> EddystoneURLFrame<tx_power: -24, url: https://www.minew.com/>
'}
la distancia del beacon ac:23:3f:8a:80:4a es : 5.877344214222982
<ac:23:3f:8a:80:45, -66> EddystoneURLFrame<tx_power: -24, url: https://www.minew.com/>
'}
la distancia del beacon ac:23:3f:8a:80:45 es : 2.1937344757933217
<ac:23:3f:8a:80:4a, -75> EddystoneURLFrame<tx_power: -24, url: https://www.minew.com/>
'}
la distancia del beacon ac:23:3f:8a:80:4a es : 5.877344214222982
<ac:23:3f:8a:80:45, -68> EddystoneURLFrame<tx_power: -24, url: https://www.minew.com/>
'}
la distancia del beacon ac:23:3f:8a:80:45 es : 2.761423613067043
<ac:23:3f:8a:80:4a, -75> EddystoneURLFrame<tx_power: -24, url: https://www.minew.com/>
'}
la distancia del beacon ac:23:3f:8a:80:4a es : 5.877344214222982
<ac:23:3f:8a:80:45, -73> EddystoneURLFrame<tx_power: -24, url: https://www.minew.com/>
'}
```

Figura 16: Valores en consola

Para poder guardar los datos se utilizó un archivo con formato csv en donde se recopi-

laba la potencia de transmisión y la distancia del beacon, esto se puede apreciar de mejor manera mirando la Figura 33. Cabe destacar que se hicieron las mediciones con los 3 beacons mencionados, los cuales fueron posicionados como se muestra en la Figura 32, para luego simular el caminar ideal del usuario mientras las señales eran escaneadas por el computador y guardandolas en el archivo dependiendo del UUID.

#### 4.1.2. Implementar triangulación: posición usuario

Luego de obtener la distancia, es necesario que sus señales se superpongan. Esta superposición se puede apreciar en la Figura 17 la interacción de 3 beacons con sus respectivas señales (círculos hechos con líneas punteadas) :

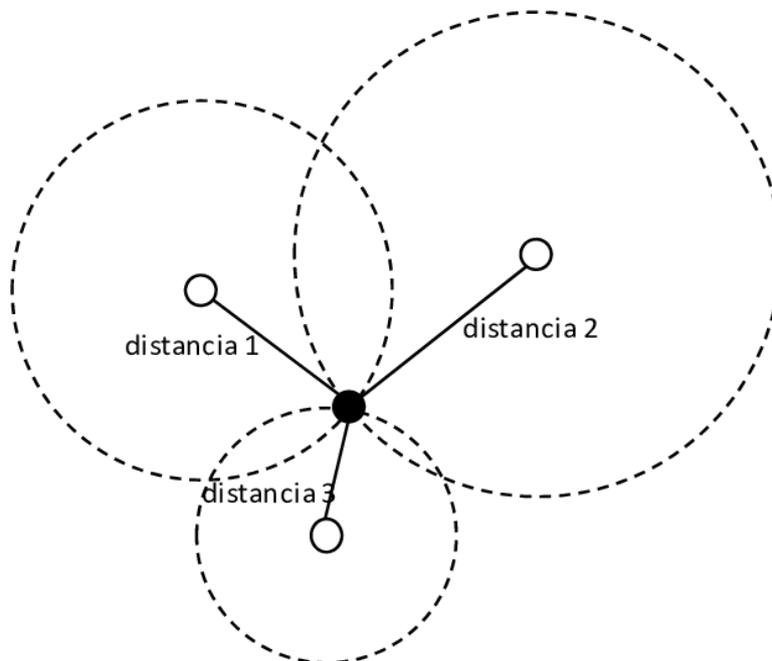


Figura 17: Tres beacons y su respectiva distancia al usuario.

Al tener el esquema de esa forma, podremos detectar la posición de usuario utilizando la ecuación 2 para despejarla y obtener la posición en X e Y mostrada en las ecuaciones 5 y 4 respectivamente.

Estas ecuaciones se programaron en lenguaje Python donde se crearon funciones necesarias para poder determinar el punto (X,Y) del usuario en cuestión. Luego, las funciones fueron migradas a Dart, el lenguaje de programación que utiliza Flutter para ser ocupadas posteriormente en la simulación.

## 4.2. Funciones principales de la aplicación

Para la realización del prototipo fue necesario definir las funciones principales que tendría la aplicación para cumplir el seguimiento, detección de ruta y asistente de voz.

Cada vista en la aplicación tiene sus propias funciones y además comparte algunas con otras vistas.

A continuación se verán las vistas principales de la aplicación:

En la siguiente Figura 18 se muestran las 3 vistas principales de la aplicación, las cuales corresponden a las creadas para el prototipo.

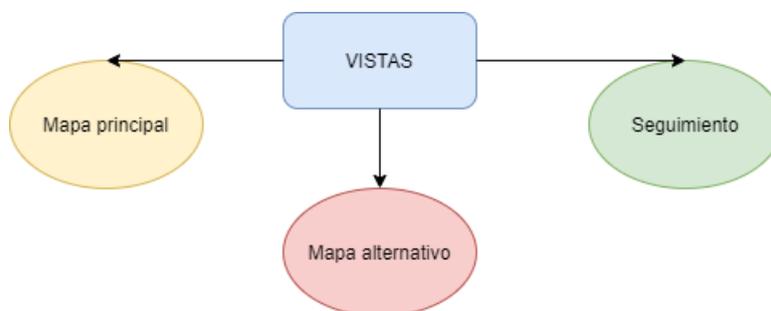


Figura 18: Vistas funcionales de la aplicación

■ Las funciones asociadas al mapa principal son las siguientes:

Se puede apreciar en la figura 19 que existen 6 funciones principales que interactúan con el mapa. También se puede apreciar cuales son los parámetros que recibe cada función.

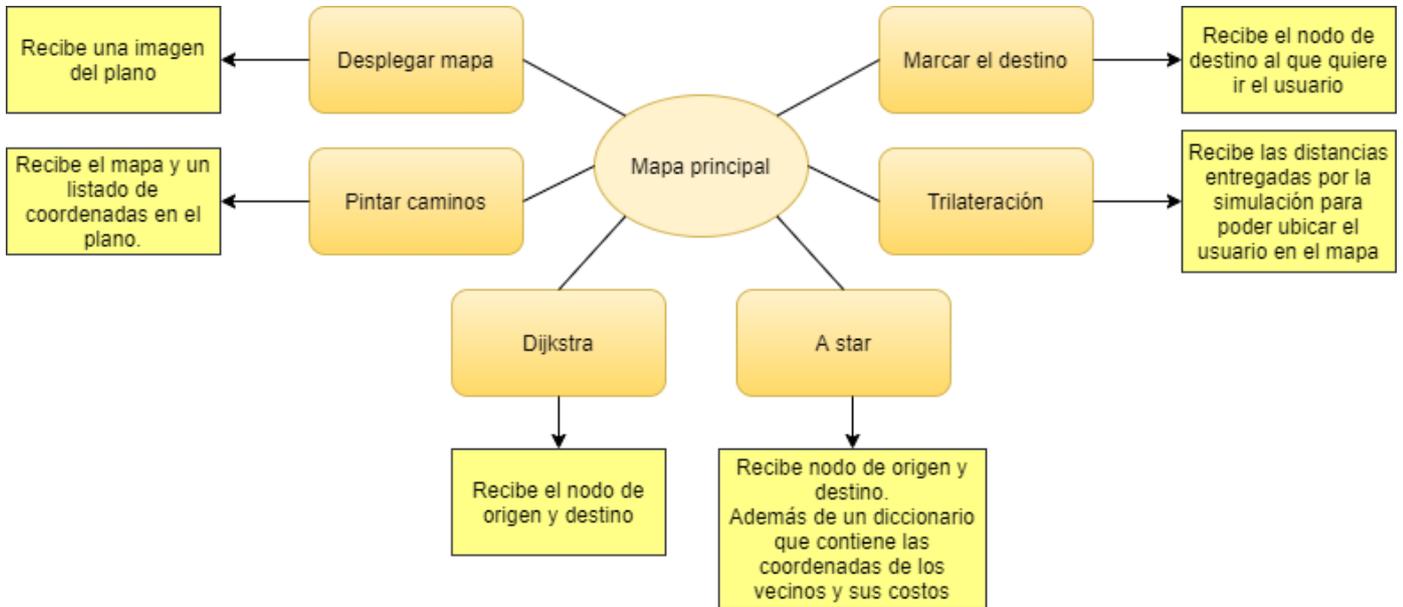


Figura 19: Funciones del mapa principal

Las funciones asociadas al mapa alternativo son las siguientes:

Posteriormente, se puede apreciar en la figura 20 las funciones principales que interactúan con esta vista, donde también se muestran los parámetros que recibe cada función.

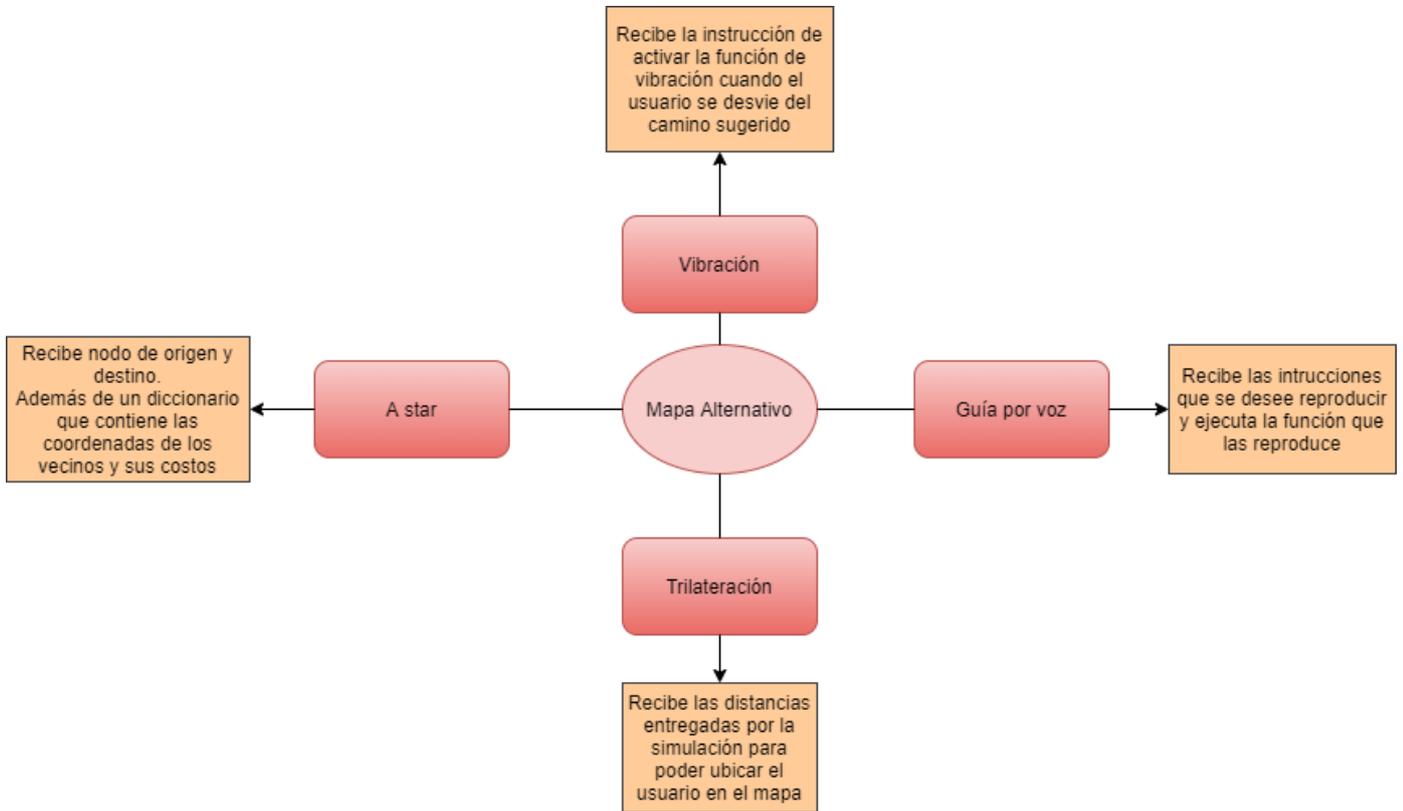


Figura 20: Funciones del mapa alternativo

Las funciones asociadas a la vista de seguimiento son las siguientes:

Finalmente, se puede apreciar en la figura 21 las funciones principales que interactúan con esta vista, donde también se muestran los parámetros que recibe cada función.

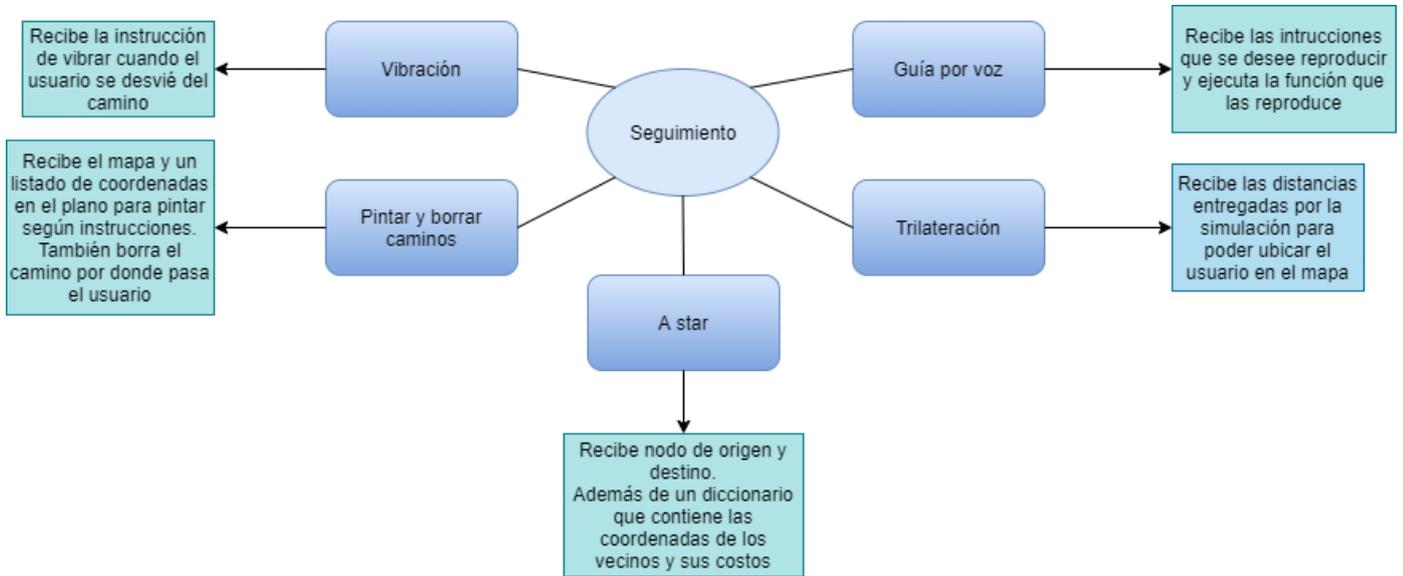


Figura 21: Funciones de vista de seguimiento

### 4.3. Verificación

Para la verificación del prototipo se realizó una simulación debido a que las condiciones sanitarias no permitieron poder desarrollar el proyecto dentro de la universidad, sumado a que los beacon llegaron con un mes antes de la entrega del prototipo funcional. Se validó la propuesta a nivel de prototipo funcional con sus componentes visuales y funcionalidades esperadas respecto a lo solicitado por el cliente.

### 4.4. Caso de uso

En primer lugar, el usuario ingresa a la aplicación y podrá observar el mapa del lugar donde se encuentra (en este caso, edificio C segundo piso) accediendo a las siguientes funciones:

- **Ingresar a la aplicación:** al momento que el usuario ingrese a la aplicación, se mostrará el mapa principal que muestra la Figura 30(a).
- **Seleccionar destino:** existen diversas rutas que el usuario puede seleccionar, en la siguiente Figura 22(b) se muestran alguna de estas rutas, para esto, el usuario debe pre-

sionar el cuadrado que posee la lupa (Figura 22(a)) y se desplegarán los destinos a los cuales el usuario puede moverse en el plano como se muestra a continuación:

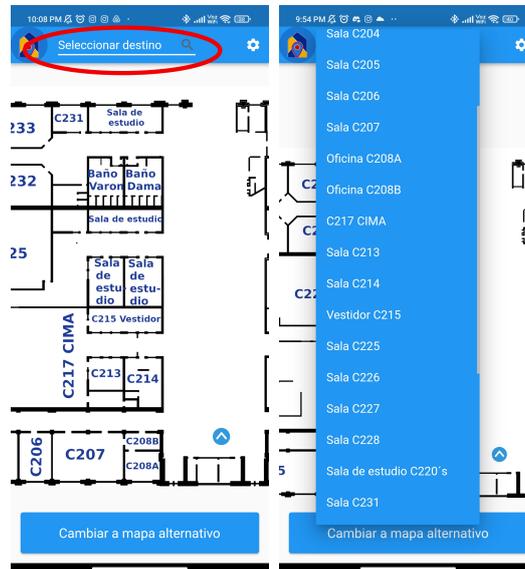


Figura 22: Selección de destino.

- **Mostrar ruta tentativa:** luego de seleccionar el destino, se mostrará una ruta tentativa y aparecerá un botón azul, al costado derecho inferior, este permitirá realizarle un seguimiento al usuario. En la siguiente Figura 23 se muestra la ruta tentativa que le marca al usuario al momento de ingresar el destino **Sala C225**:



Figura 23: Ruta tentativa después de seleccionar destino

- **Iniciar seguimiento:** posterior a seleccionar el destino y marcar la ruta tentativa, el usuario presiona el botón azul que se encuentra en la parte derecha inferior de la

aplicación (Figura 24) y se iniciará el seguimiento al usuario. En la siguiente Figura se muestra el seguimiento del usuario en una ruta.



Figura 24: Botón que inicia el seguimiento del usuario.

En las siguientes Figuras 25(a) y 25(b) se puede apreciar el movimiento de usuario hacia el destino.



Figura 25: Vista de seguimiento al usuario.

Se puede ver en la siguiente Figura 26 el momento en que el usuario llega a su destino y la aplicación lo detecta, esto muestra una ventana emergente que le indicará al usuario que ha llegado a su destino.



Figura 26: Ventana emergente que se muestra al usuario que llegó a su destino.

- **Cambiar a mapa alternativo:** el mapa alternativo es un mapa que proporciona solo instrucciones por medio de la voz y lectura de la pantalla. A continuación se muestra en la Figura 27(a) como acceder a el:

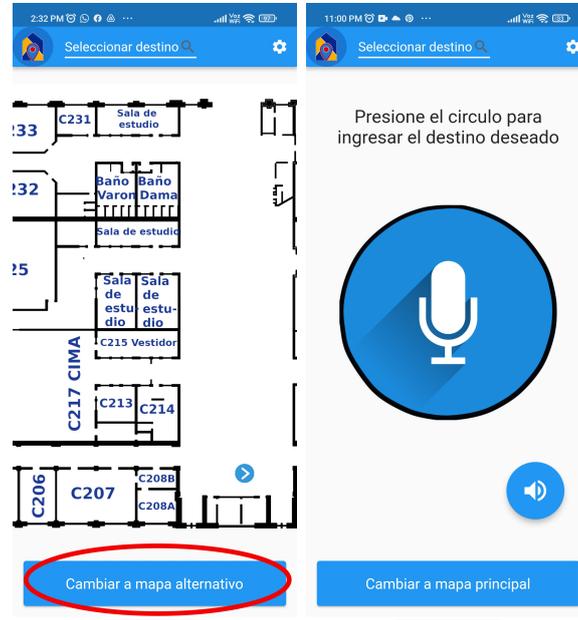


Figura 27: Izquierda: mapa principal, Derecha: mapa alternativo.

- **Configuraciones:** para ingresar a configuraciones, es necesario presionar el icono que tiene forma de una tuerca, como se muestra en la siguiente Figura 28(a):

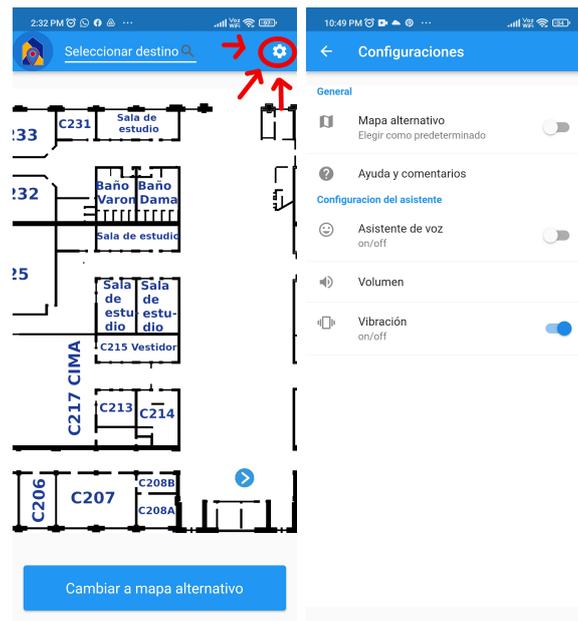


Figura 28: Izquierda: seleccionar configuraciones, Derecha: configuraciones.

Luego, el usuario puede seleccionar mapa alternativo, esta vista aún no tiene implementada todas las funciones necesarias debido a que para la simulación se dio énfasis en la mapa principal del edificio C.

- **Ingresar al mapa alternativo:** se mostrará el mapa alternativo mostrado en la Figura 27(b).
- **Seleccionar destino:** para el caso del mapa alternativo, es necesario activar el micrófono que recibirá la instrucción del usuario.
- **Cambiar a mapa principal:** el usuario puede regresar al mapa principal presionando el botón que se muestra en la Figura 29(a) :

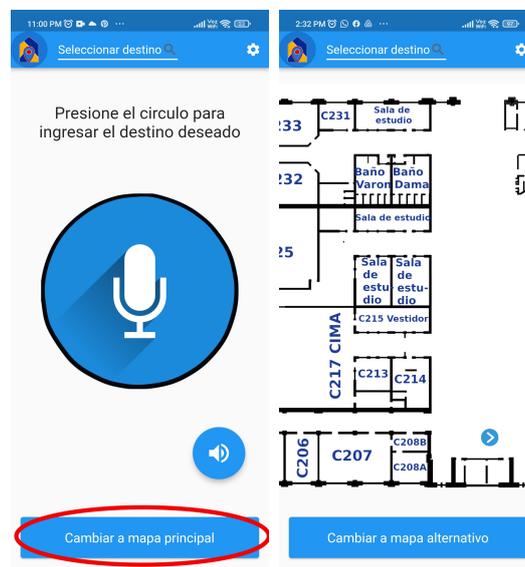


Figura 29: Izquierda: mapa alternativo, Derecha: mapa principal.

- **Configuraciones:** el menú de configuraciones le permite al usuario personalizar el uso de la aplicación, pudiendo seleccionar el mapa principal, el volumen del asistente de voz, la vibración y si desea desactivar o activar el asistente de voz. Para ingresar a configuraciones, es necesario presionar el icono que tiene forma de una tuerca, como se muestra en la siguiente Figura:

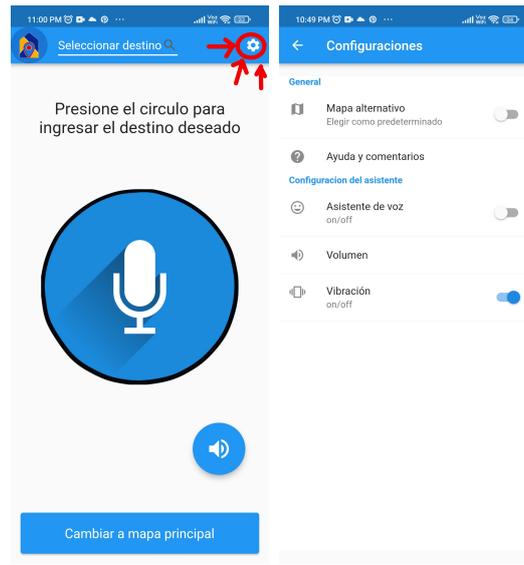


Figura 30: Izquierda: mapa alternativo, Derecha: menú configuraciones.

## 4.5. Simulación del prototipo

Para ejecutar la simulación fue necesario incorporar distintos algoritmos. A continuación se hablará un poco de los algoritmos utilizados, tratamiento de datos y las consideraciones que se tuvieron al momento de realizar dicha simulación.

### 4.5.1. Posicionamiento de los beacons

En primera instancia, hay que posicionar los beacons en el plano, para eso, se realizó una planificación de la red, utilizando la heurística (ad-hoc) explicada a continuación:

- **Entradas:** plano USM para convertirlo en matriz de posiciones válidas en formato .csv
- **Salida:** matriz de cobertura en formato .csv; esto muestra el plano resultante de la posible posición de los beacons a utilizar, además de mostrar el tiempo de demora del algoritmo y porcentaje de cobertura en el plano
- **R:** radio de cobertura de un beacon: 100[m]
- **Atenuación:** el radio de cobertura del beacon cuando toca una pared u obstáculo se reduce en tres metros
- **Distancia y cobertura mínima:** separación entre beacons  $R/2$ . Cada punto de la matriz que se ingresa debe estar cubierto por al menos dos beacons

También se tiene en cuenta los siguientes datos:

- **Beacon Padre:** es el beacon que da inicio a un ciclo
- **Beacon Hijo:** son los beacons que se colocan en un ciclo
- **Mapa a utilizar tiene que tener una S marcada para poder inicializar el algoritmo**

Se tienen los siguientes parámetros a considerar en el código:

- **MP:** matriz de posiciones válidas. (vacío si es valida, 0 si no y B si es muralla/bloqueo)
- **MC:** matriz de cobertura (número que representa la cantidad de sensores que cubren dicha celda, -1 si no es una zona que deba tener cobertura)
- **partida:** corresponde a la celda marcada con una S en el plano

En este algoritmo se tiene un beacon padre que se coloca en el mapa; para colocar al hijo se recorren las celdas cubiertas por el padre que estén a una distancia mayor a  $R/2$  y se selecciona aquella en la que al ubicar el beacon hijo se cubra la mayor cantidad de celdas posibles. Esto para dar prioridad a las celdas del padre que tienen cobertura 1 debido a que pueden haber celdas con dos o más y esto indica que hay otros beacons cubriendo esa celda. Este proceso ocurre recursivamente hasta que los beacons hijos cubren el 95% del beacon padre.

Todos los hijos que se colocaron pasan a ser padres para los futuros beacons. Después de este proceso se obtiene como salida la matriz de cobertura en formato .csv donde deben ser colocados los beacons para maximizar lo que cubrirá el radio del dispositivo en el plano. La siguiente Figura 31 muestra como deberían estar ubicados los beacons según el resultado del algoritmo mencionado.

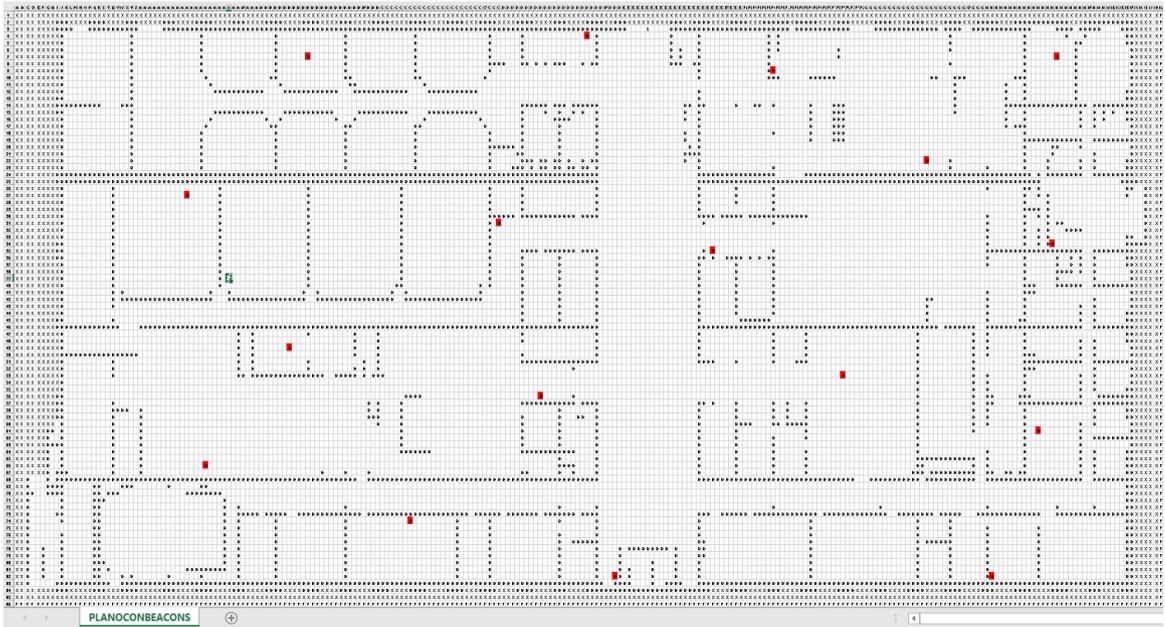


Figura 31: Posicionamiento de los beacons en el plano.

Debido a que no se pudo hacer pruebas con la posición entregada por el algoritmo, se colocaron los beacons en otra posición para lograr hacer la simulación. Estos se colocaron en línea vista (sin obstáculos como murallas) dentro del mismo plano (edificio C, segundo piso). Se colocaron en el hall del edificio C; en la siguiente Figura 32 se muestra la posición de los beacons con la que se realizó la simulación, además, los círculos rojos representa la posición de los beacons y el círculo azul representa al usuario.

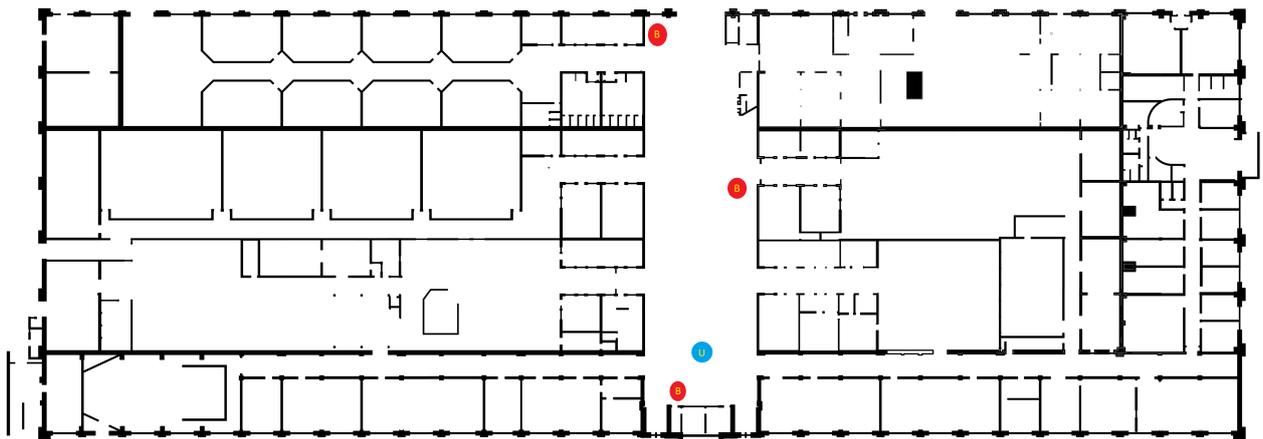


Figura 32: Posicionamiento línea vista de los beacons y el usuario en el plano.

Existe un archivo con formato.csv que contiene el id del beacon y la coordenada en donde se encuentra en el plano.

Luego de haber procesado y tratado el plano, es necesario pasar las coordenadas del excel a píxeles en una imagen, para esto, se hizo una función encargada de leer el archivo con formato .csv que transformaba e indicaba la posición de los beacons en la imagen del plano.

Hay que tener en cuenta que la posición de los beacons que se ubicaron en píxeles dentro del plano son los mostrados en la Figura 32 que se vio con anterioridad.

#### 4.5.2. Señales recibidas y tratadas

Para recibir las señales emitidas por los beacons se programó un algoritmo en Python que las captaba como lo haría un dispositivo móvil. Se ejecutaron algunas pruebas dentro de la universidad, haciendo una simulación del movimiento ideal de un usuario por unas ruta determinadas y así, analizar como variaba la intensidad de la señal si los beacons estuvieran en línea vista con el usuario (Figura 32) para luego sacar datos de la simulación.

Se almacenaron los datos de las señales y su transformación a distancia en un excel para ser utilizados en la simulación.

A continuación, en la siguiente Figura 33 se muestra algunos de los valores obtenidos de un beacon al momento de capturar los datos en el edificio C, se denota que los datos de la izquierda son la señal RSSI y la derecha es su transformación a distancia.

A	B
-64.46	1.82868142
-64.45	1.82649557
-64.44	1.82431199
-64.43	1.82213068
-64.42	1.81995164
-64.41	1.81777487
-64.4	1.81560036
-64.39	1.81342812
-64.38	1.81125814
-64.37	1.80909043
-64.36	1.80692497
-64.35	1.80476176
-64.34	1.80260082
-64.33	1.80044212
-64.32	1.79828567
-64.31	1.79613148
-64.3	1.79397953
-64.29	1.79182982
-64.28	1.78968235
-64.27	1.78753713
-64.26	1.78539414

tablaRSSI

Figura 33: Izquierda son los datos de la señal y derecha la conversión a distancia

Hay que recalcar que el error aproximado que entregó la captura de los datos fue de un metro en cada beacon probado.

### 4.5.3. Distancia y trilateración

Luego de haber obtenido los datos mostrados en la Figura 33 es necesario realizar la trilateración para ubicar el usuario en el mapa; por lo que se escribió un algoritmo que ejecuta los cálculos matemáticos explicados con anterioridad.

Seguido, se toma los resultados entregados por el excel de la Figura 33, se ingresa cada distancia de los tres beacons a la función de trilateración y esta entrega las coordenadas X,Y del usuario con respecto al mapa; luego esas coordenadas son tratadas y convertidas en píxeles sobre la imagen plano del edificio C. Posteriormente, las coordenadas del usuario se ingresan a un set state en el cuerpo del mapa que es un stack en donde se genera un widget que se llama **posición** que coloca el usuario en el mapa.

### 4.5.4. Creación de nodos en el plano

Como se aprecia en la Figura 6, se tiene la posición de los nodos en el edificio C, sector izquierdo, que es donde se concentra la mayor cantidad de salas utilizadas por los usuarios. Para poder ubicar esos nodos en la imagen de la aplicación, es necesario pasarlos a un excel que contenga los nodos descritos, dando como resultado la siguiente Figura 34 que muestra el archivo en formato .csv de los nodos colocados tal como se vio en la Figura 6 :

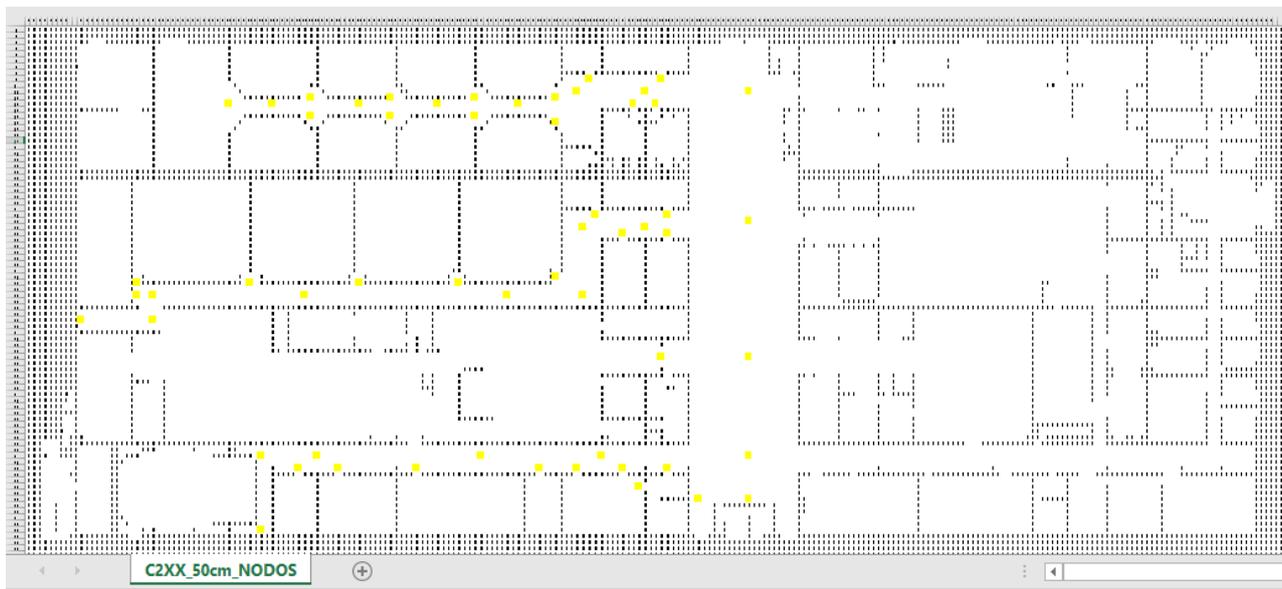


Figura 34: Nodos .csv del edificio C.

Finalmente, este archivo es procesado a píxeles para poder ubicar a los nodos en la imagen de la aplicación.

#### 4.5.5. Creación de algoritmos de rutas más cortas: dijkstra y astar (A\*)

Para empezar, el usuario ingresa el destino al que quiere ir y gracias al algoritmo de Dijkstra se le muestra una ruta tentativa creada por los nodos vistos en la Figura 6 .

Dijkstra es un algoritmo que muestra el camino más corto; este puede explorar todos los nodos que van del vértice (inicio) hasta el destino y además entrega la combinación de nodos necesaria para llegar al destino.

Estas rutas se dibujan sobre los nodos dentro de la imagen en Flutter. Se utiliza la función **Pintar** que dibuja el camino en el mapa que Dijkstra determinó como la ruta más corta. En la siguiente Figura 37 se puede apreciar el camino rojo que determina la ruta más corta:



Figura 35: Ruta tentativa después de seleccionar destino

Continuando, para la vista del seguimiento se utilizó lo que es el algoritmo de A\*, este usa la combinación entre el algoritmo UCS (ordena por costo ya recorrido) y Greedy (ordena por lo que falta por recorrer). Con esto, el algoritmo A\* estima el camino total utilizando la fórmula 6 vista anteriormente. Agregando, este algoritmo se aplica en el mapa creando una ruta dinámica utilizando la ya mencionada heurística de apoyo llamada **distancia de manhattan**.

Para comenzar a realizar A\* se tiene que leer el mapa en formato .csv debido a que este posee la información de las celdas habilitadas para que el usuario transite. Se tiene que los B y X dentro del csv son bloqueos, por ende, el usuario no puede transitar por allí. Mientras que los espacios vacíos son los lugares dentro del mapa donde el usuario puede caminar.

Esto puede apreciarse mejor en la siguiente Figura 36 en donde se marcan las X y las B para formar lo que sería el pasillo:

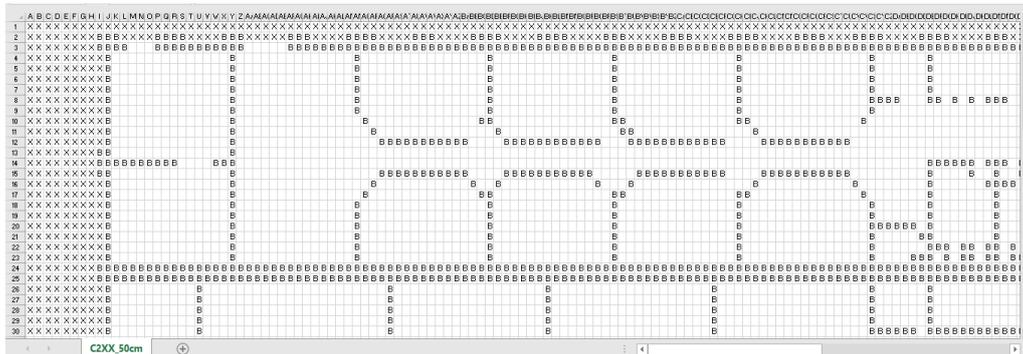


Figura 36: Parte del 3er pasillo del plano del edificio C, 2do piso.

Por lo anterior mencionado, es necesario poder saber en que posiciones puede moverse el usuario para poder crearle una ruta dinámica, así que, se decidió crear un algoritmo en python que convierta la información del archivo con extensión .csv en un archivo dart en donde existe una variable que posee lista de filas y columnas que pertenecían al csv.

Posteriormente, se hizo otra función que recibe una coordenada, verifica las 8 casillas adyacentes a la coordenada que recibió y entrega las celdas validas en donde se puede mover el usuario y el costo de moverse hacia estas; se definió además que el costo horizontal y vertical tiene el valor de 1 y diagonal es 2.

Cuando ya se tiene las celdas validas, el costo y los vecinos, se procede a realizar el algoritmo de A\* para crear la ruta dinámica del usuario.

Teniendo las funciones mencionadas con anterioridad, se puede explorar los vecinos adyacentes a donde se encuentra el usuario en ese momento, para esto, se calculan los datos necesarios (F,G,H) de la ecuación (6) ya mencionada. Al realizar el cálculo de F se obtiene el valor que tendrá dentro de la cola de prioridad que elige los posibles nodos que son explorables que están cerca del destino. Además, la cola de prioridad saca el nodo que tiene la prioridad más baja y este nodo pasa a ser parte de los nodos explorados dentro de otra lista.

Este proceso se realiza iterativamente hasta que se encuentra el nodo destino(este se debe conocer con anterioridad).

En la siguiente Figura, se puede apreciar el mapa con la vista de seguimiento en donde se le esta aplicando A\* para poder seguir el usuario:

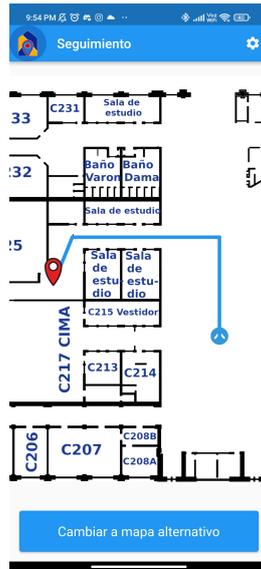


Figura 37: Ruta tentativa después de seleccionar destino

#### 4.5.6. Eventos de tiempo

Se tomaron mediciones en la universidad simulando el movimiento ideal del usuario, además de medir los segundos y la intensidad recibida de los beacons ubicados en línea vista como anteriormente se mencionó. Posteriormente, estas mediciones se pasaron a un archivo de formato .json que contiene el tiempo en segundos y el listado de las intensidades recibidas de los 3 beacons. La vista del seguimiento contiene una función que saca los datos del archivo .json y convierte las señales de los 3 beacons en distancia, para luego ubicar al usuario con la función de trilateración mencionada anteriormente.

Con estos eventos de tiempo, se puede realizar la simulación de un usuario caminando por una ruta predeterminada.

### 4.6. Análisis crítico y evaluación de riesgos

#### 4.6.1. Análisis crítico

- **Llegada tardía de los beacons:** debido a la demora que existió con respecto a la recepción de los beacons, se tiene poco tiempo para poder ser probados en su totalidad y conectarlos correctamente.
- **Problemas con el primer software de diseño:** se empezó a utilizar kivy en primera instancia, pero debido a que es un software que lleva poco tiempo y con poca información

de la comunidad, se optó por cambiarse a Flutter luego de haber intentado con Kivy aproximadamente 2 semanas.

- **Pruebas del software:** debido al retraso por el cambio de software se vuelve más complicado el poder probarlo con los usuarios en la universidad.

#### 4.6.2. Riesgos

Los riesgos que presenta el proyecto son los siguientes

1. Falta de disponibilidad de recursos de hardware para la creación del sistema
2. Beacons comprados son difíciles de configurar
3. Problemas en la implementación de un mapa con seguimiento sin utilizar APIs existentes (como GoogleMaps)
4. Retraso en plazos propuestos por cambio de software
5. Recepción del sistema por parte de los usuarios:
  - No agrado por la usabilidad, los potenciales usuarios no se sienten cómodos utilizando la aplicación
  - No agrado por diseño, los potenciales usuarios no se sienten cómodos con la aplicación
  - Usuarios que deseen usar la aplicación pero no la entiendan al tener poca cercanía con la tecnología

## 5. CAPÍTULO 4: CONCLUSIONES

En este documento se realizó una propuesta de solución para diferentes problemas que se viven en la Universidad Técnica Federico Santa María, como lo es el desorientarse y necesitar ayuda de terceros para solucionarlo. Además, se presento un prototipo funcional, que busca poder suplir la ayuda de un tercero al momento de que la persona ingrese a la universidad, específicamente al edificio C y también, que facilite a las personas con discapacidad visual a encontrar sus destinos sin depender de nadie para lograrlo.

Con esto, se busca mejorar la experiencia del usuario dando alternativas de mapas para su uso, en donde se encuentran dos tipos, los cuales son, el mapa principal que muestra un plano del sector donde se encuentra el usuario (edificio C) y un mapa alternativo que entrega la información más resumida y amigable para los usuarios que son ajenos al a tecnología o simplemente no necesitan utilizar el mapa principal. Estos mapas utilizan algoritmos especializados en encontrar caminos en el menor tiempo posible, minimizando así la duración de los traslados y encontrando a tiempo la sala requerida.

Uno de los próximos desafíos es poder lograr la conectividad con la tecnología beacon, dado que, como se pudo apreciar con anterioridad, se realizó una simulación de como funcionaria la aplicación en caso de lograr la conexión exitosamente. Sumado a esto, otro desafío importante que se debe abarcar a futuro es el cambio entre pisos, es decir, subir del piso 1 al piso 2 manteniendo la guía y la ruta del usuario. Finalmente, otro tema sustancial que se debe trabajar es el agregar más mapas del recinto para poder abarcar diferentes zonas y lograr que los usuarios puedan movilizarse por otros sectores de la universidad y no solamente en el edificio C.

Se alcanzó a estudiar, por medio de encuestas, que la aplicación sería una herramienta útil para los alumnos de la universidad, tanto para los de primer año como los más antiguos, debido a que, muchos afirman no conocer en su totalidad la universidad. Sumado a que también es una herramienta útil para las visitas que ingresan a la universidad y son ajenos a ella.

## Referencias

- [1] C. Alippi and G. Vanini, "RSSI-based and calibrated centralized localization technique for wireless sensor networks," Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06), 2006, pp. 5 pp.-305, doi: 10.1109/PERCOMW.2006.13.
- [2] Pavel Kriz. (2016, 18 abril). Improving Indoor Localization Using Bluetooth Low Energy Beacons. Hindawi. <https://www.hindawi.com/journals/misy/2016/2083094/>
- [3] Facultad de Ciencias Exactas Universidad Nacional del Centro de la Provincia de Buenos Aires, & A.C. (2019, abril). Algoritmos avanzados de posicionamiento en interiores utilizando la combinación de distintos tipos de sensores. <https://www.ridaa.unicen.edu.ar/xmlui/bitstream/handle/123456789/2046/Tesis>
- [4] R. (2015, 16 julio). Beacons. Mucho más que promociones. Marketing 4 Ecommerce. <https://marketing4ecommerce.net/beacons-mucho-mas-que-promociones/>
- [5] Ligeró, R. L. (2019, 17 abril). Beacons: el elemento fundamental del posicionamiento en interiores. Accent Systems. <https://accent-systems.com/es/blog/beacons-elemento-fundamental-posicionamiento-interiores/?v=5bc574a47246>
- [6] Universidad Veracruzana, & Zavala, C. S. Z. T. (2018, enero). Diseño de una red beacon centrada en personas con discapacidad visual. universidad veracruzana.
- [7] Universidad distrital Francisco José de Caldas, & Bustos, P. A. B. H. (2018). Aplicación móvil con tecnología beacon para el apoyo de la difusión de información de los espacios del campus universitario. universidad distrital Francisco José de Caldas.
- [8] Universidad Veracruzana. (2018). Sistema para orientar a personas con discapacidad visual en espacios cerrados mediante tecnología beacon.
- [9] Suárez, J.C.: Discapacidad visual y ceguera en adulto: Revisión del tema. Revista Medicina U.P.B., vol. 30, N<sup>o</sup> 2, pp. 170-180. (2011).
- [10] Dawson, N., Jeannette, A., Maldonado, M., Ramos, A.: Guía de Orientación para el Docente que atiende estudiantes con Discapacidad Visual. Instituto Panameño de Habilitación Especial (2014).

- [11] Ranchal, J.: Inicios, evolución y futuro del teléfono móvil. MuyCanal (31 de enero de 2014).
- [12] Sáenz, M.A.: Sistema de Posición y Orientación Móvil para Personas Ciegas en Ambientes Cerrados. Tesis de Maestría. Universidad de Chile (2009).
- [13] Esteve, M.: OnTheBus. Proyecto de Fin de Carrera. Universidad Autónoma de Barcelona. (2011).
- [14] ONCE: Aplicación ONCE-CIDAT Metro para iPhone. Centro de Investigación, Desarrollo y Aplicación Tiflotécnica de la ONCE (CIDAT). (2012).
- [15] E9 Dear Beacon. (2021). [Https://Www.Minew.Com/](https://www.minew.com/).  
<https://www.minew.com/products/e9-dearbeacon.html>
- [16] i3 Robust Beacon. (2021). [Https://Www.Minew.Com/](https://www.minew.com/).  
<https://www.minew.com/products/i3-robust-beacon.html>
- [17] Jianqiao Xiong; Qin Qin; Kemin Zeng. (2015, abril). A Distance Measurement Wireless Localization Correction Algorithm Based on RSSI. IEEE. <https://doi.org/10.1109/ISCID.2014.246>
- [18] Akhand Pratap Singh; Devesh Pratap Singh; Santosh Kumar. (2016, enero). NRS-SI: New proposed RSSI method for the distance measurement in WSNs. IEEE. <https://doi.org/10.1109/NGCT.2015.7375129>
- [19] Oguejiofor O. S, Aniedu A. N, Ejiofor H. C, Okolibe A. U. (2013, septiembre). Trilateration Based localization Algorithm for Wireless Sensor Network. Blue Eyes Intelligence Engineering Sciences Publication.
- [20] i10 durable beacon (2021). [Https://www.minew.com/product/i10-indoor-location-beacon/](https://www.minew.com/product/i10-indoor-location-beacon/)
- [21] Kittipong Kasantikul; Chundi Xiu; Dongkai Yang; Meng Yang. (2015, agosto). An enhanced technique for indoor navigation system based on WIFI-RSSI. IEEE. <https://doi.org/10.1109/ICUFN.2015.7182597>