Desarrollo de una aplicación para recorridos virtuales mediante el uso de realidad aumentada para visitas guiadas en el Centro de Innovación y Tecnología de la Universidad del Valle de Guatemala

Pablo Daniel Gonzalez Ramos

UVG UNIVERSIDAD DEL VALLE DE GUATEMALA

Universidad del Valle de Guatemala
Facultad de Ingeniería
Ingeniería en Ciencia de la Computación y Tecnologías de la Información

Desarrollo de una aplicación para recorridos virtuales mediante el uso de realidad aumentada para visitas guiadas en el Centro de Innovación y Tecnología de la Universidad del Valle de Guatemala

Evaluación y Selección de Tecnologías para sistemas IPS para el CIT

PARA OPTAR AL GRADO DE LICENCIATURA EN:
Ingeniería en Ciencia de la Computación y Tecnologías de la Información

EN MODALIDAD DE MEGAPROYECTO TECNOLÓGICO

Pablo Daniel Gonzalez Ramos 20362

Guatemala
Noviembre del 2024

Vo.Bo.:

(f) _____

Ing. Andres Tahuico

Tribunal Examinador:

(f) _____

Ing. Andres Tahuico

(f) _____

(f) _____

Fecha de aprobación: Guatemala, .

Agradecimientos

Quiero empezar agradeciendo a mi familia, que siempre ha estado a mi lado con su apoyo incondicional. Su confianza en mí y su aliento constante me han impulsado a enfrentar cada desafío en este camino. También quiero dar un agradecimiento especial al Ingeniero Andrés Tahuico, mi asesor, por su valiosa guía y apoyo a lo largo de esta tesis. Su experiencia y dedicación han sido fundamentales para mi crecimiento académico. Asimismo, agradezco a todo el equipo del Megaproyecto del Tour de Realidad Aumentada de la UVG. Su compromiso, esfuerzo y colaboración fueron esenciales para alcanzar los objetivos propuestos y superar los desafíos que enfrentamos juntos. Este trabajo no habría sido posible sin el apoyo de todos ustedes,.

No puedo dejar de mencionar a la Ingeniera Dulce Chacón, quien ha estado siempre pendiente del seguimiento del proyecto, ofreciendo consejos que me han ayudado a llevarlo de la mejor manera posible.

Por último, gracias a la Universidad del Valle de Guatemala por brindarme un entorno de aprendizaje tan enriquecedor, lo que me ha permitido desarrollar mis habilidades y prepararme para mi futuro profesional.

Índice

Índice de tablas

Índice de figuras

Resumen

Este trabajo presenta la evaluación y selección de tecnologías para sistemas de Posicionamiento en Interiores (*IPS*) en el Centro de Innovación y Tecnología (CIT) de la Universidad del Valle de Guatemala. El proyecto se centra en comparar las tecnologías *UWB* y *BLE* para determinar cuál es la más adecuada para implementar un sistema de navegación precisa en el campus universitario. La metodología incluye la implementación de pruebas piloto, análisis de resultados y selección final de la tecnología más efectiva.

Los objetivos del proyecto fueron evaluar y seleccionar tecnologías accesibles para un sistema de Posicionamiento de Interiores para el CIT de la UVG. Específicamente, se buscó implementar las tecnologías *UWB* y *BLE* para el desarrollo del sistema de posicionamiento en interiores, analizar las características y capacidades de cada tecnología, comparar y contrastar los resultados de las pruebas piloto, y desarrollar un documento guía para la configuración y mantenimiento de los sistemas *IPS*. Para alcanzar estos objetivos, se siguió un enfoque estructurado que incluyó la investigación de tecnologías disponibles, la implementación de pruebas piloto y el análisis de resultados. Se evaluaron factores clave como la precisión en la medición de distancias, el costo de cada baliza, la facilidad de integración con los diferentes sistemas, la viabilidad de las tecnologías con respecto a la conexión de los teléfonos móviles y su escalabilidad.

El objetivo principal de evaluar y seleccionar tecnologías accesibles para un sistema de Posicionamiento de Interiores para el CIT de la UVG se cumplió exitosamente. Los resultados mostraron que la tecnología *UWB* ofrece una mayor precisión y estabilidad en la localización de dispositivos móviles dentro del campus, mientras que *BLE* presentó limitaciones significativas debido a interferencias y variabilidad en las mediciones. A pesar de estas limitaciones, la implementación de *BLE* se consideró debido a su bajo costo y facilidad de integración.

Para futuras versiones del proyecto, se recomienda utilizar sensores *UWB* combinados con un algoritmo de pathfinding . De esta manera, se podrá ofrecer a la Universidad del Valle de Guatemala un sistema de navegación completamente inmersivo y satisfactorio, mejorando significativamente la experiencia de los usuarios en el campus.

CAPÍTULO 1

Introducción

En el mundo contemporáneo, donde la tecnología desempeña un papel fundamental en casi todos los aspectos de nuestra vida diaria, la integración de soluciones tecnológicas innovadoras en entornos educativos se ha vuelto una necesidad imperativa [1] . La Universidad del Valle de Guatemala (UVG), como institución líder en la región, reconoce la importancia de aprovechar las últimas tendencias tecnológicas para mejorar la experiencia de sus miembros. En este contexto, surge la iniciativa de desarrollar un tour de realidad aumentada que permitirá a estudiantes, profesores, visitantes y personal administrativo explorar el campus de manera interactiva y enriquecedora.

Sin embargo, uno de los principales desafíos que enfrentan los visitantes del campus universitario es la orientación y la navegación eficiente. La falta de un guía disponible en cualquier momento del día puede resultar en retrasos, frustración e incluso desmotivación en la exploración y utilización de los recursos disponibles en la universidad. Para abordar esta problemática, el proyecto incluye la integración de un sistema de Posicionamiento en Interiores (*IPS*), que permitirá proporcionar información precisa sobre la ubicación de los usuarios en tiempo real dentro del campus [2] La elección de la tecnología adecuada para el sistema de *IPS* es esencial para garantizar el éxito y la efectividad del proyecto en su conjunto [3] . Por lo tanto, esta tesis se centra en la evaluación y selección de tecnologías para el sistema de *IPS*, lo que constituye un paso crucial para asegurar una navegación precisa y eficiente en el tour de realidad aumentada de la UVG. La metodología utilizada en este proceso incluye la selección de tecnologías para evaluación, el diseño y la implementación de pruebas , el análisis de resultados y la selección final de la tecnología más adecuada.

La importancia de este tema radica en su capacidad para mejorar significativamente la experiencia de usuario y promover la eficiencia en el campus universitario. Al proporcionar una solución innovadora para la orientación y navegación en interiores, el proyecto no solo puede contribuir al desarrollo tecnológico en la UVG, sino que también sentará las bases para futuros desarrollos y mejoras en el ámbito de la navegación *indoor* y la realidad aumentada en entornos educativos. En resumen, la evaluación y selección de tecnologías para el sistema de *IPS* son fundamentales para el éxito del proyecto de tour de realidad aumentada de la UVG, y su importancia se refleja en el impacto potencial que tendrá en la experiencia y el desarrollo tecnológico en el campus universitario.

CAPÍTULO 2

Objetivos

2.1. Objetivo General

Evaluar y seleccionar tecnologías accesibles para un sistema de Posicionamiento de Interiores para el CIT de la UVG.

2.2. Objetivos Específicos

- Implementar las tecnologías *UWB* y *BLE* para el desarrollo del sistema de posicionamiento en interiores (*IPS*) en el contexto del campus universitario.
- Analizar las características y capacidades de cada tecnología, incluyendo precisión, alcance, escalabilidad, costo y requisitos de infraestructura.
- Comparar y contrastar los resultados de las pruebas piloto para determinar cuál es la tecnología más adecuada.
- Desarrollar un documento guía que describa el proceso de configuración y mantenimiento de los sistemas de Posicionamiento en Interiores *UWB* y *BLE*.

CAPÍTULO 3

Justificación

En el contexto mundial actual en donde la tecnología juega un papel fundamental en casi todos los aspectos de nuestra vida diaria, la integración de soluciones tecnológicas innovadoras en entornos educativos se ha convertido en una necesidad para cualquier ente educativo [4]. La Universidad del Valle de Guatemala, como una institución educativa líder en la región, reconoce la importancia de aprovechar las últimas tendencias tecnológicas para mejorar la experiencia de sus estudiantes, profesores, visitantes y personal administrativo. Sin embargo, uno de los principales desafíos que enfrentan los visitantes del campus universitario es la orientación y la navegación eficiente. La falta de un guía disponible en cualquier momento del día para ayudar a los nuevos estudiantes, visitantes o incluso miembros del personal a conocer las instalaciones de la universidad puede resultar en retrasos, frustración e incluso desmotivación en la exploración y utilización de los recursos disponibles en la universidad.

En este contexto, el proyecto de implementación de un tour de realidad aumentada para la UVG y la integración de un sistema de Posicionamiento en Interiores adquieren una importancia significativa. La combinación de estas soluciones tecnológicas innovadoras tiene como objetivo abordar los desafíos de orientación y navegación en el campus universitario. La realidad aumentada permite superponer información digital sobre el entorno físico, facilitando a los usuarios una exploración visualmente atractiva y comprensible del campus. Además, el sistema de Posicionamiento en Interiores ofrece una localización precisa en tiempo real, mejorando la eficiencia de los desplazamientos dentro de las instalaciones. Estas características no solo ayudan a los usuarios a orientarse de manera más efectiva, sino que también enriquecen su experiencia al proporcionar un acceso fácil y directo a información relevante y contextualizada, como detalles históricos de los edificios, eventos, y otros recursos útiles [5].

El tour de realidad aumentada debe proporcionar una plataforma intuitiva y accesible para explorar el campus, mientras que el sistema de *IPS* debe garantizar una navegación precisa y eficiente en entornos interiores. La justificación para la evaluación y selección de tecnologías para el sistema de *IPS* en este proyecto radica en su papel fundamental para garantizar el éxito y la efectividad de la iniciativa en su conjunto [2]. La elección de la tecnología adecuada para el sistema de *IPS* determinará en gran medida la calidad y la experiencia del usuario en el tour de realidad aumentada [2].

CAPÍTULO 4

Alcance y limitaciones

4.0.1. Limitaciones

Desde el inicio del proyecto, se enfrentaron algunas limitaciones. Un obstáculo importante fue el presupuesto, ya que no se pudieron adquirir más *beacons* Estimote para implementar el sistema *UWB* de forma completa. Esto limitó la extensión de la prueba de implementación, que quedó restringida a una cantidad reducida de 3 y, por tanto, a un área de cobertura menor. Sin embargo, se contó con tiempo suficiente para desarrollar las configuraciones necesarias del sistema y realizar las pruebas planeadas.

Respecto a las fuentes de información, se disponía de documentación confiable, incluyendo los foros de Arduino, que brindaron soporte práctico para la programación y configuración de los dispositivos. Además, la documentación oficial del *SDK* de Estimote fue fundamental para la correcta programación y recepción de señales de las balizas *BLE*. En cuanto a los resultados, estos pueden ser publicados libremente en la universidad, ya que no incluyen datos sensibles, permitiendo así que otros estudiantes y profesores accedan a los hallazgos y continúen con investigaciones relacionadas.

4.0.2. Alcance

A futuro, el alcance de este proyecto puede expandirse significativamente con la implementación completa de sensores UWB. Esto permitirá una experiencia de navegación más precisa y mejorada dentro del Centro de Innovación y Tecnología de la Universidad del Valle de Guatemala. La tecnología UWB proporcionará a los usuarios una orientación más detallada y en tiempo real, facilitando la localización de áreas y servicios dentro del campus.

Además el sistema también abriría la posibilidad de poder implementar nuevos enfoques como lo puede ser guiar a un estudiante nuevo en la universidad por los diferentes salones de clase para poder encontrar su destino basado en la ubicación en la que se encuentra dicho estudiante, también al recopilar los datos de las ubicaciones se podría visualizar datos como las áreas de interés que los usuarios tienen dentro del campus.

CAPÍTULO 5

Marco Teórico

5.1. Sistemas IPS

Un sistema *Indoor Position System* o por su siglas en inglés (*IPS*) es un sistema que constantemente en tiempo real está determinando la posición de una persona u objeto en un ambiente cerrado o interno. [6]. Los *IPS* tienen un amplio rango de aplicaciones, desde la navegación y el guiado en tiempo real en espacios grandes y complejos, hasta el seguimiento de activos y personas en áreas cerradas [7]. En el ámbito educativo, como el campus de la Universidad del Valle de Guatemala, un sistema de *IPS* puede mejorar significativamente la experiencia de los usuarios, facilitando la orientación y optimizando el uso de los recursos disponibles en las instalaciones [8]

5.1.1. ¿Por qué sistemas de Posicionamiento de Interiores?

Existen muchas características las cuales hacen diferentes a los sistemas de posicionamiento de interiores [9],en comparación con los sistemas de localización de exteriores, los sistemas de posicionamiento de interiores suelen ser más complejos debido a objetos como paredes, vidrios, computadoras o cualquier otro equipo en donde se encuentre instalado el sistema que pueda reflejar las señales causando problemas como trayectorias múltiples de la señales y retardo en el tiempo de llegada de la señal. Además, la existencia de objetos conduce a una alta dispersión de la señal, se puede mencionar que los sistemas de posicionamiento de interiores se ven afectados por la estabilidad que tiene una señal, ya que la intensidad de la señal tiende a fluctuar debido a las interferencias como lo pueden ser dispositivos móviles, dispositivos bluetooth, microondas entre otros. [10]

En comparación con los sistemas de exteriores, los sistemas *IPS* están sujetos a movimientos estructurales en donde los puntos de referencia pueden moverse de un lugar a otro por lo que dichos cambios podrían ajustar y calibrar el sistema de posicionamiento para hacer frente a los cambios que tienen las estructuras con el día a día. En la mayoría de los casos estos sistemas requieren una mayor precisión y exactitud en comparación a los sistemas exteriores para poder lidiar con áreas que son más pequeñas y que contienen obstáculos.

Pero también se puede mencionar que hay características de estos sistemas que facilitan el posicionamiento [9] , un ejemplo de esto puede ser que como se trabaja con áreas pequeñas de localización, esto puede ayudar a las configuraciones iniciales del sistema teniendo áreas de cobertura no tan

grandes que son más fácilmente seguidas qué áreas extensas y por otro lado también estos sistemas son menos dinámicos dado que los objetos se mueven a velocidades considerables.

5.1.2. Métricas de Rendimiento de un sistema de Interiores

Los sistemas *IPS* utilizan numerosos mecanismos de posicionamiento que pueden variar en diferentes métricas como precisión, tecnología, escalabilidad [11], algunas de las implementaciones de un sistema de *IPS* pueden requerir un sistema de bajo costo mientras que otras pueden requerir sistemas de alta precisión lo que incrementa el costo de las implementaciones. En la siguiente tabla se describirán las métricas que puede tener un sistema de *IPS*

Métrica	**Definición**
Precisión	Grado de concordancia entre un valor de magnitud medido y un valor de magnitud real de una medida.
Disponibilidad	La disponibilidad en términos de porcentaje de tiempo.
Costo	Costo monetario de la implementación.
Escalabilidad	El grado en el que el sistema garantiza su correcto funcionamiento cuando se escala en cantidad de usuarios o cantidad de Beacons.

Tabla 5.1: Tabla de métricas para sistemas IPS.

5.1.3. Tecnologías para un sistema IPS

Para un sistema *IPS* se pueden utilizar diferentes tecnologías para poder obtener ventajas de cada una de ellas, para cada proyecto esta tecnología se debe seleccionar de la manera más detallada posible la tecnología a utilizar basada en las necesidades que se tienen, las tecnologías *IPS* se pueden clasificar en dos grandes grupos que son las tecnologías que requieren hardware especial y las tecnologías que son autónomas [6] un ejemplo de una tecnología que necesita hardware especial es la tecnología *UWB* al necesitar *beacons* o balizas para su correcto funcionamiento y por otro lado , tecnologías autónomas como dead reckoning no dependen de hardware externo, calculando la posición basada en la velocidad y el movimiento previo del objeto.

5.2. Beacons

Los *beacons* o mejor conocidos como balizas, son dispositivos de transmisión que utilizan diferentes tecnologías para enviar señales a dispositivos cercanos, estos dispositivos juegan un gran papel en un sistema *IPS*, ya que estos son capaces de proporcionar información de ubicación en entornos donde el GPS no es efectivo. [11]

Los *beacons* trasmiten señales periódicas en intervalos regulares, estos son conocidos por su bajo consumo energético . Cada beacon cuenta con un identificador único junto con datos adicionales, como lo puede ser la señal recibida o por sus siglas en ingles (*RSSI*), las cuales los dispositivos móviles cercanos pueden recibir. Los dispositivos móviles o tablets, pueden utilizar la intensidad de la señal y otros parámetros para estimar su distancia Beacon y por ende su ubicación. [12]

Los *beacons* ofrecen varias ventajas, incluyendo un bajo consumo de energía, que permite que funcionen durante largos períodos con una batería pequeña, ademas de ser sencillos de instalar, lo que los hace adecuados para una gran gama de aplicaciones [13]. A pesar de sus beneficios, estos también presentan algunas limitaciones, en donde la precisión puede verse afectada por obstáculos físicos como paredes, vidrios o muebles. La intensidad de la señal puede variar debido a la interferencia de otras señales electromagnéticas, lo que puede llevar a errores en la estimación de un usuario [14]

5.3. Tecnología UWB

UWB es una señal de radio frecuencia que ocupa una porción del espectro de frecuencia que es mayor al 20 % de la frecuencia de la portadora central, o tiene un ancho de banda mayor a 500 MHz. *UWB* es un canal de comunicación que distribuye información sobre una amplia porción del espectro de frecuencia. Esto permite que los transmisores *UWB* transmiten grandes cantidades de datos mientras se consume poca energía [15], esta tecnología se puede utilizar para posicionamiento utilizando la diferencia de tiempo de llegada o por sus siglas en inglés (*TDOA*) de las señales de radio frecuencia para obtener la distancia entre el punto de referencia y el objetivo.

UWB es una de las tecnologías más recientes en el mercado. La tecnología precursora de *UWB* se conoce como la tecnología de banda base e impulso. La tecnología *UWB* se volvió comercial en el año 1990 cuando el departamento de Estados Unidos fue el primero en utilizar el término *UWB* [16] . Esta tecnología transmite pulsos extremadamente cortos a lo largo de una banda de frecuencia amplia de baja densidad espectral de potencia, lo que permite altas tasas de datos y una eficaz penetración en obstáculos [16].

Existen 3 áreas principales en donde la tecnología *UWB* es una buena implementación la primera es para la comunicación de sensores la segunda en posicionamiento y monitoreo y la tercera en radares [17] En particular, el posicionamiento *UWB* ofrece un seguimiento preciso en tiempo real en entornos interiores [18], siendo útil para inventarios móviles, servicios de emergencia, navegación para personas con discapacidad visual, seguimiento de personas o instrumentos y reconocimiento militar. *UWB* destaca por su alta tasa de datos de hasta 100 Mbps [19] , su capacidad para reducir interferencias multipath y su precisión de localización, que puede alcanzar sub-centímetros. Además, no requiere línea de vista y es resistente a interferencias externas, lo que lo hace más fiable que otras tecnologías de posicionamiento. Sistemas comerciales como Ubisense y aplicaciones militares como Alereon ya utilizan *UWB* para seguimiento y detección [20]. Empresas como Decawave integran *UWB* en aplicaciones de gestión de inventarios y monitoreo de flujo de producción, mientras que start-ups como BeSpoon están incorporando chips *UWB* en smartphones para mejorar funciones de localización, como encontrar pertenencias o personalizar el dispositivo según la ubicación interior. [21]

La tecnología *UWB* representa una solución eficiente y precisa para diversas aplicaciones críticas que demandan alta precisión, como el posicionamiento en interiores, el seguimiento de personas o equipos y la navegación en situaciones complejas. Su capacidad para operar sin necesidad de línea de vista, su resistencia a interferencias externas, y la alta tasa de transferencia de datos que ofrece, la posicionan como una de las tecnologías más avanzadas en el campo de la localización. Al emplear modulación de señales avanzada y algoritmos como *TDOA*, *UWB* permite calcular distancias con gran exactitud, lo que mejora su aplicabilidad en sistemas de monitoreo en tiempo real. A medida que la tecnología sigue evolucionando, su integración en dispositivos móviles y sistemas comerciales sigue abriendo nuevas oportunidades para su implementación en industrias que requieren soluciones innovadoras y altamente confiables.

5.4. Tecnología BLE

Bluetooth low energy o por sus siglas en inglés *BLE* es una tecnología de comunicación diseñada para aplicaciones que requieren bajo consumo de energía y corto alcance, esta tecnología operan en un ancho de banda de 2.4 GHz y se destaca por su capacidad para poder transmitir datos en pequeños paquetes, lo que reduce significativamente el consumo energético comparado con las tecnologías como el Bluetooth clásico. Esto permite que los dispositivos que implementan estas tecnología puedan funcionar por un largo periodo de tiempo con baterías pequeñas, haciéndolos ideales para proyectos como puede ser sensores en dispositivos de salud o en soluciones que implementan tecnologías con Internet. *BLE* puede tramitar datos a una velocidad de hasta 1 Mbits. [22]

A pesar de su baja potencia, *BLE* es capaz de soportar múltiples conexiones simultáneas y transmitir datos de manera eficiente a través de su protocolo de baja energía. Una característica importante es su capacidad de alternar entre tres canales de publicidad, lo que reduce la interferencia y mejora la calidad de la señal en entornos concurridos [23]. Este método, conocido como diversificación de canales, se utiliza comúnmente en aplicaciones de posicionamiento en interiores , donde se busca minimizar las fluctuaciones del indicador de potencia de la señal recibida causadas por la reflexión y la dispersión de las ondas de radio.

En términos de posicionamiento, *BLE* es ampliamente utilizado en sistemas que requieren localizar dispositivos en tiempo real. La técnica más común empleada para estimar la ubicación de un dispositivo *BLE* es el Received Signal Strength Indicator o por sus siglas en ingles *RSSI* , que mide la potencia de la señal recibida en función de la distancia entre el transmisor y el receptor [22] . Sin embargo, el posicionamiento basado en *RSSI* presenta desafíos debido a las fluctuaciones del *RSSI*, causadas por obstáculos como paredes, personas y el efecto multipath, donde las señales reflejadas interfieren con la señal directa, lo que puede degradar la precisión. Para mejorar la precisión en estos escenarios, se emplean métodos como la trilateración ponderada, donde las distancias se calculan con base en las señales recibidas de múltiples puntos de referencia, y se aplican algoritmos como el filtro de Kalman, que suaviza las estimaciones eliminando las mediciones erróneas [24]

En cuanto a las técnicas de modulación de señales, *BLE* utiliza la modulación por desplazamiento de frecuencia gaussiana para transmitir datos. Esta modulación ayuda a minimizar el consumo de energía y es compatible con las necesidades de los dispositivos móviles [25]. Además, *BLE* admite la conmutación de canales adaptativa, lo que permite elegir el canal de comunicación óptimo para minimizar las interferencias en tiempo real [26]. Las nuevas versiones de *BLE*, como Bluetooth 5.0, han mejorado aún más el rendimiento al aumentar el rango de alcance, la velocidad de transmisión de datos, y la capacidad de los dispositivos de realizar actualizaciones de firmware a través de conexiones inalámbricas, lo que extiende sus aplicaciones en diversas áreas como la automatización industrial, el seguimiento en hospitales y la domótica [22]

BLE es ampliamente utilizado en aplicaciones de monitorización de la salud, donde dispositivos como pulseras y relojes inteligentes recopilan información sobre la actividad física, la frecuencia cardíaca y otros datos biométricos en tiempo real. Estos dispositivos pueden enviar datos a smartphones o a servidores en la nube para su procesamiento, todo con un consumo de energía optimizado. En entornos de atención médica, *BLE* también se emplea en soluciones de seguimiento de pacientes, ayudando a ubicar a personas en hospitales o residencias, mejorando la eficiencia del personal médico y la seguridad de los pacientes [22]

BLE es una tecnología clave en el campo de las comunicaciones inalámbricas de corto alcance, con aplicaciones que van desde el posicionamiento en interiores y el seguimiento en tiempo real, hasta la monitorización de la salud y el IoT industrial. Su bajo consumo de energía, combinado con su capacidad de proporcionar un posicionamiento preciso y fiable, lo convierte en una de las tecnologías más versátiles y ampliamente adoptadas en la actualidad. [22]

5.4.1. RSSI

El Indicar de intensidad recibida o *RSSI*, es utilizado en sistemas de comunicación inalámbrica, como en tecnologías *BLE*, calcula la intensidad de una señal de radio que se ha recibido. En *BLE* se utiliza con frecuencia para medir la distancia entre dispositivos, mejorar la comunicación y para poder encontrar y descubrir objetos. La intensidad de señal se expresa en db y cuanto mayor sea el valor de la señal, mas fuerte sera la señal. [27]

RSSI desempeña un papel de suma importancia en las redes, ya que proporciona una medida en tiempo real de la intensidad de la señal, lo que hace que esta media sea esencial para poder mantener conexiones, esta ayuda a los administradores de dispositivos o red a poder evaluar donde las señales inalámbricas son mas débiles y mas fuertes dentro de una área determinada, esta información se puede considerar crucial para optimar el rendimiento de los dispositivos o de una red. [28]

La interpretación de una lectura de RSSI es esencial para poder evaluar la calidad de una conexión . Los valores *RSSI* suelen presentarse en números negativos y en donde las cifras se acercan mas a 0 indica una señal mas fuerte, se puede mencionar que una lectura entre -30 decibeles a -50 decibeles se puede considerar una señal excelente, cuando los valores se encuentran entre -60 decibeles a -70 decibeles se puede considerar una señal buena pero los usuarios pueden experimentra problemas con la conectividad y por ultimo lo valores menos a -70 decibeles son señales mas débiles, lo que se puede tener efectos como perdida de señales o velocidades de datos lentas. [28]

El indicar de intensidad de señal recibida es un componente importante en la calidad de las conexiones inalámbricas, particularmente en tecnologías como *BLE*. Su capacidad para medir en tiempo real la fuerza de la señal permite no solo optimizar la comunicación entre dispositivos, sino también identificar áreas con cobertura insuficiente. Al interpretar correctamente estos valores se puede mejorar los sistemas en términos de rendimiento, garantizando así una mejor experiencia de los usuarios que utilizan estas señales y minimizando los problemas de perdida de señal.

5.4.2. Conversión de una señal a Distancia

La conversión de una señal *RSSI* a una distancia en metros es un proceso el cual es clave en el contexto en una a aplicación de un sistema *IPS*, en donde cuando se mide esta señal por medio de un proceso matemático o algoritmos específicos, es posible estimar la distancia entre el transmisor hacia un receptor, uno de los modelos matemáticos más conocidos para poder realizar este tipo de tareas es el modelo de propagación logarítmica que se ve descrito en la siguiente formula. [29]

$$d = d_0 \cdot 10^{\frac{(P_0 - P_r)}{10 \cdot n}}$$

- d: distancia estimada entre el transmisor y el receptor (en metros).
- d_0: distancia de referencia (en metros).
- P_0: potencia de la señal recibida ($RSSI$) a la distancia de referencia d_0 (en dBm).
- P_r: potencia de la señal recibida en el receptor (en dBm).
- n: exponente de pérdida de trayectoria.

5.5. Trilateración

La trilateración es una técnica geométrica utilizada para determinar la posición de un punto en el espacio mediante la medición de distancias desde tres puntos referidos conocidos un punto de comparación con una triangulación es que la trilateración se basa únicamente en distancias mientras que la triangulación utiliza ángulos para realizar los cálculos. [30]

En el contexto de un sistema *IPS*, la trilateración se aplica midiendo las distancias entre un dispositivo móvil y varios puntos de referencias fijos, como lo son las balizas *BLE* o las balizas *UWB* . Las distancias entre el teléfono y las balizas se calculan mediante indicadores *RSSI* o el ToF. Estas mediciones de distancia se utilizan para construir círculos alrededor de los puntos de referencia para poder encontrar la intersección de estas figuras definiendo la posición del dispositivo como se puede ver en la siguiente imagen.

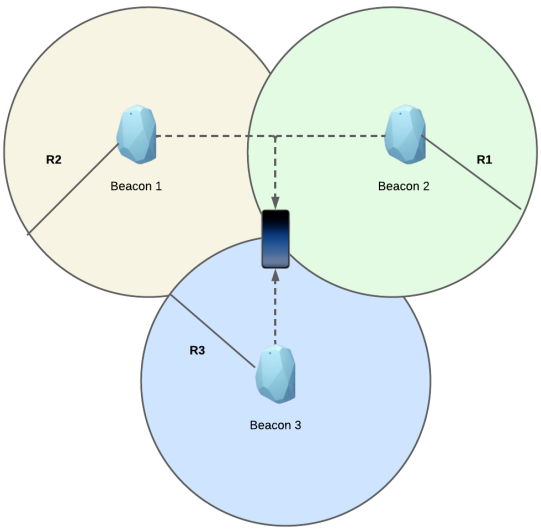

Figura 5.1: Ejemplo de trilateración usando balizas.

Para cada uno de los puntos de referencia, la distancia al dispositivo objetivo en las coordenadas desconocidas (x, y) se describe con la siguiente ecuación de distancia euclidiana:

$$(x - x_1)^2 + (y - y_1)^2 = d_1^2$$
$$(x - x_2)^2 + (y - y_2)^2 = d_2^2$$
$$(x - x_3)^2 + (y - y_3)^2 = d_3^2$$

Aquí, x y y son las coordenadas desconocidas del dispositivo objetivo, y d_1, d_2, y d_3 son las distancias desde el objetivo a cada uno de los puntos de referencia [30].

Este sistema de ecuaciones no lineales puede resolverse paso a paso. La técnica común es reducir el sistema de ecuaciones a un conjunto lineal para facilitar su solución.

Primera ecuación:

Empezamos con las tres ecuaciones originales. Restamos la primera ecuación de la segunda para eliminar los términos cuadráticos (x^2 y y^2):

$$(x - x_2)^2 + (y - y_2)^2 - (x - x_1)^2 - (y - y_1)^2 = d_2^2 - d_1^2$$

Al expandir y simplificar, obtenemos una ecuación lineal en x y y:

$$2(x_1 - x_2)x + 2(y_1 - y_2)y = d_1^2 - d_2^2 + x_2^2 - x_1^2 + y_2^2 - y_1^2$$

A esto lo llamamos **Ecuación A**.

Segunda ecuación:

Ahora, restamos la primera ecuación de la tercera de manera similar:

$$(x - x_3)^2 + (y - y_3)^2 - (x - x_1)^2 - (y - y_1)^2 = d_3^2 - d_1^2$$

Al expandir y simplificar, obtenemos otra ecuación lineal en x y y:

$$2(x_1 - x_3)x + 2(y_1 - y_3)y = d_1^2 - d_3^2 + x_3^2 - x_1^2 + y_3^2 - y_1^2$$

A esto lo llamamos **Ecuación B**.

Sistema lineal

Ahora tenemos dos ecuaciones lineales en x y y, las cuales pueden resolverse usando métodos algebraicos estándar, como la sustitución o la eliminación de variables. El sistema de ecuaciones es:

$$\begin{aligned} 2(x_1 - x_2)x + 2(y_1 - y_2)y &= d_1^2 - d_2^2 + x_2^2 - x_1^2 + y_2^2 - y_1^2 \quad \text{(Ecuación A)} \\ 2(x_1 - x_3)x + 2(y_1 - y_3)y &= d_1^2 - d_3^2 + x_3^2 - x_1^2 + y_3^2 - y_1^2 \quad \text{(Ecuación B)} \end{aligned}$$

El siguiente paso es resolver este sistema de ecuaciones utilizando métodos matriciales o sustitución. Para simplificar, podemos resolver para x a partir de una de las ecuaciones y luego sustituir en la otra para encontrar y, o podemos usar el método de eliminación [30].

Solución

Una vez obtenidas las ecuaciones lineales simplificadas, resolvemos el sistema para x y y. Esto nos da las coordenadas del dispositivo objetivo. Si hay errores en las distancias medidas es posible que los círculos no se intersecten en un solo punto exacto, por lo que se pueden emplear técnicas como el método de mínimos cuadrados para obtener la mejor estimación posible de la ubicación.

CAPÍTULO 6

Metodología

Para el desarrollo del proyecto se siguió un enfoque estructurado, cubriendo cada uno de los objetivos establecidos para garantizar el logro de los resultados deseados. El proceso comenzó con una investigación sobre las tecnologías disponibles para implementación de algoritmos de triangulación. Durante esta fase, se evaluaron factores clave como la precisión en la medición de distancias, el costo de cada balizas, la facilidad de integración con los diferentes sistemas, la viabilidad de las tecnologías con respecto a la conexión de los teléfonos móviles y su escalabilidad. De este análisis emergieron dos tecnologías principales primero la *UWB*, que destacó por su alta precisión en la medición de distancias entre los dispositivos y las balizas y la segunda *BLE* implementada en arduinos esto debido a su bajo costo haciéndola una opción atractiva para proyectos donde no se tiene presupuestos altos. En las siguientes secciones se puede ver la metodología empleada para cada sistema implementado.

6.1. UWB

En esta sección, se utiliza la tecnología *UWB* para medir la distancia entre los dispositivos y calcular la posición de un celular en un espacio interior utilizando métodos de trilateración los pasos para poder realizar esto son los siguientes.

6.1.1. Inicialización del entorno de desarrollo

El primer paso fue la creación de un repositorio de control de versiones para gestionar el código del proyecto. Posteriormente, procedió a inicializar el entorno de desarrollo utilizando Xcode 15.4 y Swift 5.10 como lenguaje de programación, creando el proyecto base que permitiría la integración de la tecnología *UWB*.

6.1.2. Implementación del SDK de Estimote UWB

Una vez inicializado el entorno, se integró el SDK de Estimote *UWB*, disponible en el repositorio oficial de GitHub Link: https://github.com/Estimote/iOS-Estimote-UWB-SDK . Este SDK permite la comunicación con los *beacons UWB* para recibir señales y medir distancias en metros. La insta-

lación del SDK se realizó mediante CocoaPods, gestionando las dependencias necesarias para el uso de la tecnología *UWB* en dispositivos iOS.

6.1.3. Configuración del entorno de prueba

Para las pruebas de trilateración, se definió un entorno físico con tres *beacons UWB* de Estimote, ubicados en un área configurada como un triángulo isósceles. Las posiciones de los *beacons* se definieron en coordenadas cartesianas las cuales fueron cambiando según las pruebas realizadas . Estas posiciones permiten crear un espacio cerrado y medir distancias desde cualquier punto dentro de este triángulo. Un ejemplo del sistema se describe en la siguiente imagen:

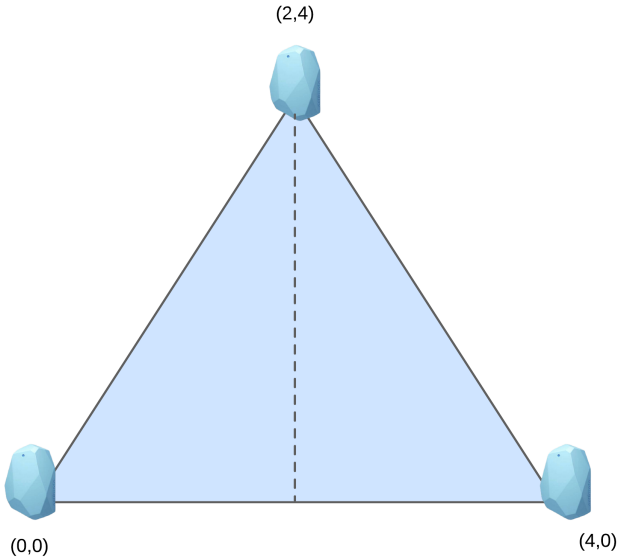

Figura 6.1: Ambiente Utilizado para *UWB*

6.1.4. Recepción y procesamiento de señales

Con los *beacons UWB* configurados y el entorno de prueba establecido, el siguiente paso fue la recepción de señales *UWB* en metros, obtenidas desde el dispositivo móvil a través del SDK de

Estimote. Estas señales representan la distancia entre el dispositivo móvil y cada uno de los tres *beacons*.

6.1.5. Algoritmo de trilateración

Una vez recopilados los datos de distancia, se aplicó el algoritmo de trilateración para calcular las coordenadas X e Y aproximadas del dispositivo móvil dentro del área definida. El algoritmo de trilateración utiliza las distancias desde el dispositivo móvil hacia los tres *beacons* como entrada para determinar su posición en el plano 2D. Mediante el cálculo de intersecciones de los círculos definidos por cada distancia, se obtiene una aproximación de las coordenadas X y Y, que se muestran en tiempo real en la pantalla de la aplicación.

6.1.6. Toma de resultados y cálculo del error

Una vez implementado el sistema de posicionamiento basado en *UWB*, se procedió a capturar la posición aproximada calculada por el algoritmo de trilateración. Esta posición fue comparada con la ubicación real del dispositivo en el entorno de prueba para medir la precisión del sistema.

6.2. BLE

En esta sección se describe el uso de la tecnología *BLE* p el sistema de posicionamiento , utilizando Arduino *BLE* para enviar y procesar señales *RSSI* por medio de un teléfono móvil A continuación, se detalla el proceso implementado para la integración de esta tecnología.

6.2.1. Inicialización del entorno de desarrollo

El primer paso fue crear un repositorio para gestionar el código del proyecto. A partir de ahí, se inició el desarrollo del sistema *BLE* utilizando Xcode 15.4 y Swift 5.10 para el manejo de la aplicación móvil, y el Arduino IDE 2.3.3 para programar los dispositivos Arduino *BLE*. En el repositorio se incluyó una carpeta dedicada exclusivamente a los archivos correspondientes a los Arduinos, donde se gestionan los códigos de cada dispositivo.

6.2.2. Implementación de los servidores BLE en Arduino

Cada dispositivo Arduino fue programado para actuar como un servidor *BLE*. Se implementó un código específico para cada Arduino con el objetivo de emitir señales *BLE*, proporcionando el indicador de fuerza de la señal de manera periódica. Cada Arduino *BLE* fue configurado con un identificador único para distinguir las señales provenientes de cada dispositivo. Esta etapa fue clave para garantizar la correcta identificación de las señales recibidas en el dispositivo móvil, lo que permitió conocer de qué Arduino provenía cada señal.

6.2.3. Configuración del entorno de prueba

El entorno de prueba consistió en la disposición de tres dispositivos Arduino *BLE* en un área definida.Las posiciones de los *beacons* se definieron en coordenadas cartesianas las cuales fueron

cambiando según las pruebas realizadas . Estas posiciones forman un triángulo dentro del cual se puede realizar el cálculo de la posición del dispositivo móvil utilizando las señales *BLE* recibidas. Cada dispositivo emitía señales *RSSI* que fueron captadas por el sistema móvil para el procesamiento. Un ejemplo del sistema se describe en la siguiente imagen:

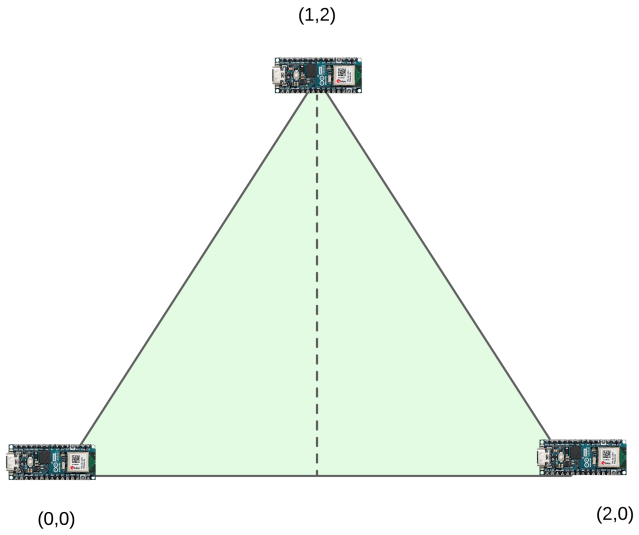

Figura 6.2: Ambiente Utilizado para BLE

6.2.4. Conversión de RSSI a distancia

Para convertir las señales *RSSI* en una medida útil de distancia, se utilizó la fórmula de pérdida de trayectoria , que permite calcular la distancia aproximada en metros entre el dispositivo móvil y cada uno de los Arduinos *BLE*, basándose en la atenuación de la señal.

6.2.5. Algoritmo de trilateración

Con las distancias calculadas entre el celular y cada uno de los tres *beacons BLE*, se aplicó el algoritmo de trilateración para determinar la ubicación del dispositivo móvil en un plano 2D. Al igual que en la fase *UWB*, se utilizaron las intersecciones de los círculos generados por cada distancia para calcular las coordenadas X e Y aproximadas, que se mostraron en la pantalla de la aplicación en tiempo real.

6.2.6. Toma de resultados y cálculo del error

De manera similar al sistema *UWB*, el sistema de posicionamiento basado en *BLE* también fue evaluado en cuanto a su precisión. Después de implementar el algoritmo de trilateración y convertir las señales *RSSI* a distancias, se calculó la posición aproximada del dispositivo móvil dentro del área delimitada por los tres *beacons BLE*.

CAPÍTULO 7

Resultados

7.1. Pseudocódigo Algoritmo Configuración BLE

El siguiente pseudocódigo detalla el proceso de configuración de un dispositivo Bluetooth Low Energy utilizando un Arduino, específicamente el modelo ESP32.

Algorithm 1 Pseudocódigo para configuración de un dispositivo BLE en Arduino

procedure SETUP
    Inicializar la comunicación serial con una velocidad de 115200 baudios
    Inicializar el dispositivo BLE con el nombre .esP32_Beacon_2"
    Crear un servidor BLE
    Obtener la referencia para la publicidad BLE
    Crear un objeto de datos de publicidad BLE
    Establecer los flags de publicidad a `BR_EDR_NOT_SUPPORTED` (0x04)
    Establecer el UUID de los servicios como "87654321-4321-4321-4321-210987654321"
    Asignar los datos de publicidad al objeto de publicidad
    Iniciar la publicidad
    Imprimir el mensaje "Beacon 2 started!.en el monitor serial
end procedure
procedure LOOP
    No es necesario hacer nada en este bucle
end procedure

7.2. Pseudocódigo Algoritmo Triangulacion

El siguiente pseudocódigo describe el proceso de cálculo de la posición utilizando la técnica de triangulación basada en las distancias medidas desde tres sensores ubicados en posiciones conocidas. La función CalculatePosition recibe como entrada las distancias desde cada sensor y, a través de una serie de cálculos, determina las coordenadas (x,y) de la posición del objeto en el espacio.

Algorithm 2 Pseudocódigo para calcular la posición usando Trilateración

procedure CalculatePosition
  Salida: Coordenadas (x, y) de la posición calculada
  Definir las coordenadas de los sensores:
  $sensor1 \leftarrow (x_1 = 0.0, y_1 = 0.0)$
  $sensor2 \leftarrow (x_2 = 4.0, y_2 = 0.0)$
  $sensor3 \leftarrow (x_3 = 2.0, y_3 = 4.0)$
  Convertir las distancias a tipo Double:
  $d1 \leftarrow distance1$, $d2 \leftarrow distance2$, $d3 \leftarrow distance3$
  Calcular los coeficientes:
  $A \leftarrow 2 \cdot (x_2 - x_1)$
  $B \leftarrow 2 \cdot (y_2 - y_1)$
  $C \leftarrow 2 \cdot (x_3 - x_1)$
  $D \leftarrow 2 \cdot (y_3 - y_1)$
  Calcular las constantes:
  $E \leftarrow d1^2 - d2^2 - x_1^2 + x_2^2 - y_1^2 + y_2^2$
  $F \leftarrow d1^2 - d3^2 - x_1^2 + x_3^2 - y_1^2 + y_3^2$
  Resolver para x:
  $denominatorX \leftarrow A \cdot D - B \cdot C$
  $numeratorX \leftarrow E \cdot D - B \cdot F$
  Si $denominatorX \neq 0$ **entonces**
    $x \leftarrow \frac{numeratorX}{denominatorX}$
  Sino
    $x \leftarrow 0.0$
  Resolver para y:
  $denominatorY \leftarrow C \cdot B - A \cdot D$
  $numeratorY \leftarrow E \cdot C - A \cdot F$
  Si $denominatorY \neq 0$ **entonces**
    $y \leftarrow \frac{numeratorY}{denominatorY}$
  Sino
    $y \leftarrow 0.0$
  Retornar las coordenadas (x, y)
end procedure

7.3. Resultados Distancia sensores

A continuación, se presentan los resultados de las mediciones realizadas utilizando dos tecnologías de localización: *UWB beacon* y Arduino *BLE beacon*. La primera tabla detalla las distancias reales en metros, las medidas obtenidas por los *beacons UWB*, y el porcentaje de error asociado. La segunda tabla muestra un análisis con el mismo enfoque para el sistema Arduino *BLE beaco*, donde se incluyen las mismas variables. Al comparar los errores porcentuales de ambas tecnologías, se pueden identificar las diferencias en su rendimiento y exactitud en la medición de distancias.

Tabla 7.1: Resultados Medición Mts UWB Beacon

Distancia Real (mts)	**Medida Beacons (mts)**	**Error (%)**
1	0.95	5.0 %
2	1.92	4.0 %
3	2.98	0.7 %
4	3.92	2.0 %
5	4.91	1.8 %
6	5.93	1.2 %
7	6.84	2.3 %
Promedio de Error		**2.42 %**

Tabla 7.2: Resultados Medición Mts Arduino BLE Beacon

Distancia Real (mts)	**Medida Beacons (mts)**	**Error (%)**
1	1.08	8.0 %
2	1.58	21.0 %
3	2.15	28.3 %
4	3.1	22.5 %
5	4.3	14.0 %
6	4.64	22.7 %
7	0.0	100.0 %
Promedio de Error		**30.93 %**

7.4. Resultados Sistema IPS UWB

En esta sección se muestran los resultados obtenidos a partir de la evaluación del sistema de posicionamiento interno (utilizando *beacons UWB* en tres configuraciones diferentes de un entorno

de prueba. Cada tabla presenta las coordenadas reales y experimentales de los puntos medidos, junto con el porcentaje de error en las direcciones , lo que permite analizar el rendimiento del sistema en entornos reales. A través de estas configuraciones, se busca identificar variaciones en la precisión y efectividad de la tecnología *UWB* para mejorar la localización de dispositivos en espacios cerrados.

Tabla 7.3: Tabla de Resultados 1 UWB Beacons

X Real	**Y Real**	**X Experimental**	**Y Experimental**	**Error X %**	**Error Y %**
0.5	0.5	0.57	0.60	14	20
4	0.5	3.95	0.61	1.25	22
2	4	1.95	3.92	2.5	2
Promedio de error				5.92	14.67

Tabla 7.4: Tabla de Resultados 2 UWB Beacons

X Real	**Y Real**	**X Experimental**	**Y Experimental**	**Error X %**	**Error Y %**
0.5	0.5	0.58	0.31	16	38
5	0.5	4.86	0.65	2.8	30
2.5	5	2.37	4.86	5.2	2.8
Promedio de error				8.00	23.60

Tabla 7.5: Tabla de Resultados 3 UWB Beacons

X Real	**Y Real**	**X Experimental**	**Y Experimental**	**Error X %**	**Error Y %**
0.5	0.5	0.64	0.38	28	24
6	0.5	5.76	0.63	4	26
3	6	2.76	5.85	8	2.5
Promedio de error				13.33	17.50

Tabla 7.6: Promedio Porcentaje de Error Sistema UWB

X	**Y**
9.08	13.94

7.5. Resultados Sistema IPS BLE

En esta sección se muestran los resultados obtenidos a partir de la evaluación del sistema de posicionamiento interno utilizando *beacons BLE* en tres configuraciones diferentes de un entorno de prueba. Cada tabla presenta las coordenadas reales y experimentales de los puntos medidos, junto con el porcentaje de error en las direcciones , lo que permite analizar el rendimiento del sistema en entornos reales. A través de estas configuraciones, se busca identificar variaciones en la precisión y efectividad de la tecnología *BLE* para mejorar la localización de dispositivos en espacios cerrados.

Tabla 7.7: Tabla de Resultados 1 BLE Beacons

X Real	**Y Real**	**X Experimental**	**Y Experimental**	**Error X %**	**Error Y %**
0.5	0.5	1.45	0.30	190	40
2	0.5	3.50	0.95	75	90
1	2	4.00	0.61	300	69.50
Promedio de error				188.33	66.50

Tabla 7.8: Tabla de Resultados 2 BLE Beacons

X Real	Y Real	X Experimental	Y Experimental	Error X %	Error Y %
0.5	0.5	1.68	0.23	236	54
3	0.5	5.23	1.23	74.33	146
1.5	3	6.32	6.00	321.33	100
Promedio de error				210.56	100.00

Tabla 7.9: Tabla de Resultados 3 BLE Beacons

X Real	Y Real	X Experimental	Y Experimental	Error X %	Error Y %
0.5	0.5	1.96	0.36	292	28
4	0.5	7.22	2.39	80.50	378
2	4	9.76	7.25	388	81.25
Promedio de error				253.50	162.42

Tabla 7.10: Promedio Porcentaje de Error Sistema BLE

X	Y
217.46	109.64

7.6. Costo Beacons

En esta sección , se presenta una tabla que detalla los costos asociados a los productos utilizados en el desarrollo del sistema de posicionamiento . En esta tabla se incluyen los precios de dos componentes clave: el Estimote *UWB beacon* y el Arduino ESP 32. Esta análisis de costos también permite a los interesados comprender la inversión necesaria para implementar soluciones basadas en tecnología *UWB* y Arduino en aplicaciones prácticas.

Tabla 7.11: Tabla de Costos

Producto	Costo
Estimote UWB Beacon	33.33 $
Arduino ESP 32	23.90 $

7.7. Costo de inversión en sensores uwb

En esta sección, se presenta una tabla que detalla los costos asociados a los elementos clave que se utilizaran en el desarrollo del sistema que se trabajaron en el modulo con nombre Desarrollo del plan de implementación de la aplicación de recorridos virtuales para diseñar e innovar el servicio y la efectividad de la aplicación mediante un análisis de costos e índices estratégicos y operacionales en el Centro de Innovación y Tecnología de la Universidad del Valle de Guatemala.", organizados en costos fijos y costos variables. La tabla incluye información sobre la periodicidad, cantidad, costos unitarios y totales de cada categoría.

Entre los costos fijos, se destacan la inversión en licencias necesarias para subir la aplicación a las plataformas App Store y Play Store, la adquisición de sensores para localización interna, y el material de promoción para marketing. Por otro lado, los costos variables incluyen elementos como la impresión de rótulos y afiches informativos, además de la adquisición de souvenirs promocionales.

Tabla 7.12: Tabla inversión con sensores

Categoría	Descripción	Cantidad	Costo Unitario (Q)	Costo Total (Q)	Periodicidad
Costos Fijos					
Licencia	Subir aplicación a App Store	1	2,308.00	2,308.00	Anual
Licencia	Subir aplicación a Play Store	1	193.00	193.00	Único
Sensores	Localización interna	100	257.33	25,733.33	Único
Marketing	Material de Promoción	1500	1.00	1,500.00	Anual
Costos Variables					
Impresión	Rótulos por Nivel	12	7.00	84.00	Anual
Impresión	Afiche Informativo	100	7.00	350.00	Anual
Souvenirs	Lapiceros	100	5.00	500.00	Anual
Souvenirs	Llaveros	100	3.00	300.00	Anual
Souvenirs	Stickers	100	2.00	200.00	Anual
			Total	**31,168.33**	

7.8. Disponibilidad de sistemas

En cuanto a la disponibilidad de los sistemas implementados, se observan diferencias importantes entre *UWB* y Arduino *BLE*. Los *beacons UWB*, según las especificaciones del fabricante, utilizan baterías AA, lo que les permite funcionar de manera continua durante aproximadamente dos años antes de necesitar un reemplazo, Por otro lado, los dispositivos Arduino *BLE* pueden estar conectados de forma indefinida a un transformador que los alimenta de manera constante. No obstante, es posible implementar un circuito adicional que permita alimentarlos mediante baterías, evitando la dependencia de una fuente de alimentación externa.

7.9. Escalabilidad

En términos de escalabilidad, ambos sistemas, *UWB* y Arduino *BLE*, demostraron ser igualmente capaces de integrar un mayor número de dispositivos. La adición de nuevos *beacons* en ambos sistemas requiere simplemente la configuración e incorporación de los IDs de los dispositivos adicionales en el software, lo que permite su integración sin mayores complicaciones. Este proceso de configuración es relativamente sencillo , lo que asegura que tanto *UWB* como Arduino *BLE* puedan escalar eficientemente a medida que se necesite aumentar la cantidad de dispositivos para cubrir áreas en donde no se cuenta con estos dispositivos.

7.10. Infraestructura Universitaria

La infraestructura de la Universidad del Valle de Guatemala es adecuada para la instalación de los sensores necesarios para ambos sistemas, *UWB* y Arduino *BLE*. Los pasillos, aulas y áreas comunes ofrecen ubicaciones estratégicas que permiten la colocación de los *beacons* o dispositivos Arduino sin dificultades. Estos espacios cuentan con suficiente cobertura y distribución para que los sensores puedan operar. Además, la arquitectura del campus facilita la expansión del sistema a otras áreas , lo que respalda la escalabilidad de la solución en el entorno Universitario.

7.11. Guía para configuración de sistemas IPS

En esta sección se presentan el desarrollo de una guía detallada sobre la conexión y configuración de los sistemas de posicionamiento en interiores basados en Bluetooth Low Energy, la cual se encuentra disponible en el siguiente documento: `https://docs.google.com/document/d/1koYWgfGGBmpup86wp8osIgRHBHIGfKiPWJYhtBZqJ5k/edit?tab=t.0`. Este recurso proporciona instrucciones precisas sobre cómo implementar el sistema, incluyendo los pasos para instalar el software necesario, configurar los UUID de los dispositivos *BLE*, configurar los sensores *UWB*, y ajustar las posiciones de las balizas según la configuración implementada.

7.12. Recursos de red universitaria

En este proyecto, se determinó que no es necesario utilizar la red universitaria para el funcionamiento de los sistemas *IPS* basados en tecnologías *UWB* y *BLE*. Esto se debe a que cada una de estas tecnologías opera de manera independiente de una infraestructura de red externa.

Los dispositivos BLE utilizados en este sistema, implementados a través de Arduinos, actúan como servidores BLE autónomos. Estos servidores transmiten información de manera directa al dispositivo móvil sin necesidad de conectarse a la red universitaria ni a otra red Wi-Fi o de datos.

Por otro lado, los sensores UWB tampoco requieren de una red externa para su operación. Esta tecnología se basa en la emisión y recepción de señales de radiofrecuencia de alta precisión[15], lo que permite calcular la posición del dispositivo móvil de forma local.

Dado estas características, tanto los Arduinos BLE como los sensores UWB ofrecen una solución independiente que no necesita estar conectada a la red para su correcto funcionamiento.

7.13. Aplicación

En esta sección se presentan las imágenes de las aplicaciones desarrolladas para el sistema de posicionamiento utilizando tecnología *UWB* y *BLE*. La primera imagen ilustra la interfaz de la aplicación *IPS UWB*, mostrando cómo se visualizan las posiciones en tiempo real dentro del entorno de prueba. La segunda imagen corresponde a la aplicación basada en Bluetooth Low Energy .

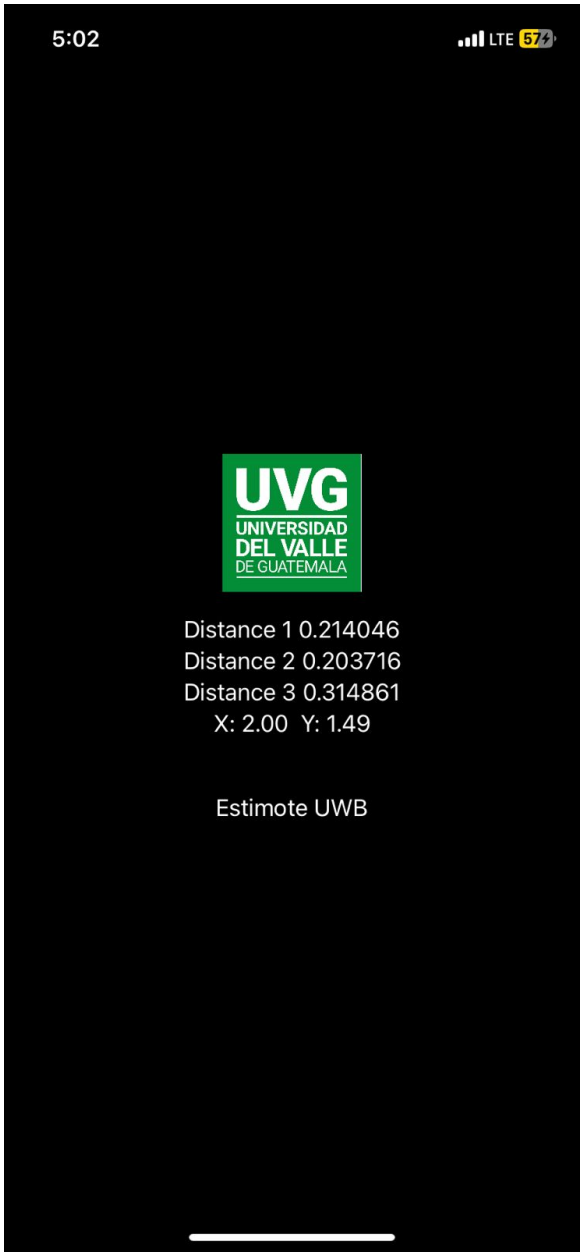

Figura 7.1: Resultado aplicación UWB

Figura 7.2: Resultado aplicación BLE

CAPÍTULO 8

Discusión

En esta sección se presenta un análisis detallado de los resultados obtenidos a lo largo del proyecto, destacando los aspectos que funcionaron según lo que se esperaba, aquellos que no se alcanzaron con las metas planteadas y las posibles causas detrás de los resultados obtenidos. Este análisis permite comprender mejor el rendimiento de los sistemas implementados ofreciendo así una visión clara de las fortalezas y debilidades de cada enfoque tomando, proponiendo así una base para proponer mejoras y optimizaciones en futuras aplicaciones que implementen estos algoritmos.

El objetivo principal del proyecto fue evaluar y seleccionar tecnologías accesibles para un sistema de Posicionamiento de Interiores para el Centro de Innovación y Tecnología de la Universidad del Valle de Guatemala. Este objetivo se puede mencionar que se cumplió, dado que se lograron implementar dos sistemas *IPS* uno *UWB* y otro *BLE*, que fueran capaces de localizar a un usuario dentro de un triángulo de balizas, aunque este objetivo se cumpliera y los resultados de los *UWB* fueron positivos, es importante mencionar que se obtuvieron resultados negativos con los sensores *BLE*, estos resultados eran los esperados debido a las limitaciones de hardware que se tenia con las balizas *BLE* y la cantidad de dispositivos que emiten interferencias dentro del contexto Universitario.

8.1. Implementación de algoritmo de Trilateración

Para poder empezar a entender los resultados de los sistemas *IPS* se tiene como base el algoritmo de trilateración este fue implementado exitosamente en el lenguaje de programación Swift, aprovechando las capacidades matemáticas y de procesamiento del lenguaje para calcular la posición de un dispositivo móvil con respecto a tres balizas. El objetivo de este algoritmo es obtener las coordenadas x e y del dispositivo móvil basándose en las distancias conocidas a tres puntos de referencia, cuyas posiciones son fijas también. El algoritmo comienza tomando las distancias proporcionadas como entrada, que se convierten a tipos `Double` para asegurar la precisión en los cálculos. Luego, utiliza una serie de ecuaciones geométricas para resolver un sistema de ecuaciones lineales que determinan las coordenadas x e y del dispositivo en cuestión.

En términos de eficiencia, la implementación en Swift permitió obtener resultados en tiempo real, debido a su capacidad de procesar rápidamente los cálculos matemáticos involucrados en la trilateración. El algoritmo calcula los coeficientes A, B, C y D utilizando las posiciones conocidas de los sensores, así como las constantes E y F, derivadas de las distancias entre los sensores y el dispositivo móvil. Esto le permite resolver las coordenadas, controlando también posibles errores de

cálculo como divisiones por cero para mantener la estabilidad y evitar resultados inválido Por lo que se puede mencionar que el algoritmo de trilateración se implementó de manera satisfactoria.

8.2. Distancias medias hacia un sensor

Al analizar los resultados obtenidos de las pruebas de medición de distancias hacia una única baliza , se puede observar diferencias notables en el rendimiento entre el sistema *UWB* y el sistema basado en *BLE* arduino.

En la primera tabla, los resultados del sistema *UWB* muestran una alta precisión en las mediciones . A media que la distancia real aumenta, las diferencias entre la distancia medida y la distancia real permanecen bajas, con un error promedio de 2.42 %. Esto sugiere que la tecnología *UWB* es capaz de mantener una precisión estable en diversas distancias. El error más alto se encuentra en medición a 1 metro con un 5 % de error, lo que podría atribuirse a un ligero sesgo de las mediciones de las distancias cortas pero en general el comportamiento de las distancias siempre es consistente.El error bajo en las mediciones muestra que las balizas *UWB* son una tecnología eficaz para sistemas de posicionamiento de interiores, debido a su baja sensibilidad a interferencias y la precisión que ofrece, incluso en entornos donde hay muchos dispositivos.

En la segunda tabla, los resultados obtenidos para la baliza *BLE* basado en Arduino se muestra un comportamiento menos confiable. El error promedio asciende a 30.93 % con una gran variabilidad en las mediciones. A 1 metro, el error es del 8 %, que ya es considerablemente más alto que en los sistemas *UWB*. Ademas, a distancias más largas, el error incrementa de gran manera alcanzando el 100 % a 7 metros de distancia, donde la baliza *BLE* no pudo emitir una señal que se captara para el sistema *UWB*.

Los resultados obtenidos con las mediciones individuales hacia una baliza muestran diferencias claras entre la tecnología *UWB* y el *BLE*. El sistema *UWB* demostró ser mucho más preciso y estable, con un error promedio del 2.42 %, lo que lo hace ideal para aplicaciones de posicionamiento en interiores que requieren alta precisión. En cambio, el sistema *BLE* basado en Arduino mostró un rendimiento significativamente inferior, con un error promedio del 30.93 %.

8.3. Resultados Sistemas IPS

Al momento de analizar los resultados de las pruebas de los sistemas *IPS* con *BLE* y *UWB*, se pueden observar diferencias significativas en la precisión y estabilidad entre ambas implementaciones.

Los resultados obtenidos para el sistema *IPS UWB* refleja una alta precisión considerable en la localización del dispositivo móvil en el entorno de la Universidad del Valle de Guatemala. En la primera configuración del sistema evaluada, el error promedio en la coordenada X fue de 5.92 % y en la coordenada Y de 14.67 %. En la segunda configuración, los valores de error promedio aumentaron ligeramente, alcanzando 8.00 % en X y 23.60 % en Y. Por ultimo en la tercera configuración , el error promedio alcanzo un total de 13.33 % en X y 17.50 % en Y. Al obtener los promedio para el sistema *UWB*, se obtuvo un 9.08 % en la coordenada X y un 13.94 % en la coordenada Y. Esta consistencia baja y variabilidad indica que el sistema UWB mantiene una precisión alta y estable incluso cuando se varia las posiciones del entorno inicial al cual se configuran los sensores.

El buen rendimiento del sistema *UWB* puede atribuirse a diferentes razones, una de las primeras razones es el sistema *UWB* trabaja en frecuencias altas lo cual reduce significativamente la interferencia permitiendo medir las distancias con una mayor precisión [31],en el contexto de la Universidad del Valle, cabe la pena mencionar que esto es un gran punto a favor debido a que en la Universidad

existen diferentes dispositivos como los routers wifi, teléfonos celulares que cuentan con Bluetooth activo y computadoras también cuentan con bluetooth que trabajan en una frecuencia de 2.4 GHZ [32] lo que hace que este tipo de señales no interfieran en el calculo del punto x e y del sistema *IPS*, ademas los dispositivos *UWB Beacons* de ESTIMOTE implementados para este sistema están diseñados para este tipo de tareas, lo que hace mas efectivo la implementación en entornos como la Universidad del Valle de Guatemala, contribuyendo a la exactitud y fiabilidad que presenta este sistema.

Por otro lado los resultados obtenidos para el sistema de posicionamiento *BLE* reflejan una precisión limitada en la localización de puntos en un entorno de la Universidad del Valle de Guatemala. En la primera configuración del sistema evaluada, el error promedio en la coordenada X fue de 188.33 % y en la coordenada Y de 66.50 %. En la segunda configuración, los valores de error promedio alcanzaron 210.56 % en X y 100.00 % en Y. Por último, en la tercera configuración, el error promedio alcanzó un total de 253.50 % en X y 162.42 % en Y. Al calcular el promedio general para el sistema *BLE*, se obtuvo un 217.46 % en la coordenada X y un 109.64 % en la coordenada Y. Esta alta variabilidad y los márgenes de error significativos indican que el sistema *BLE* enfrenta desafíos en la obtención de una localización precisa.

El rendimiento deficiente del sistema *BLE* se debe a varios factores. En primer lugar, *BLE* opera en frecuencias de menor ancho de banda en comparación con *UWB*, lo que lo hace más susceptible a interferencias de dispositivos cercanos. Un factor crítico es la interferencia en la banda de 2.4 GHz, donde operan tanto los dispositivos Bluetooth como los routers Wi-Fi [32] y es donde opera este tipo de tecnología opera en el entorno de la Universidad del valle de Guatemala en donde se cuenta con múltiples redes Wi-Fi y otros aparatos funcionando, es probable que estas interferencias generen degradaciones significativas en el rendimiento, aumentando la tasa de error en la recepción de paquetes y reduciendo la precisión de los cálculos de distancia [33]. Además, el *BLE* maneja las distancias mediante estimaciones basadas en la intensidad de la señal o RSSI, siendo particularmente sensible a las variaciones en el entorno. Esto se traduce en errores de alta magnitud en la estimación de coordenadas. [34]

Los Arduinos, aunque son dispositivos versátiles y de bajo costo, no están diseñados para aplicaciones industriales robustas que requieren resistencia a interferencias y mayor capacidad de procesamiento de señales [35]. Para tareas críticas de posicionamiento en tiempo real, los sistemas especializados, como aquellos que utilizan tecnologías de radiofrecuencia más avanzadas o UWB, son más adecuados debido a su mejor tolerancia a interferencias y su mayor precisión. [36]

8.4. Costo de implementación

El costo de implementación fue un factor importante desde los incios del proyecto dado que las inversiones necesarias para poder realizar una implementación completa de los sistemas es grande. En este contexto, se ha evaluado el costo de implementación de un sistema *IPS* básico basado en *UWB*, que asciende a $99.99, en comparación con un sistema equivalente basado en Arduino, que tiene un costo de $71.70.

La diferencia de aproximadamente $28.29 entre ambos sistemas refleja no solo el costo de los dispositivos en sí, sino también las capacidades y el rendimiento que cada uno ofrece. Los sistemas *UWB*, aunque más costosos, brindan ventajas significativas en términos de precisión y fiabilidad en la localización.

Tomando en cuenta el costo de las implementaciones a gran escala para el CIT, el sistema basado en *UWB* tendrá un costo de 39.46 % mayor en comparación con el sistema basado en arduinos *BLE*, esta diferencia porcentual es considerable pero al hacer una inversión inicial en el proyecto donde se prioriza la precisión en la aplicación del tour guiado la implementación de *UWB* es mejor.

8.5. Disponibilidad y escalabilidad

La disponibilidad de los sistemas *UWB* y arduino *BLE* presentan diferencias relevantes en terminos de su autonomia energética. Los *Beacons UWB* Estimote. al operar con baterías AAA, tinen la ventaja de ofrecer una disponibilidad continua de 2 años hasta requerir un cambio de baterías [37], se puede mencionar que su mantenimiento es sencillo al solo realizar el cambio de baterías dentro de este periodo de tiempo. En contraste los Arduinos *BLE* requieren una fuente de alimentación constante y suelen estar conectados a transformadores, lo cual en este sentido reduce la flexibilidad de ubicación y aumenta la dependencia de instalaciones eléctricas , sin embargo estos dispositivos también pueden configurarse con circuitos adicionales para poder ser alimentados por baterías que es la implementación mas optima para este tipo de balizas.

En cuanto a la escalabilidad de los sistemas los dos demuestran ser flexibles dado que la adición de nuevos dispositivos es sencilla al integrar al código los IDs correspondientes para su correcto funcionamiento esto permite expandir en cobertura de manera escalable y sin complejidad técnica.

8.6. Limitaciones

8.6.1. Limitaciones de recibimiento de señales

Durante las pruebas realizadas con los sistemas de posicionamiento en interiores basados en *UWB* y *BLE*, surgieron varias limitaciones técnicas que es importante resaltar. Una de las limitaciones más notables fue la dificultad de recepción de señal por parte de los teléfonos celulares, especialmente cuando los dispositivos móviles no estaban orientados directamente hacia las balizas. Esto sugiere que ambos sistemas presentan sensibilidad en la dirección de la señal y que una alineación directa es un factor importante para lograr mediciones precisas.

Esta característica implica que para maximizar la eficacia en un entorno de uso, como en un campus universitario, los dispositivos receptores y emisores deben tener una orientación adecuada para reducir la pérdida de señal. Esta observación es particularmente relevante para futuras aplicaciones, donde la posición y la orientación de los dispositivos pueden requerir ajustes para asegurar una conexión consistente y confiable con el sistema de balizas.

8.6.2. Compatibilidad con teléfonos moviles

La mayoria de los te teléfonos inteligentes actuales son compatibales con *BLE* esta tecnología fue introducida oficialmente con Bluetooth 4.0 en 2010. [38] Hoy en día, prácticamente todos los teléfonos Android que cuentan con Android 4.3 o superior son compatibles con *BLE*. En el ecosistema de Apple, todos los modelos a partir del iPhone 4S, el iPad de 3ra generación, y el iPod Touch de 5ta generación soportan *BLE*. . En la siguiente tabla se podrá visualizar con mayor exactitud los teléfonos compatibles con dicha tecnología. [39]

Dispositivo	**Modelos Compatibles**
Teléfonos Android	Todos los teléfonos Android con Android 4.3 o superior
iPhone	iPhone 4S y modelos superiores
iPad	iPad de 3ra generación y modelos superiores
iPod Touch	iPod Touch de 5ta generación y superiores

Tabla 8.1: Tabla de Compatibilidad de Dispositivos BLE

En comparación, la tecnología *UWB* aún está limitada a dispositivos de gama alta debido a sus aplicaciones especializadas en rastreo de precisión y comunicación de corto alcance. Apple integró *UWB* por primera vez en el iPhone 11 a través de su chip U1, extendiendo esta tecnología a todos los modelos posteriores. Por su parte, dispositivos Android también han adoptado el estándar de comunicación *UWB*, implementado a partir de Android 12. Modelos destacados incluyen el Google Pixel 6 Pro y 7 Pro, así como varios dispositivos de Samsung, como el Galaxy Note20 Ultra, la serie Galaxy S21+ y S22+, y los plegables Galaxy Z Fold2, Z Fold3 y Z Fold4. Adicionalmente, Android ha habilitado la *Android Core UWB API* mediante la biblioteca *Jetpack*, facilitando el desarrollo de aplicaciones que aprovechen esta tecnología. [37]

Marca	**Modelo**
Apple	iPhone 11 en adelante
Google	Pixel 6 Pro, Pixel 7 Pro
Samsung	Galaxy Note20 Ultra, Galaxy S21+, Galaxy S22+, Galaxy Z Fold2, Z Fold3, Z Fold4

Tabla 8.2: Compatibilidad de Dispositivos Móviles con Tecnología UWB

8.6.3. Limitaciones futuras con implementaciones UWB

Si la Universidad del Valle de Guatemala considerara implementar sistemas adicionales que operen en frecuencias similares a *UWB*, como en la banda de 3.1 a 10.6 GHz , podría enfrentarse a desafíos similares a los observados con *BLE* en la banda de 2.4 GHz. Esto se debe a que el uso compartido de anchos de banda por diferentes dispositivos genera interferencias y degradación en la calidad de la señal, especialmente en entornos densos como un campus universitario. A pesar de que *UWB* tiene una mayor resistencia a interferencias debido a su baja densidad de potencia, su rendimiento aún podría verse comprometido si otros dispositivos comparten la misma banda.

CAPÍTULO 9

Conclusiones

- El objetivo principal de evaluar y seleccionar tecnologías accesibles para un sistema de Posicionamiento de Interiores para el CIT de la UVG se cumplió exitosamente. La implementación de las tecnologías *UWB* y *BLE* permitió obtener una visión clara de sus aplicaciones y su desempeño en el contexto del CIT de la UVG, evaluando la capacidad de ambas para ofrecer precisión en la localización dentro del campus. La comparación entre estas tecnologías proporciona una base sólida para determinar la más adecuada según los requerimientos específicos de precisión y accesibilidad económica.
- Se logró una implementación exitosa de las tecnologías *UWB* y *BLE* para el desarrollo del sistema de posicionamiento en interiores en el contexto del campus universitario. Esto se alcanzó a través del desarrollo de algoritmos de trilateración en Swift, que facilitaron la localización precisa de dispositivos. Además, se diseñó una interfaz que permite una comunicación eficiente con las balizas *BLE* y *UWB*, lo que optimiza la visualización de las posiciones en tiempo real dentro del campus.
- Se llevó a cabo un análisis de las características y capacidades de las tecnologías *UWB* y *BLE*, abarcando aspectos clave como precisión, alcance, escalabilidad, costo y requisitos de infraestructura. Este estudio reveló que, según los resultados obtenidos en las pruebas, la tecnología *UWB* se destaca como la opción más adecuada para implementar en el CIT de la Universidad del Valle de Guatemala. La *UWB* no solo demostró una mayor precisión en la localización, sino que también ofreció un rendimiento superior en entornos complejos. Esto la convierte en una solución idónea para el posicionamiento en interiores en el contexto universitario, alineándose con las necesidades específicas del campus y garantizando una experiencia de usuario óptima.
- Se logró cumplir con éxito el objetivo de comparar y contrastar los resultados de las pruebas piloto, llevando a cabo evaluaciones mediante las aplicaciones desarrolladas. A través de estas pruebas, se determinó que la tecnología *UWB* es la más adecuada para el sistema de posicionamiento en interiores en el CIT de la Universidad del Valle de Guatemala. Los resultados evidenciaron que, en comparación con la tecnología *BLE*, la *UWB* ofrece una mayor precisión y consistencia en la localización, consolidándose como la opción preferible para satisfacer las necesidades específicas del entorno universitario.
- Se cumplió con éxito el objetivo de realizar una guía detallada para la configuración de los sistemas de posicionamiento en interiores *UWB* y *BLE*. Esta guía proporciona instrucciones sobre los pasos necesarios para configuración de ambos sistemas. La documentación facilita la implementación y asegura que el personal pueda operar los sistemas de una manera sencilla.

CAPÍTULO 10

Recomendaciones

10.1. Opciones de implementación

Para futuras implementaciones del proyecto de posicionamiento en el campus de la Universidad del Valle de Guatemala, se recomienda probar una red de dispositivos *BLE* que incluyan múltiples Tag portátiles para cada estudiante o miembro de la comunidad. Estos Tag deben mantenerse a una distancia mínima de 10 metros de los puntos de referencia denominados Wanesy Wave para lograr un reconocimiento efectivo y constante de la ubicación de cada usuario. Debido a la extensión del campus, se sugiere instalar un Wanesy Wave en cada aula, lo que permitiría una cobertura completa y adecuada en todas las áreas clave.

Cada Wanesy Wave recolectaría datos de los Tag y los transmitiría a un dispositivo central llamado iFemtoCell. Este dispositivo centralizado actuaría como punto de recolección, enviando los datos de posicionamiento a un servidor o dispositivo de análisis para su procesamiento y gestión. Esto permitiría evaluar la precisión y eficiencia de la red *BLE* y su capacidad para mejorar la experiencia de navegación en tiempo real. La implementación de este tipo de sistema se puede ver en la siguiente imagen.

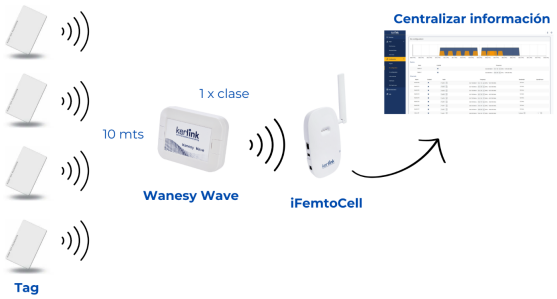

Figura 10.1: Red de dispositivos BLE con tags

10.2. Equipo multiciplinario

En la implementación del megaproyecto, seria recomendado incorporar estudiantes de Ingeniería mecatrónica en el equipo de trabajo, dado que su inclusión en el mismo puede generar un gran valor en el proyecto, específicamente en el conocimiento profundo del hardware de los dispositivos *UWB*, *BLE* o cualquier otra tecnología de comunicaciones que se dese implementar. Ademas estos estudiantes pueden aportar una perspectiva innovadora al considerar desarrollo de balizas personalizadas, utilizando diversas tecnologías de comunicación. La creación de balizas adaptadas a las necesidades especificas del proyecto podría ofrecer grandes beneficios como lo puede ser una mayor flexibilidad en la integración de nuevas tecnologías, una mejor eficiencia energética o la reducción de costos dado que las balzias se estarían fabricando dentro de la Universidad.

10.3. Pathfinding

En el contexto de la implementación de un tour guiado en la Universidad del Valle de Guatemala, se recomienda utilizar un algoritmo de pathfinding basado en balizas *UWB* en lugar de uno basado en trilateración *UWB*, esto se puede fundamentar basada en ventajas claves que hacen que un algoritmo de pathfinding sea una solución más adecuada para este proyecto. Uno de los primeros puntos claves es que los algoritmos de pathfinding representan una menor complejidad en su implementación, ya que requerirían manejar un numero significativamente menor de señales *UWB* en comparación con los algoritmos de trilateración, al trabajar con menos balizas el procesamiento de los datos es más sencillo, lo que facilita la instalación y el ajuste del sistema, reduciendo además el riesgo de errores o interferencias en la recepción de las señales.

En segundo lugar, el uso de pathfinding implica una importante reducción de costos. Dado que se necesitan menos balizas *UWB* para operación del sistema, un punto de comparación clave es que en este tipo de algoritmos se necesitan puntos de referencia a los cuales llegar en este caso se necesitaría 2 balizas para llegar de la baliza A la baliza B en cambio para un algoritmo de trilateración se necesitarían 3 balizas para su correcto funcionamiento. Por ultimo el mantenimiento de un sistema basado en pathfinding resulta más simple y eficiente, ya que al tener un numero menor de balizas en el sistema, las tareas de mantenimiento se vuelven menos complejas, lo que se traduce en una menor inversión de tiempo y recursos para garantizar el correcto funcionamiento del sistema a lo largo del tiempo. En la siguiente Imagen se puede visualizar el set de como se realizaría una implementación de los dispositivos *UWB* para un Algoritmo de pathfinding en el nivel 6 del CIT de la Universidad del Valle de Guatemala.

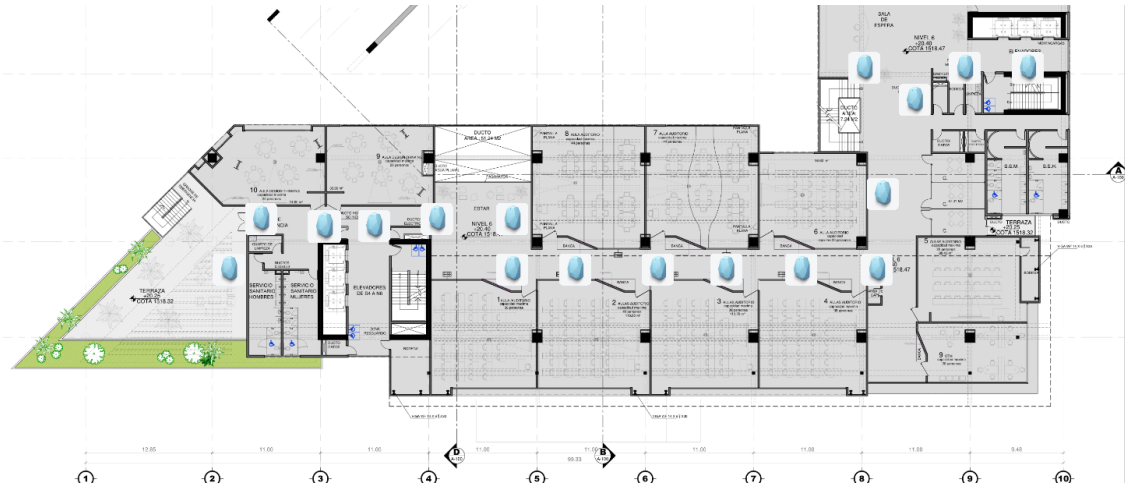

Figura 10.2: Ejemplo de implementación Dispositos UWB Para un Algoritmo de PathFinding

Bibliografía

[1] J. Salinas and J. Adell, *Hacia una visión contemporánea de la Tecnología Educativa*, 37th ed. Digital Education Review, 2020.

[2] J. Gibert and A. Gómez, *CIENCIA, TECNOLOGÍA Y SOCIEDAD EN AMÉRICA LATINA: LA MIRADA DE LAS NUEVAS GENERACIONES*. RIL editores, 2017.

[3] C. Arciniegas, "Estudio e implementaciÓn de tecnologÍas para posicionamiento en espacios interiores en dispositivo mÓvil por medio de un prototipo de aplicaciÓn en sistema operativo android," Master's thesis, Universidad Autónoma de Bucaramanga, 2020.

[4] C. Dziuban, C. R. Graham, P. D. Moskal, A. Norberg, and N. Sicilia, "Blended learning: the new normal and emerging technologies," *International Journal of Educational Technology in Higher Education*, vol. 15, no. 1, Feb 2018.

[5] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual displays," *IEICE Transactions on Information and Systems*, vol. E77-D, no. 12, pp. 1321–1329, 1994.

[6] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, pp. 13–32, 2009.

[7] J. Torres-Sospedra and J. Huerta, "Indoor positioning technologies based on wireless communication," in *Advances in Emerging Trends and Technologies*. MDPI, 2019, pp. 69–93.

[8] S. K. Xiao, L. L. Liu, and W. J. Ding, "Campus navigation and information systems based on augmented reality," *IEEE Access*, vol. 6, pp. 51 672–51 678, 2018.

[9] R. Mautz, "Indoor positioning technologies," Ph.D. dissertation, ETH Zürich, Zürich, Switzerland, 2012.

[10] J. D. Shi, "The challenges of indoor positioning," National University of Singapore, Singapore, Tech. Rep., 2013.

[11] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, Nov 2007.

[12] T. H. S. L. Lin, "Bluetooth low energy: A survey of the protocol, applications, and implementations," *IEEE Access*, vol. 8, pp. 131 687–131 704, 2020.

[13] M. Y. A. Z. O. Han, "A review on indoor positioning system using bluetooth low energy (ble) technology," *Journal of Computer Networks and Communications*, vol. 2020, pp. 1–14, 2020.

[14] J. M. So and W. S. Won, "Analysis of signal interference in indoor location-based systems using ble beacons," in *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2019, pp. 35–42.

[15] M. S. Svalastog, "Indoor positioning-technologies, services and architectures," Cand Scient Thesis, University of Oslo, Oslo, Norway, 2007.

[16] M. Ghavami, L. B. Michael, and R. Kohno, *Ultra Wideband Signals and Systems in Communication Engineering.* Newark, NJ, USA: John Wiley & Sons, Ltd, 2006.

[17] K. Siwiak and D. McKeown, *Ultra-Wideband Radio Technology.* Newark, NJ, USA: John Wiley & Sons, Ltd, 2005.

[18] G. Cheng, "Accurate toa-based uwb localization system in coal mine based on wsn," *Physics Procedia*, vol. 24, pp. 534–540, 2012.

[19] M. Segura, V. Mut, and C. Sisterna, "Ultra wideband indoor navigation system," *IET Radar, Sonar & Navigation*, vol. 6, pp. 402–411, 2012.

[20] Ubisense Company, "Ubisense website," 2009, accessed on 30 September 2024. [Online]. Available: http://www.ubisense.net/en/

[21] C., "Uwb's dream is still alive in micro-location," 2014, accessed on 30 September 2024. [Online]. Available: http://www.rethink-wireless.com/2014/10/21/uwbs-dream-alive-micro-location-page1

[22] V. Cantón Paterna, A. Calveras Augé, J. Paradells Aspas, and M. A. Pérez Bullones, "A bluetooth low energy indoor positioning system with channel diversity, weighted trilateration and kalman filtering," *Sensors*, vol. 17, no. 12, p. 2927, 2017.

[23] Y. Zhuang, J. Yang, Y. Li, L. Qi, and N. El-Sheimy, "Smartphone-based indoor localization with bluetooth low energy beacons," *Sensors*, vol. 16, p. 596, 2016.

[24] R. Faragher and R. Harle, "An analysis of the accuracy of bluetooth low energy for indoor positioning applications," in *Proceedings of the 27th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2014)*, Tampa, FL, USA, 2014, pp. 201–210.

[25] D. Capriglione, D. Casinelli, and L. Ferrigno, "Use of frequency diversity to improve the performance of rssi-based distance measurements," in *Proceedings of the 2015 IEEE International Workshop on Measurements Networking (M&N)*, Coimbra, Portugal, 2015, pp. 1–6.

[26] D. Pissoort, H. Hallez, and J. Boydens, "Bluetooth low energy interference awareness scheme and improved channel selection algorithm for connection robustness," *Sensors*, vol. 21, no. 7, p. 2257, 2021.

[27] MetaGeek, "Understanding rssi," https://www.metageek.com/training/resources/understanding-rssi/, accessed: 19-Oct-2024.

[28] W. Castle, "Understanding rssi signal strength: What it means for your connection," 2024, accessed: 19-Oct-2024. [Online]. Available: https://wraycastle.com/es/blogs/knowledge-base/what-is-rssi

[29] N. Vara, G. A. Poletto, M. Cáceres, and A. J. Busso, "Cálculo de distancia entre los nodos de una red inalámbrica zigbee en función del parámetro rssi," n.d., (*).

[30] M. Gende and I. Molina, "Trilateración," June 2011, disponible en: mgende@fcaglp.unlp.edu.ar.

[31] D. Dadari and M. Shafi, "Ultra-wideband technology: A review of uwb principles and applications," *Journal of Telecommunications and Information Technology*, vol. 1, pp. 29–35, 2020.

[32] Intel, "¿cómo funciona bluetooth?" n.d., consultado el 28 de octubre de 2024. [Online]. Available: https://www.intel.la/content/www/xl/es/products/docs/wireless/how-does-bluetooth-work.html#:~:text=El%20Wi%2DFi%20funciona%20en,frecuencia%20de%202%2C4%20GHz.

[33] ATT, "Interferencia wi-fi: cosas que bloquean la señal wi-fi," n.d., consultado el 28 de octubre de 2024. [Online]. Available: https://www.att.com/es-us/internet/wifi-interference-things-that-block-wifi-signals/

[34] Abalta Technologies, "Micro-location part 2: Ble and rssi," 2021, consultado el 28 de octubre de 2024. [Online]. Available: https://abaltatech.com/blog/2021/01/microlocation2/

[35] Control.com, "Arduino applications in industrial automation," 2022, consultado el 28 de octubre de 2024. [Online]. Available: https://control.com/technical-articles/arduino-applications-in-industrial-automation/

[36] Zediot, "Uwb positioning and uwb communication: The two sharp edges of uwb technology," 2023, consultado el 28 de octubre de 2024. [Online]. Available: https://zediot.com/blog/uwb-positioning-and-uwb-communication-the-two-sharp-edges-of-uwb-technology/

[37] Estimote, "Estimote - indoor location and proximity solutions," 2024, consultado en octubre de 2024. [Online]. Available: https://estimote.com/?srsltid=AfmBOoqCyGFrmqTMuMghReKTq0sP9-iLgPMygPl40JOOCOEreJQ7WZN4

[38] Memfault, "Bluetooth low energy: A primer," 2024, consultado en octubre de 2024. [Online]. Available: https://interrupt.memfault.com/blog/bluetooth-low-energy-a-primer

[39] Mokoblue, "Guía sobre bluetooth low energy," 2024, consultado en octubre de 2024. [Online]. Available: https://www.mokoblue.com/es/guide-on-bluetooth-low-energy/

CAPÍTULO 11

Anexo

11.1. Especificaciones técnicas UWB Beacons

Tabla 11.1: Especificaciones de Estimote UWB Beacons Dev Kit

Especificación	**Descripción**
UWB Radio	Compatible con Fira y chip U1 de Apple para una precisión a nivel de pulgadas
NFC Radio	Permite obtener el identificador del beacon o abrir una app correspondiente al tocarlo
Real Time Clock (RTC)	Proporciona seguridad avanzada y optimización de energía basada en el tiempo
Inertial Sensor (IMU)	Optimiza la batería para objetos en movimiento; útil para pruebas y desarrollo
Bluetooth	Permite beaconing BLE de bajo consumo y activación segura de la radio UWB
Ambient Light Sensor	Ayuda a optimizar la vida útil de la batería en condiciones de baja luz
Batería	2 x AA con duración de hasta 2 años en modo inactivo o 4,000 minutos en UWB activo
LED Light	Parpadea para indicar proximidad, facilitando pruebas y desarrollo
Enclosure de Silicona	Resistente y reciclable, adecuado para uso en interiores y exteriores
Adhesivo Inteligente	Se adhiere a la pared y crea un puente permanente después de 45 minutos
Agujero para Montaje	Permite instalación permanente mediante un canal oculto
Peso	120 gramos en condiciones normales de humedad
Dimensiones	80 mm de alto, 100 mm de ancho y 60 mm de profundidad

Figura 11.1: UWB BEACON

11.2. Especificaciones técnicas ESP32

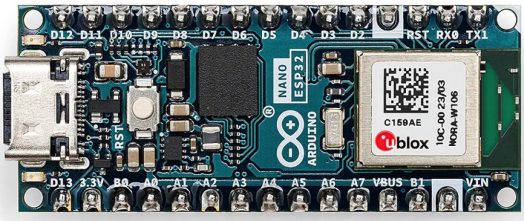

Figura 11.2: Arduino ESP32

Tabla 11.2: Especificaciones del Arduino Nano® ESP32 con cabezales

Tabla	Nombre: Arduino Nano® ESP32 con cabezales
SKU	ABX00083
Microcontrolador	u-blox® NORA-W106 (ESP32-S3)
Conector USB	USB-C®
Pines	Pin LED incorporado: 13
	Pines LED RGB incorporados: 14-16
	Pines de E/S digitales: 14
	Pines de entrada analógica: 8
	Pines PWM: 5
	Interrupciones externas: Todos los pines digitales
Conectividad	Wi-Fi®: u-blox® NORA-W106 (ESP32-S3)
	Bluetooth®: u-blox® NORA-W106 (ESP32-S3)
Comunicación	UART: 2x
	I2C: 1x, A4 (SDA), A5 (SCL)
	SPI: D11 (COPI), D12 (CIPO), D13 (SCK). Usar cualquier GPIO para Chip Select (CS)
Poder	Voltaje de E/S: 3,3 V
	Voltaje de entrada (nominal): 6-21 V
	Fuente de corriente por pin de E/S: 40 mA
	Corriente de sumidero por pin de E/S: 28 mA
Velocidad de reloj	Procesador: hasta 240 MHz
Memoria	ROM: 384 kB
	SRAM: 512 kB
	Flash externo: 128 Mbit (16 MB)
Dimensiones	Ancho: 18 mm
	Largo: 45 mm

11.3. Implementación

11.3.1. Implementación Codificada de IPS UWB

```
//  ContentView.swift
//  EstimoteAPP
//  Created by Pablo Gonzalez Santos Barrios on 20/09/24.

import SwiftUI
import EstimoteUWB

struct ContentView: View {
    @ObservedObject
    var uwb = EstimoteUWBManagerExample()

    var body: some View {
        VStack {
            // Agregar el logo arriba del texto
            Image("Imagen1")
                .resizable()
                .scaledToFit()
                .frame(width: 100, height: 100)
                .padding(.bottom, 10)

            Text("Distance 1 \(uwb.distance1)")
            Text("Distance 2 \(uwb.distance2)")
            Text("Distance 3 \(uwb.distance3)")

            HStack {
```

```
                Text("X: \(uwb.x, specifier: "%.2f")")
                Text("Y: \(uwb.y, specifier: "%.2f")")
            }
        }
        .padding()
        Text("Estimote UWB")
            .padding()
    }
}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}

class EstimoteUWBManagerExample: NSObject, ObservableObject {
    private var uwbManager: EstimoteUWBManager?
    let sensor1Id = "fd7d34c180ba4e9610a4439cd205712c"
    let sensor2Id = "8cc2e0a06f1666b863df2ea773e7ad02"
    let sensor3Id = "4056668d19dfe5fab47c9e83c82a982b"

    @Published var distance1: Float = 0.0
    @Published var distance2: Float = 0.0
    @Published var distance3: Float = 0.0
    @Published var x: Float = 0.0
    @Published var y: Float = 0.0

    override init() {
        super.init()
        setupUWB()
    }

    private func setupUWB() {
        print("Estimote setup")
        uwbManager = EstimoteUWBManager(delegate: self,
                options: EstimoteUWBOptions(shouldHandleConnectivity: true,
                                            isCameraAssisted: false))
        uwbManager?.startScanning()
    }

    func calculatePosition() {
        // Coordenadas de los sensores
        let sensor1 = (x: 0.0, y: 0.0)
        let sensor2 = (x: 4.0, y: 0.0)
        let sensor3 = (x: 2.0, y: 4.0)

        let d1 = Double(distance1)
        let d2 = Double(distance2)
        let d3 = Double(distance3)

        // Resolver para x e y
    }
}

extension EstimoteUWBManagerExample: EstimoteUWBManagerDelegate {
    func didUpdatePosition(for device: EstimoteUWBDevice) {
        let id = device.id
        switch id {
        case sensor1Id:
            distance1 = device.distance
        case sensor2Id:
            distance2 = device.distance
        case sensor3Id:
            distance3 = device.distance
        default:
            break
        }
```

```
94          calculatePosition()
95      }
96
97      func didConnect(to device: UWBIdentifiable) {
98          print("Successfully connected to: \(device.publicIdentifier)")
99      }
100     func didDisconnect(from device: UWBIdentifiable, error: Error?) {
101         print("Disconnected from device: \(device.publicIdentifier) - error: \(String
                (describing: error))")
102     }
103 }
```

11.3.2. Implementación Codificada de IPS BLE

```
1  import SwiftUI
2  import CoreBluetooth
3
4  // Variables globales para almacenar las distancias
5  var d1: Double = 0
6  var d2: Double = 0
7  var d3: Double = 0
8
9  struct Beacon: Identifiable {
10     let id = UUID()
11     var info: String
12     var distance: Double?
13 }
14
15 struct ContentView: View {
16     @StateObject private var bleManager = BLEManager()
17
18     var body: some View {
19         VStack {
20             Text("Beacon Scanner")
21                 .font(.largeTitle)
22                 .padding()
23             if bleManager.beacons.isEmpty {
24                 Text("Buscando beacons...")
25                     .font(.headline)
26                     .padding()
27             } else {
28                 List(bleManager.beacons) { beacon in
29                     HStack {
30                         Text(beacon.info)
31                             .font(.headline)
32                         if let distance = beacon.distance {
33                             Text("Distancia: \(String(format: "%.2f", distance))
                                    metros")
34                                 .font(.headline)
35                         }
36                     }
37                 }
38             }
39             // Mostrar coordenadas calculadas
40             Text("Posici n calculada: (X: \(String(format: "%.2f", bleManager.x)), Y
                    : \(String(format: "%.2f", bleManager.y)))")
41         }
42         .onAppear {
43             bleManager.startScanning()
44         }
45         .onDisappear {
46             bleManager.stopScanning()
47             bleManager.stopUpdating()
48         }
49     }
```

```
}

class BLEManager: NSObject, ObservableObject, CBCentralManagerDelegate,
    CBPeripheralDelegate {
    private var centralManager: CBCentralManager!
    private var discoveredPeripherals: [CBPeripheral] = []
    private let targetUUIDs: [CBUUID] = [
        CBUUID(string: "12345678-1234-1234-1234-123456789012"),
        CBUUID(string: "87654321-4321-4321-4321-210987654321"),
        CBUUID(string: "9abcdef0-1234-5678-9abc-def012345678")
    ]

    @Published var beacons: [Beacon] = []

    // Variables para almacenar la posicion
    @Published var x: Float = 0.0
    @Published var y: Float = 0.0

    private var timer: Timer?
    private let A: Double = -52
    private let n: Double = 3.0

    override init() {
        super.init()
        centralManager = CBCentralManager(delegate: self, queue: nil)
    }

    func startScanning() {
        if centralManager.state == .poweredOn {
            centralManager.scanForPeripherals(withServices: targetUUIDs, options: [
                CBCentralManagerScanOptionAllowDuplicatesKey: true])
        }
    }

    func stopScanning() {
        centralManager.stopScan()
    }

    func startUpdating() {
        timer = Timer.scheduledTimer(withTimeInterval: 0.1, repeats: true) { _ in
            self.updateRSSI()
        }
    }

    func stopUpdating() {
        timer?.invalidate()
        timer = nil
    }

    func updateRSSI() {
        for peripheral in discoveredPeripherals {
            peripheral.readRSSI()
        }
    }

    func calculateDistance(rssi: NSNumber) -> Double {
        let rssiValue = rssi.doubleValue
        return pow(10, (A - rssiValue) / (10 * n))
    }

    func calculatePosition() {
        // Coordenadas de los sensores
        let sensor1 = (x: 0.0, y: 0.0)
        let sensor2 = (x: 0.0, y: 2.0)
        let sensor3 = (x: 1.0, y: 2.0)

        let d1 = Double(d1)
        let d2 = Double(d2)
```

```
116         let d3 = Double(d3)
117
118         let x1 = sensor1.x
119         let y1 = sensor1.y
120         let x2 = sensor2.x
121         let y2 = sensor2.y
122         let x3 = sensor3.x
123         let y3 = sensor3.y
124
125         let A = 2 * (x2 - x1)
126         let B = 2 * (y2 - y1)
127         let C = 2 * (x3 - x1)
128         let D = 2 * (y3 - y1)
129
130         let E = d1 * d1 - d2 * d2 - x1 * x1 + x2 * x2 - y1 * y1 + y2 * y2
131         let F = d1 * d1 - d3 * d3 - x1 * x1 + x3 * x3 - y1 * y1 + y3 * y3
132
133         // Resolver para x
134         let denominatorX = A * D - B * C
135         let numeratorX = E * D - B * F
136         let x = denominatorX != 0 ? numeratorX / denominatorX : 0.0
137
138         // Resolver para y
139         let denominatorY = C * B - A * D
140         let numeratorY = E * C - A * F
141         let y = denominatorY != 0 ? numeratorY / denominatorY : 0.0
142
143         // Actualizar coordenadas
144         self.x = Float(x)
145         self.y = Float(y)
146     }
147
148     func centralManagerDidUpdateState(_ central: CBCentralManager) {
149         if central.state == .poweredOn {
150             startScanning()
151         } else {
152             print("Bluetooth no est  activado o no est  disponible.")
153         }
154     }
155
156     func centralManager(_ central: CBCentralManager, didDiscover peripheral:
        CBPeripheral, advertisementData: [String : Any], rssi RSSI: NSNumber) {
157         if !discoveredPeripherals.contains(peripheral) {
158             discoveredPeripherals.append(peripheral)
159             let beaconInfo = "Dispositivo: \(peripheral.name ?? "Sin nombre"), RSSI:
                \(RSSI)"
160             let distance = calculateDistance(rssi: RSSI)
161             let newBeacon = Beacon(info: beaconInfo, distance: distance)
162             beacons.append(newBeacon)
163             centralManager.connect(peripheral, options: nil)
164             startUpdating()
165         }
166     }
167
168     func centralManager(_ central: CBCentralManager, didConnect peripheral:
        CBPeripheral) {
169         print("Conectado a \(peripheral.name ?? "Sin nombre")")
170         peripheral.delegate = self
171         peripheral.discoverServices(targetUUIDs)
172     }
173
174     func centralManager(_ central: CBCentralManager, didFailToConnect peripheral:
        CBPeripheral, error: Error?) {
175         print("No se pudo conectar a \(peripheral.name ?? "Sin nombre"). Error: \(
            error?.localizedDescription ?? "desconocido")")
176         // Intenta reconectar
177         centralManager.connect(peripheral, options: nil)
178     }
```

```
179
180     func centralManager(_ central: CBCentralManager, didDisconnectPeripheral
            peripheral: CBPeripheral, error: Error?) {
181         print("Desconectado de \(peripheral.name ?? "Sin nombre"). Intentando
                reconectar...")
182         // Intenta reconectar al dispositivo
183         centralManager.connect(peripheral, options: nil)
184     }
185
186     func peripheral(_ peripheral: CBPeripheral, didReadRSSI RSSI: NSNumber, error:
            Error?) {
187         if error == nil {
188             if let index = discoveredPeripherals.firstIndex(of: peripheral) {
189                 let beaconInfo = "Dispositivo: \(peripheral.name ?? "Sin nombre"),
                        RSSI: \(RSSI)"
190                 let distance = calculateDistance(rssi: RSSI)
191                 beacons[index] = Beacon(info: beaconInfo, distance: distance)
192
193                 // Actualizar distancias globales
194                 switch peripheral.name {
195                 case "ESP32_Beacon":
196                     d1 = distance // Guardar la distancia en d1
197                     print("Distancia d1 actualizada: \(d1) metros")
198                 case "ESP32_Beacon_2":
199                     d2 = distance // Guardar la distancia en d2
200                     print("Distancia d2 actualizada: \(d2) metros")
201                 case "ESP32_Beacon_3":
202                     d3 = distance // Guardar la distancia en d3
203                     print("Distancia d3 actualizada: \(d3) metros")
204                 default:
205                     break // No hacer nada si no coincide
206                 }
207
208                 // Calcular posici n despu s de actualizar distancias
209                 calculatePosition()
210             }
211         } else {
212             print("Error al leer RSSI: \(error?.localizedDescription ?? "desconocido"
                    )")
213         }
214     }
215 }
```

11.3.3. Implementación Codificada de la Baliza 1 BLE

```
1  #include <BLEDevice.h>
2  #include <BLEUtils.h>
3  #include <BLEServer.h>
4
5  void setup() {
6    Serial.begin(115200);
7    BLEDevice::init("ESP32_Beacon");
8    BLEServer *pServer = BLEDevice::createServer();
9    BLEAdvertising *pAdvertising = pServer->getAdvertising();
10   BLEAdvertisementData oAdvertisementData = BLEAdvertisementData();
11   oAdvertisementData.setFlags(0x04); // BR_EDR_NOT_SUPPORTED
12   oAdvertisementData.setCompleteServices(BLEUUID("
         12345678-1234-1234-1234-123456789012"));
13   pAdvertising->setAdvertisementData(oAdvertisementData);
14   pAdvertising->start();
15   Serial.println("Beacon started!");
16 }
17
18 void loop() {
19   // No need to do anything here
```

```
}
```

11.3.4. Implementación Codificada de la Baliza 2 BLE

```
#include <BLEDevice.h>
#include <BLEUtils.h>
#include <BLEServer.h>

void setup() {
  Serial.begin(115200);
  BLEDevice::init("ESP32_Beacon_2");
  BLEServer *pServer = BLEDevice::createServer();
  BLEAdvertising *pAdvertising = pServer->getAdvertising();
  BLEAdvertisementData oAdvertisementData = BLEAdvertisementData();
  oAdvertisementData.setFlags(0x04); // BR_EDR_NOT_SUPPORTED
  oAdvertisementData.setCompleteServices(BLEUUID("
      87654321-4321-4321-4321-210987654321"));
  pAdvertising->setAdvertisementData(oAdvertisementData);
  pAdvertising->start();
  Serial.println("Beacon 2 started!");
}

void loop() {
  // No need to do anything here
}
```

11.3.5. Implementación Codificada de la Baliza 3 BLE

```
#include <BLEDevice.h>
#include <BLEUtils.h>
#include <BLEServer.h>

void setup() {
  Serial.begin(115200);
  BLEDevice::init("ESP32_Beacon_3");
  BLEServer *pServer = BLEDevice::createServer();
  BLEAdvertising *pAdvertising = pServer->getAdvertising();
  BLEAdvertisementData oAdvertisementData = BLEAdvertisementData();
  oAdvertisementData.setFlags(0x04); // BR_EDR_NOT_SUPPORTED
  oAdvertisementData.setCompleteServices("9abcdef0-1234-5678-9abc-def012345678");
  pAdvertising->setAdvertisementData(oAdvertisementData);
  pAdvertising->start();
  Serial.println("Beacon 3 started!");
}

void loop() {
  // No need to do anything here
}
```

11.3.6. Ejemplo de toma de medidas de los sistemas IPS en entorno Universitario

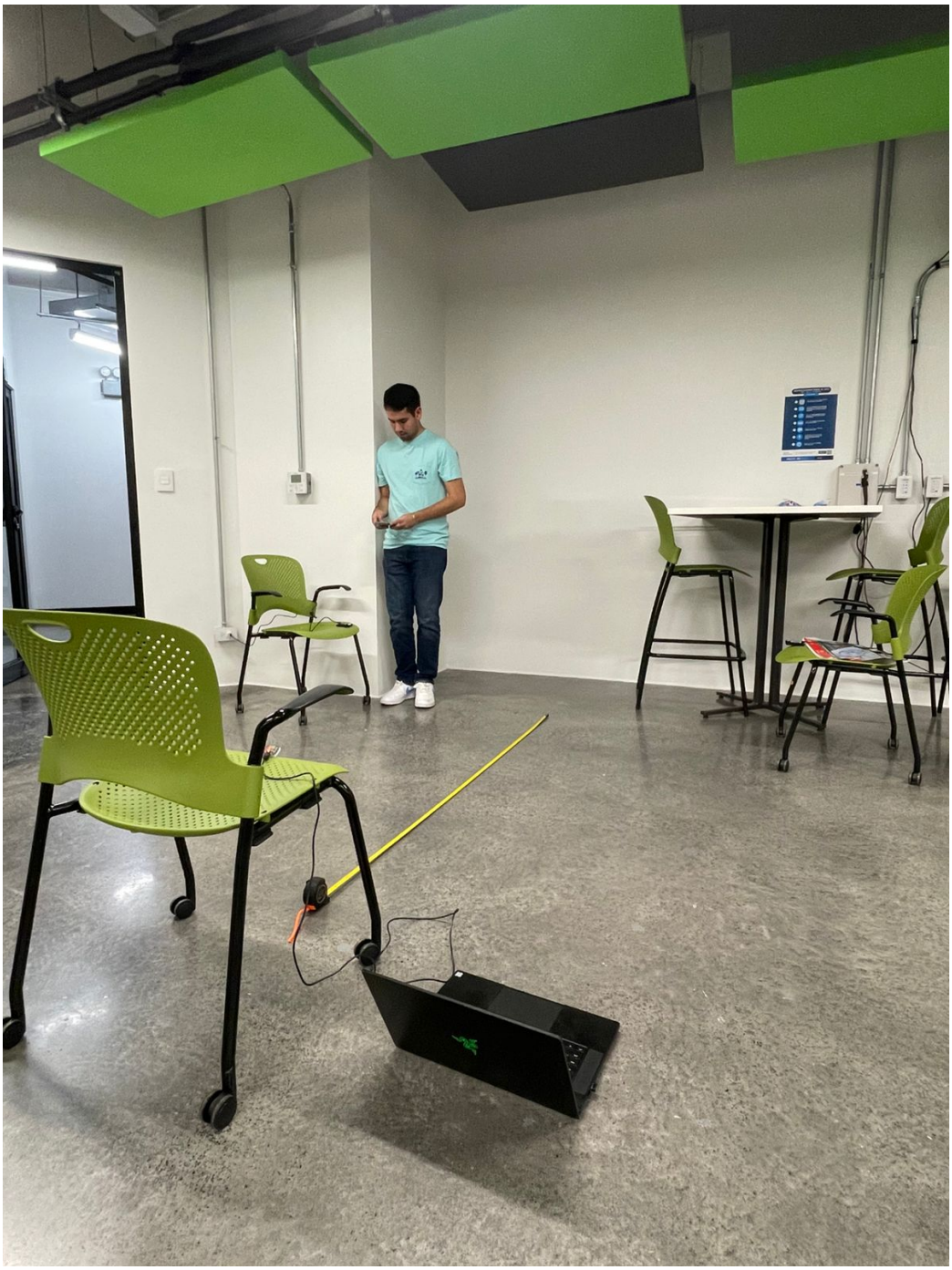

Figura 11.3: Ejemplo Toma de mediciones

11.3.7. Repositorio de Github

Este repositorio contiene las implementaciones de las aplicaciones basadas en tecnologías UWB y BLE . En este proyecto, se desarrollaron y evaluaron diferentes tecnologías de posicionamiento para recorridos virtuales, incluyendo aplicaciones para ubicaciones precisas utilizando dispositivos UWB y BLE. Puedes acceder al repositorio en el siguiente enlace: https://github.com/IPablo271/Sistemas-IPS-CIT