



REPÚBLICA DEL ECUADOR

Escuela Politécnica Nacional

" E SCIENTIA HOMINIS SALUS "

La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.
- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.
- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

Respeto hacia sí mismo y hacia los demás.

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

IMPLEMENTACIÓN DE UN ALGORITMO DE LOCALIZACIÓN EN INTERIORES PARA RESOLVER EL PROBLEMA DE UBICACIÓN DE OBJETOS USANDO TECNOLOGÍA *BLUETOOTH LOW ENERGY*

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

WILLIAM PATRICIO MOYA VELASCO

william.moya@epn.edu.ec

DIRECTOR: RAMIRO EDUARDO MOREJON TOBAR M.Sc.

ramiro.morejon@epn.edu.ec

Quito, diciembre 2019

AVAL

Certifico que el presente trabajo fue desarrollado por William Patricio Moya Velasco, bajo mi supervisión.

RAMIRO EDUARDO MOREJÓN TOBAR
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo William Patricio Moya Velasco, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

WILLIAM PATRICIO MOYA VELASCO

DEDICATORIA

A mi madre, por todo el amor, esfuerzo y sacrificio. Nada de esto sería posible sin su apoyo incondicional.

A la memoria de mi abuelito, siempre vivirá en mi corazón.

AGRADECIMIENTO

A mi madre quien fue la base fundamental durante todo este proceso, gracias por nunca dejarme solo y siempre extenderme tu mano cuando lo necesité, gracias por todo tu amor y paciencia, por enseñarme a ser honesto, trabajador y responsable siempre. Me faltan palabras para describir lo que siento y solo me queda un TE AMO mamá y lo logramos.

A mi tío Ramiro por todo el apoyo incondicional siempre, por ser el mejor ejemplo de hijo, padre, hermano. Gracias por todo tu esfuerzo y sacrificio, cada consejo, cada palabra de aliento y cada jalón de orejas.

A toda mi familia, tías, tíos, primos y a mi abuelita que siempre han estado a mi lado a pesar de la distancia y nunca han dejado de preocuparse por mí.

A mi padre que ha estado a mi lado a pesar de cualquier circunstancia.

A mi director Ingeniero Ramiro Morejón, por todo el apoyo, voluntad y sobre todo paciencia al ayudarme a culminar con éxito este proyecto.

A mis amigos Santi, Jorge, Diego y Jhony porque nada de esto hubiera sido igual sin su apoyo.

A mis amigos Carlos R. y Pablo M. quienes con su ayuda desinteresada fueron una pieza fundamental en la culminación de mi carrera.

A mis amigas Anggie, Carmita y Vale, gracias por todos los concejos, risas y apoyo que me han brindado siempre.

A Mishu quien fue un gran apoyo durante mi carrera y el desarrollo de este proyecto. Gracias por todo lo vivido, por las desveladas, las risas, la felicidad y la complicidad a lo largo de todos estos años. Siempre me llevare los mejores recuerdos. After all this time...

ÍNDICE DE CONTENIDO

AVAL	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VIII
ABSTRACT	IX
1. INTRODUCCIÓN	1
1.1 OBJETIVOS	1
1.2 ALCANCE	2
1.3 MARCO TEÓRICO.....	3
1.3.1 SISTEMAS DE POSICIONAMIENTO LOCAL.....	3
1.3.2 TECNOLOGÍAS PARA LOCALIZACIÓN EN INTERIORES.....	5
1.3.3 MÉTODOS DE LOCALIZACIÓN	14
1.3.4 TÉCNICAS DE MEDIDA DE DISTANCIA	17
1.3.5 INDICADOR DE POTENCIA DE SEÑAL RECIBIDA RSSI.....	18
1.3.6 ESPECIFICACIÓN <i>BLUETOOTH LOW ENERGY</i>	21
1.3.7 <i>BEACONS</i> Y PROTOCOLOS BLE <i>BEACON</i>	30
2. METODOLOGÍA	34
2.1 DESCRIPCIÓN DEL MÓDULO <i>BLUETOOTH LOW ENERGY</i>	34
2.1.1 MÓDULO HM-10.....	35
2.1.2 CONEXIÓN.....	37
2.1.3 ACTUALIZACIÓN <i>FIRMWARE</i>	38
2.1.4 CONFIGURACIÓN DE LOS MÓDULOS.....	40
2.2 MODELO DE PROPAGACIÓN	44
2.2.1 ADQUISICIÓN DE DATOS	44
2.2.2 AJUSTE DEL MODELO DE PROPAGACIÓN	46
2.2.3 MEJORA EN LA ADQUISICIÓN DE DATOS.....	51
2.3 IMPLEMENTACIÓN DE LA RED <i>BLUETOOTH LOW ENERGY</i>	56
2.4 IMPLEMENTACIÓN DEL ALGORITMO DE TRILATERACIÓN	58

2.4.1	TRILATERACIÓN ESFÉRICA.....	59
2.4.2	TRILATERACIÓN CIRCULAR	61
2.4.3	MÉTODO DE GAUSS NEWTON APLICADO A LA TRILATERACIÓN ESFÉRICA	61
2.4.4	IMPLEMENTACIÓN DEL ALGORITMO.....	64
2.5	IMPLEMENTACIÓN DE LA INTERFAZ GRÁFICA.....	79
2.5.1	PORTADA.....	80
2.5.2	INTERFAZ DE USUARIO	82
2.6	DESCRIPCIÓN DE ESQUEMAS DE PRUEBA.....	87
2.6.1	ESCENARIO DE PRUEBAS EN SIMULACIÓN.....	89
2.6.2	ESCENARIO DE PRUEBAS EN AMBIENTE DE PRUEBA	90
2.6.3	ESCENARIO DE PRUEBAS EN AMBIENTE INTERIOR.....	91
3.	RESULTADOS Y DISCUSIÓN	92
3.1	RED INALÁMBRICA IMPLEMENTADA	92
3.1.1	<i>BEACONS</i>	92
3.1.2	BLE CENTRAL.....	94
3.1.3	RED <i>BLUETOOTH LOW ENERGY</i>	94
3.2	RESULTADOS DE SIMULACIÓN DEL ALGORITMO.....	96
3.2.1	PRUEBA 1: ESTIMACIÓN DE LA POSICIÓN SIN ERROR EN LAS DISTANCIAS.....	97
3.2.2	PRUEBAS 2, 3 Y 4: ESTIMACIÓN DE LA POSICIÓN CON ERRORES EN LAS DISTANCIAS.	98
3.3	RESULTADOS EN AMBIENTE DE PRUEBA	102
3.3.1	PRUEBAS 1, 2 Y 3: ESTIMACIÓN DE LA POSICIÓN CON TRES, CUATRO Y CINCO <i>BEACONS</i>	102
3.4	RESULTADOS EN AMBIENTE INTERIOR.....	106
3.4.1	PRUEBA 1: ESTIMACIÓN DE LA POSICIÓN CON TRES <i>BEACONS</i>	106
3.4.2	PRUEBA 2: ESTIMACIÓN DE LA POSICIÓN CON CUATRO <i>BEACONS</i>	108
3.4.3	PRUEBA 3: ESTIMACIÓN DE LA POSICIÓN CON CINCO <i>BEACONS</i>	111
4.	CONCLUSIONES	115
4.1	CONCLUSIONES.....	115
4.2	RECOMENDACIONES	116
5.	REFERENCIAS BIBLIOGRÁFICAS.....	118

ANEXOS 122

RESUMEN

El presente proyecto presenta la implementación de un algoritmo de localización basado en trilateración y tecnología *Bluetooth Low Energy* (BLE) para la ubicación de objetos o personas en ambientes interiores. Su desarrollo se divide en dos fases: la primera define el modelo de propagación que establece la relación entre la potencia de la señal y la distancia, la segunda contempla la implementación de la red inalámbrica y el algoritmo.

Inicialmente, se hace un estudio breve sobre los módulos HM-10, los cuales permiten obtener las mediciones de potencia que requiere el algoritmo. Para definir el modelo de propagación, se analiza el comportamiento del indicador RSSI (*Received Signal Strength Indicator*) dentro de los ambientes de prueba; gracias al uso de métodos no lineales de ajuste de curvas se encuentran las constantes del modelo en base a los valores de potencia obtenidos. Entonces, se somete a los datos a un análisis estadístico para establecer un proceso de filtrado que minimice el error al estimar la distancia entre dispositivos BLE.

Por otro lado, se implementa la red inalámbrica con los módulos BLE, de los cuales se configuran cinco como *Beacon* y uno como dispositivo central móvil. A continuación, se implementa el algoritmo que recoge las mediciones de potencia, las convierte en distancia y mediante el proceso de localización basado en trilateración circular y esférica estima la ubicación del móvil dentro del área de prueba.

Finalmente, el algoritmo es evaluado en diferentes escenarios con el fin de determinar los niveles de precisión que puede alcanzar.

PALABRAS CLAVE: *Bluetooth Low Energy*, *Beacon*, Localización en interiores, RSSI, Trilateración.

ABSTRACT

This project presents the implementation of a location algorithm based on trilateration and *Bluetooth Low Energy* (BLE) technology for the location of objects or people in *indoor* environments. Its development is divided into two phases: the first defines the propagation model that establishes the relationship between the power of the signal and the distance, the second contemplates the implementation of the wireless network and the algorithm.

Initially, a brief study is made on the HM-10 modules, which allow obtaining the power measurements required by the algorithm. To define the propagation model, the behavior of the RSSI indicator (Received Signal Strength Indicator) within the test environments is analyzed; Thanks to the use of non-linear methods of adjustment of curves are the constants of the model based on the power values obtained. Then, the data is subjected to a statistical analysis to establish a filtering process that minimizes the error when estimating the distance between BLE devices.

On the other hand, the wireless network is implemented with the BLE modules, of which five are configured as *Beacon* and one as a mobile central device. Next, the algorithm that collects the power measurements, converts them into distance and through the location process based on circular and spherical trilateration, estimates the location of the mobile within the test area is implemented.

Finally, the algorithm is evaluated in different scenarios in order to determine the levels of accuracy it can achieve

KEYWORDS: *Bluetooth Low Energy*, *Beacon*, *Indoor* location, RSSI, Trilateration.

1. INTRODUCCIÓN

La localización en interiores es una técnica que pretende estimar la ubicación de personas u objetos dentro de edificios o ambientes cerrados [1]. El estudio y desarrollo de tecnologías de posicionamiento en interiores presenta en los últimos años un crecimiento considerable [2], debido principalmente a las demandas que generan día a día los usuarios por conocer con precisión y en tiempo real la ubicación de bienes o personas.

Sistemas de posicionamiento en interiores se ven necesarios en edificios, centros comerciales, centros educativos, industrias, entre otros. Los centros hospitalarios son un claro ejemplo. Con un número elevado de equipos y pacientes, la ubicación en tiempo real de las personas y bienes puede mejorar las condiciones de seguridad de los mismos. Conocer la ubicación de los trabajadores en casos de emergencia, incendios, por ejemplo, puede mejorar su seguridad y la respuesta de los equipos de rescate.

La búsqueda de rutas óptimas de evacuación en ambientes cerrados durante situaciones de emergencia se presenta como un desafío para los sistemas de posicionamiento interior [3], en el cual es necesaria la implementación de algoritmos de ubicación que trabajen con redes inalámbricas que no dependan de otras redes de telecomunicaciones, ni de la disponibilidad de la red eléctrica.

Dentro de este contexto, la importancia del presente proyecto radica en que el algoritmo implementado permite ubicar objetos o personas en ambientes cerrados, con poca carga computacional y que basa su funcionamiento en mediciones de potencia de una red inalámbrica de bajo consumo energético y completa independencia de otras redes. Además, la amplia distribución de la tecnología *Bluetooth Low Energy* en dispositivos móviles, hace que el algoritmo tenga una gran aplicabilidad en el desarrollo de aplicaciones y sistemas capaces de determinar rutas de evacuación durante emergencias.

1.1 OBJETIVOS

El objetivo general de este Proyecto Técnico es: Implementar un algoritmo de localización basado en trilateración y tecnología *Bluetooth Low Energy* para la ubicación de objetos o personas en ambientes interiores.

Los objetivos específicos de este Proyecto Técnico son:

- Estudiar el algoritmo de trilateración.

- Estudiar la especificación *Bluetooth Low Energy* y los parámetros de configuración de módulos BLE.
- Implementar el algoritmo de trilateración a partir de los datos de potencia de los módulos BLE.
- Realizar pruebas de funcionamiento del algoritmo.
- Analizar los resultados.

1.2 ALCANCE

El presente proyecto pretende implementar un algoritmo para ubicación en interiores, el cual se desarrollará alrededor de la tecnología *Beacons* BLE. El proyecto estará conformado por dos componentes: un conjunto de módulos BLE configurados como *Beacons* y principalmente el algoritmo para el análisis y presentación de la localización estimada. Debe notarse que el algoritmo desarrollado se enfocará en técnicas de medición de potencia para estimar la ubicación en interiores y aprovechar las características de los *Beacons* BLE, además, la precisión en la ubicación estará en un rango acorde al tamaño de objetos que se puedan monitorear en el ambiente de prueba, computadoras, por ejemplo.

Cada módulo BLE se configurará adecuadamente para que trabaje como un *Beacon* periférico capaz de enviar información a un dispositivo BLE central que le permitirá al algoritmo el procesamiento de los datos; esta información será principalmente un identificador de cada nodo, estados y mediciones de RSSI. Se ensamblará la topología inalámbrica en un ambiente de prueba adecuado; la topología estará conformada inicialmente por tres *Beacons* (número mínimo de dispositivos para que funcionen las técnicas de localización basadas en medición de potencia) [2] hasta un máximo de cinco, esto con el fin de que el algoritmo sea capaz de responder a las demandas de escalabilidad de las redes inalámbricas y evaluar la relación costo precisión.

El algoritmo captará toda la información receptada por el dispositivo BLE central. En función a los datos recibidos realizará dos tareas: seleccionar los *Beacons* (tres) más idóneos para la localización y determinar la ubicación del dispositivo central dentro del ambiente *indoor*. La primera tarea discriminará las señales recibidas, evaluará todas las combinaciones posibles que agrupen señales provenientes de tres *Beacons*, en base a las combinaciones generadas y sus respectivas mediciones de potencia se seleccionarán la terna de *Beacons* más adecuada para el proceso de localización. La segunda tarea se basará en una técnica

de medición de potencia para estimar la ubicación, considerando la posición de los tres *Beacons* seleccionados en la etapa previa y mediante un proceso de trilateración se determinará la ubicación estimada del dispositivo central dentro del ambiente de prueba. El algoritmo generará una ubicación en dos dimensiones del objeto dentro de un plano del ambiente de prueba. Se presentará un prototipo de la red inalámbrica junto con el algoritmo implementado para realizar las pruebas correspondientes.

Para analizar los resultados obtenidos con el algoritmo implementado, se evaluará la exactitud con la que se puede determinar la posición del dispositivo central dentro del ambiente de prueba. Para ello se realizará la variación del número de *Beacons* y sus configuraciones. Esto conlleva a los siguientes escenarios:

- Red desde tres hasta cinco *Beacons* con intervalos de transmisión mínimos y máximos.
- Red desde tres hasta cinco *Beacons* con potencia de transmisión mínima y máxima.

De esta manera se analizará la efectividad del algoritmo propuesto.

1.3 MARCO TEÓRICO

La sección detalla los conceptos, principios y tecnologías para el desarrollo del algoritmo de localización en interiores. Se abordará temas que facilitaran la configuración de los dispositivos BLE y la implementación del algoritmo de trilateración.

1.3.1 SISTEMAS DE POSICIONAMIENTO LOCAL

El crecimiento tecnológico en las comunicaciones inalámbricas y el avance de los sistemas inteligentes en los últimos años han hecho posible el desarrollo de aplicaciones y servicios, donde es primordial conocer la posición de un equipo móvil o de un usuario dentro de un área específica, la cual puede ser en un entorno cerrado (interiores) o al aire libre (exteriores).

Según [4] y [5]: un sistema de posicionamiento básico, está conformado por varias tecnologías de localización, un medio de comunicación para transmitir y recibir información y software con capacidad de procesamiento.

En función de donde se lleva a cabo la medición de la posición, estos sistemas se pueden clasificar en dos tipos:

- **Posicionamiento basado en red (remoto):** La posición se estima en un servidor remoto el cual recoge las señales enviadas desde el equipo móvil, dicho servidor procesa la información recibida y envía la ubicación al receptor móvil.
- **Posicionamiento basado en el dispositivo (Auto posicionamiento):** El equipo móvil determina su posición basado en señales recibidas de transmisores distribuidos en el ambiente de medición.

1.3.1.1 Sistemas de posicionamiento para exteriores

El sistema más utilizado para la localización en exteriores es GPS (Sistema de Posicionamiento Global), el cual utiliza una constelación de 24 satélites denominada NAVSTAR (*Navigation Satellite Timing and Ranging*), una red de computadores y el principio de triangulación para determinar la posición en función de la longitud, latitud y altitud [6] (Figura 1.1). La posición se estima en base a mediciones del tiempo de llegada de las señales al receptor móvil y la velocidad de propagación, para esto el receptor GPS debe sincronizar su reloj con al menos tres satélites de posición fija y conocida, con lo cual se determina la distancia entre el objetivo y cada uno de los satélites [7].

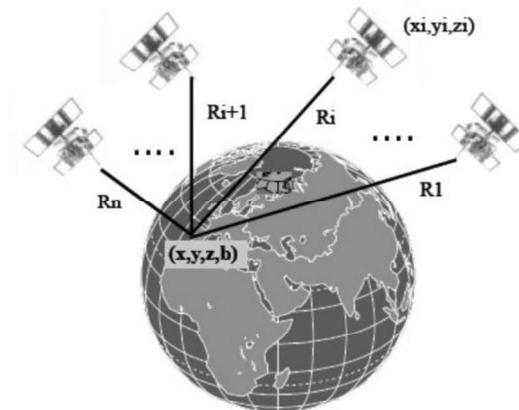


Figura 1.1. Posicionamiento GPS [6]

GPS dentro de los sistemas de posicionamiento local presenta como ventaja que las señales no son afectadas por el efecto multitrayectoria. Por lo contrario, una de sus mayores desventajas es su pobre exactitud en ambientes interiores, ya que la señal se debilita frente a grandes obstáculos naturales o arquitectónicos y no es capaz de penetrarlos; los errores de estimación de GPS están entre los 5 y 10 metros según [6] y [7], lo cual es demasiado grande para las necesidades que se requieren en interiores.

1.3.1.2 Sistemas de posicionamiento para interiores

La localización en interiores es un área de investigación que ha crecido activamente en los últimos años, el interés por sistemas y servicios con ubicaciones cada vez más precisas ha obligado la búsqueda de tecnologías más económicas y fáciles de implementar, que permitan obtener resultados menos ambiguos y más exactos; asimismo han empezado a surgir sistemas comerciales que aprovechan el conocimiento sobre la ubicación del usuario [8]. Las soluciones existentes se pueden clasificar en función de sus diferentes características, por ejemplo, los requerimientos de línea de vista, la tecnología empleada, la precisión, la cobertura o la escalabilidad. Algunas de las aplicaciones que demandan una localización precisa en interiores son:

- Vigilancia de pacientes y equipos en entornos hospitalarios.
- Localización de personas en casos de emergencias.
- Ubicación de equipos y materiales dentro de almacenes.
- Publicidad, entretenimiento.

Como se mencionó anteriormente el GPS facilita la ubicación de un dispositivo móvil en exteriores, sin embargo, no es aplicable con la misma fiabilidad en áreas interiores o con línea de vista limitada del cielo. Para poder sobrellevar la problemática anterior se han desarrollado diversos sistemas empleando un amplio espectro de tecnologías inalámbricas, considerando que la propagación de señales de radiofrecuencia en interiores se ve afectada por varios factores: efectos multitrayectoria, pérdida de línea de vista, reflexión y refracción de las señales, variación en el ambiente, temperatura, humedad, inmobiliaria, entre otras [8] [9].

1.3.2 TECNOLOGÍAS PARA LOCALIZACIÓN EN INTERIORES

La amplia distribución de las tecnologías inalámbricas en ambientes interiores y el incremento de usuarios al alcance de dispositivos móviles con conexión inalámbrica (Figura 1.2), hacen que dichas tecnologías sean consideradas como la principal solución frente a los requerimientos de los sistemas de localización en interiores. Entre los múltiples beneficios que las tecnologías inalámbricas presentan se puede destacar, su fácil despliegue, bajo costo de instalación y la gran cantidad de redes ya existentes [9].



Figura 1.2. Índices de crecimiento de usuarios de dispositivos móviles en Ecuador [10]

Algunas de las tecnologías que se pueden usar para determinar la ubicación de un usuario o dispositivo móvil en entornos cerrados son: Redes Celulares, *Bluetooth*, *ZigBee*, WiFi, entre otras. Cabe destacar que no todas las tecnologías se adecúan a las necesidades de cada aplicación, debido a los diferentes grados de precisión y cobertura que manejan [11]. En la Figura 1.3 se muestra el nivel de precisión de cada una de las tecnologías.

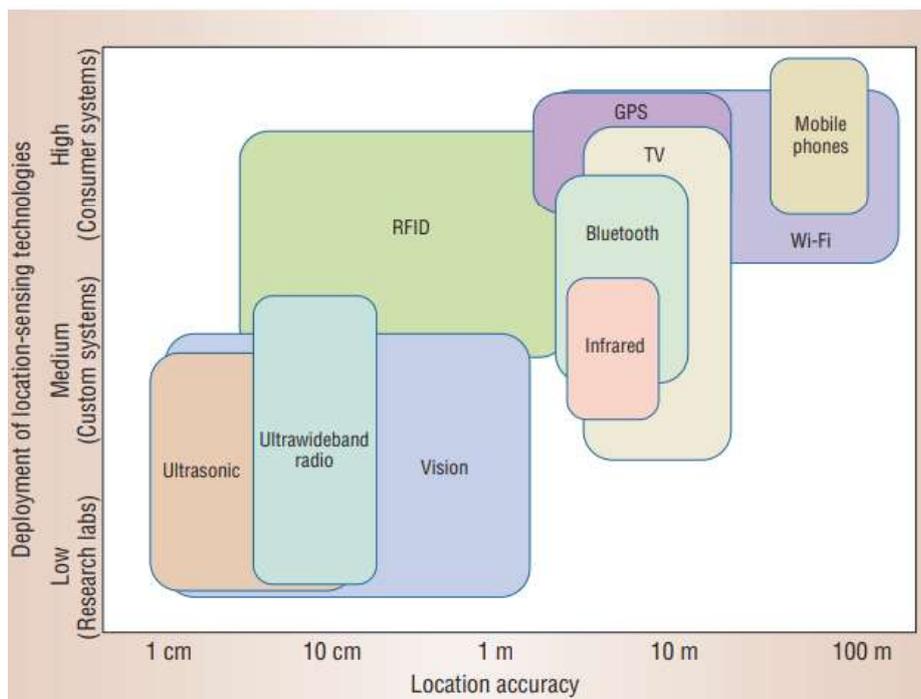


Figura 1.3. Nivel de precisión de las tecnologías inalámbricas utilizadas en sistemas de localización en interiores [11]

1.3.2.1 Infrarrojos

La tecnología basada en infrarrojos se fundamenta en la radiación de un haz de luz dentro del espectro de frecuencia infrarrojo (850 - 900 nm). La información se modula bajo un esquema de modulación de amplitud y detección directa, para ser transportada en la señal de luz desde un emisor a un receptor a una distancia relativamente corta. Las principales limitantes para estos enlaces son los obstáculos, el espacio y las distancias.

La localización mediante esta tecnología se consigue al dotar a cada usuario de un dispositivo móvil emisor de luz infrarroja, por otro lado, dentro del ambiente cerrado se instala sensores que captan la señal IR y envían los datos recibidos a un servidor que determina la ubicación de cada transmisor, de tal manera que, si el dispositivo móvil comparte la misma habitación que un receptor su posición será establecida [8] (Figura 1.4). Considerando que el rango de transmisión IR está limitado a longitudes en unidades de metros y que se requiere línea de vista despejada entre el sensor y el emisor de luz, esta localización se limita a alcances cortos, en consecuencia, el costo de implementación se vuelve significativo por el número elevado de detectores necesarios, además factores como excesiva iluminación o temperatura pueden disminuir el desempeño del sistema [8] [9].

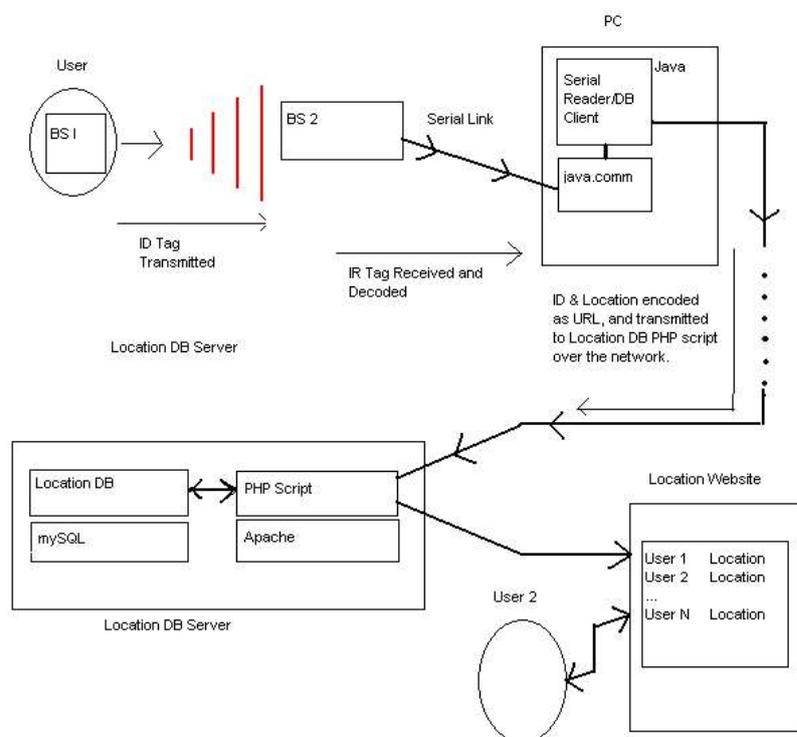


Figura 1.4. Sistema de localización basado en tecnología infrarroja [12]

1.3.2.2 Ultrasonidos

Los ultrasonidos son transmisores de señales de audio a frecuencias alrededor de los 40 kHz por encima de la máxima audible por el oído humano. Los sistemas de medición de distancia basados en ultrasonido usan la reflexión como principio básico de funcionamiento; un transmisor emite un pulso de ultrasonido que rebota sobre una superficie u objeto, de manera que la señal reflejada es detectada por un receptor ultrasónico como se presenta en la Figura 1.5.

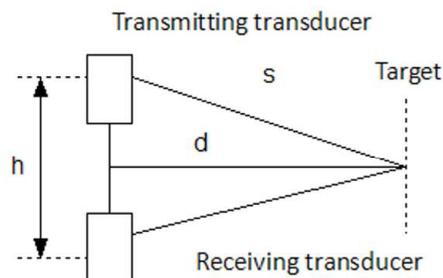


Figura 1.5. Medición de distancia con ultrasonido [13]

Los transmisores generan un pulso de muy corta duración e inicializan los relojes de los sensores, los cuales miden el tiempo que transcurre hasta que detectan la señal del emisor [8] [14]. La distancia del sensor se calcula con la información del retardo de tiempo y la velocidad de propagación del sonido a través del aire.

Las precisiones que pueden alcanzar estos sistemas son consecuencia de la densidad infraestructura y del grado de calibración de los mismos, llegando a alcanzar valores desde unos pocos metros hasta centímetros. Existen sistemas de localización como *BAT* de *AT&T*, el cual llega a tener errores máximos de tres centímetros en un 95% de las medidas [14].

1.3.2.3 RFID (*Radio Frequency Identification*)

La tecnología RFID permite identificar objetos de forma única, gracias al uso de dispositivos denominados etiquetas o *tags*; la información se transmite por medio de señales de radio a un sistema central de control.

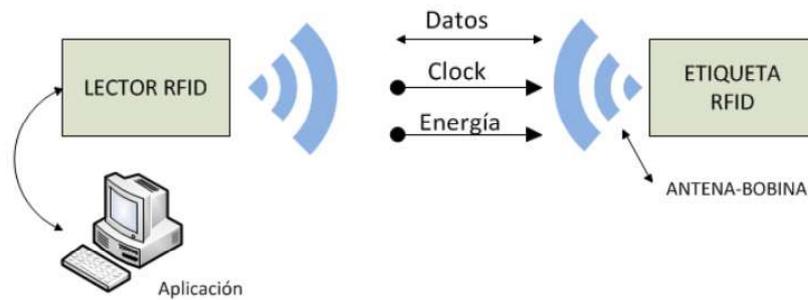


Figura 1.6. Sistema RFID [15]

Estos sistemas están conformados por tres elementos: un sistema central de control, *tags* o etiquetas y lectores, como se puede observar en la Figura 1.6 Existen dos tipos de etiquetas RFID: activas y pasivas. Las etiquetas activas son capaces de radiar su información (identificación) gracias a la presencia de una batería incorporada en su diseño, en consecuencia, su alcance llega a ser de 30 metros aproximadamente; cabe recalcar que el tiempo de vida del *tag* es proporcional al de la batería que se encuentra alrededor de los cinco años. Por lo contrario, los *tags* pasivos al no disponer de una batería propia, dependen de la señal emitida por el lector para energizar su antena y transmitir su identificación. Estas características hacen que este tipo de etiquetas sean pequeñas y de bajo costo; sin embargo, a pesar de estas ventajas, el alcance de estos *tags* es reducido, alrededor de un metro, lo cual se convierte en su principal desventaja.

Las técnicas de localización que utilizan tecnología RFID pueden usar algoritmos de proximidad, medición de potencia de la señal RFID recibida o tiempos de vuelo para determinar la posición de los objetos, en este caso las etiquetas. Para alcanzar niveles elevados de precisión en los servicios de localización con esta tecnología se requiere una gran cantidad de lectores, en consecuencia, se incrementan los costos de implementación.

1.3.2.4 ZigBee

El estándar *ZigBee* desarrollado en el año 2002 por la *ZigBee Alliance*, define un conjunto de protocolos para redes inalámbricas de radio frecuencia y baja tasa de envío de datos (máximo 250 kbps). Está basado en el estándar IEEE 802.15.4 para las capas física y MAC, además opera en las bandas de 868 MHz, 915 MHz y 2.4 GHz; es muy apropiado para el desarrollo e implementación de WSN (*Wireless Sensor Networks*). La localización basada en WSN se estima en base a mediciones de distancia o ángulos de llegada.

ZigBee como tecnología para sistemas de localización está compuesto por dispositivos de transmisión móviles de tamaño reducido (llaveros, colgantes, pulseras, etc.), un conjunto

de nodos fijos trabajando como *routers* y un dispositivo configurado como coordinador que procesara toda la información de la red (Figura 1.7), con una cobertura aproximada de 30 metros. La posición del objetivo es calculada en una central de control, dicha central recoge la información concerniente al dispositivo móvil que es captada por cada nodo fijo. El margen de error en la localización con *ZigBee*, depende en gran medida de la cantidad de nodos con que se implemente el sistema [8] [9].

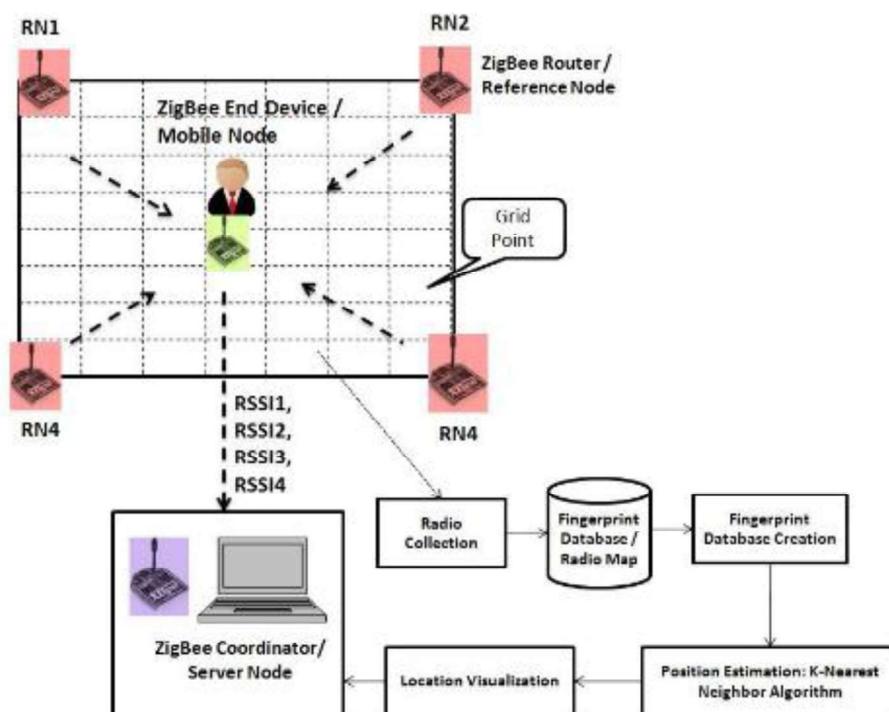


Figura 1.7. Sistema de localización *ZigBee* [16]

En el desarrollo de sistemas de localización, *ZigBee* presenta una gran ventaja gracias a su facilidad para obtener los niveles de RSSI, ya que esta información viene integrada dentro de los paquetes que se envían y no hace falta adicionar hardware alguno para ello.

1.3.2.5 WiFi (*Wireless Fidelity*)

WiFi define su funcionamiento en el estándar IEEE 802.11, con un alcance medio entre 50 y 100 metros operando en las bandas de frecuencia de 2.4 GHz y 5 GHz [17]. Esta es una de las tecnologías de radiofrecuencia con mayor despliegue en el mundo, gracias a que permite que diferentes dispositivos móviles tengan conectividad inalámbrica a internet. Su arquitectura está definida básicamente por dos equipos: Puntos de acceso (*AP's*) y Estaciones móviles (*MS's*) [9] [18]. La Tabla 1.1 resume la evolución y principales características del estándar.

Tabla 1.1. Evolución del estándar IEEE 802.11 [18]

Estándar IEEE	Año	Frecuencia [GHz]	Velocidad de transmisión máxima	Alcance máximo
802.11a	1999	5	54 Mbps	400 ft
802.11b	1999	2.4	11 Mbps	450 ft
802.11g	2003	2.4	54 Mbps	450 ft
802.11n	2009	2.4 / 5	600 Mbps	825 ft
802.11ac	2014	5	1 Gbps	1000 ft
802.11ad	2016	60	7 Gbps	30 ft
802.11af	2014	2.4 / 5	26.7 – 568.9 Mbps (Depende del canal)	1000 m
802.11ah	2016	2.4 / 5	347 Mbps	1000 m
802.11ax	2019 (Se espera)	2.4 / 5	10 Gbps	1000 ft
802.11ay	Finales de 2019 (Se espera)	60	100 Gbps	300 – 500 m

La mayoría de soluciones que utilizan WiFi para localización se basan en mediciones del nivel de potencia de la señal, comúnmente usan el identificador RSSI (Identificador de Intensidad de Señal Recibida); la estación móvil (*MS*) mide la potencia recibida de los diferentes puntos de accesos (*AP*) distribuidos en la infraestructura (Figura 1.8). Las principales ventajas del uso de esta tecnología son: su fácil implementación, despliegue, crecimiento, dispositivos móviles diseñados para trabajar con este estándar, y bajos costos. Dentro de las desventajas que presenta el uso de la señal WiFi es importante mencionar que las características de la misma se ven afectadas por factores medio ambientales, personas, obstáculos arquitectónicos e inmobiliarios que se encuentren dentro del espacio de medición, en consecuencia, la estimación de la posición puede presentar errores [17].

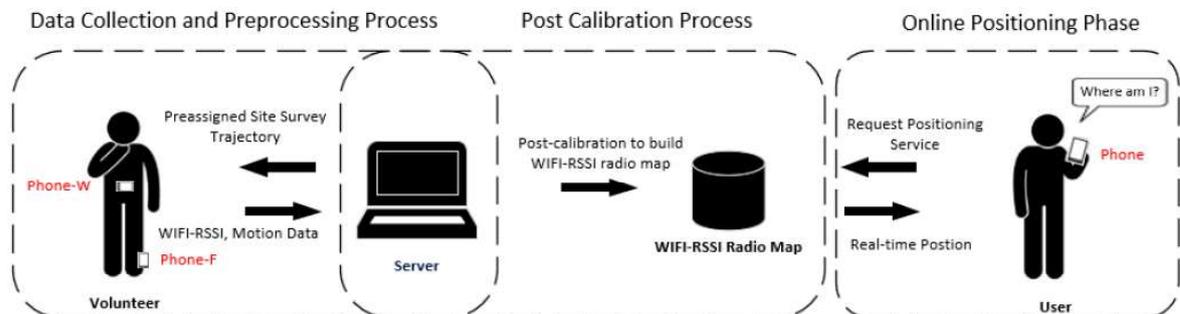


Figura 1.8. Sistema de localización WiFi [19]

Métodos de triangulación WiFi se sobrepone a la necesidad de línea de vista directa, sin embargo, se ven afectados por efectos multitraectoria generados por el mismo entorno

cerrado. La estimación se mejora usando *Fingerprint*, ya que tiene en cuenta los efectos causados por las características de los ambientes interiores citadas anteriormente. Sin embargo, no se puede dejar de lado que cualquier cambio en el medio puede causar variaciones considerables en las señales WiFi e interferir en la estimación de la ubicación en un mismo punto a lo largo del tiempo [8] [9].

1.3.2.6 Bluetooth

Bluetooth es una tecnología perteneciente al estándar IEEE 802.15.1, opera en la banda ISM (*Industrial Scientific and Medical*; 2.4 GHz) con un alcance corto dentro de un rango comprendido entre 10 y 15 metros [17]; permite transmisión de voz y datos brindando conectividad a redes personales (PAN). Este protocolo presenta como sus características más importantes: ancho de banda de 79 MHz, con espaciamiento entre canales adyacentes de 1 MHz, transmisión *half duplex*, *Frequency Hopping Spread Spectrum* (FHSS) con 1600 saltos por segundo y una velocidad de transmisión de hasta 54 Mbps en su versión 3.0 (Tabla 1.2).

Tabla 1.2. Evolución *Bluetooth* [20]

Versión de Core	Año	Mejoras
1.0	1999	
1.2	2003	<i>Adaptive Frequency Hopping, Inquiry-based RSSI</i>
2.0	2004	2.1 Mbps
2.1	2007	3.0 Mbps
3.0	2009	24 Mbps
4.0	2010	Menor consumo de energía. Protocolo <i>LE</i>

Las soluciones existentes para estimar la ubicación miden las señales de radiofrecuencia de los dispositivos móviles equipados con *Bluetooth*, tales como *tablets*, *laptops*, celulares, entre otros; la señal recibida se puede caracterizar por los parámetros LQ (*Link Quality*) y RSSI (Figura 1.9). Los métodos de triangulación y trilateración se adecúan a las características de *Bluetooth*; los niveles de potencia de varios dispositivos (RSSI), cuya posición es fija y conocida, son medidos y convertidos en longitudes, las cuales son usadas para aproximar la posición del usuario. En vista de que los rangos de trabajo de *Bluetooth* son relativamente cortos y que el cálculo de la posición depende de la relación entre el RSSI y la separación de los dispositivos, la exactitud del sistema puede estar limitada y presentar un error medio de 1,5 metros.

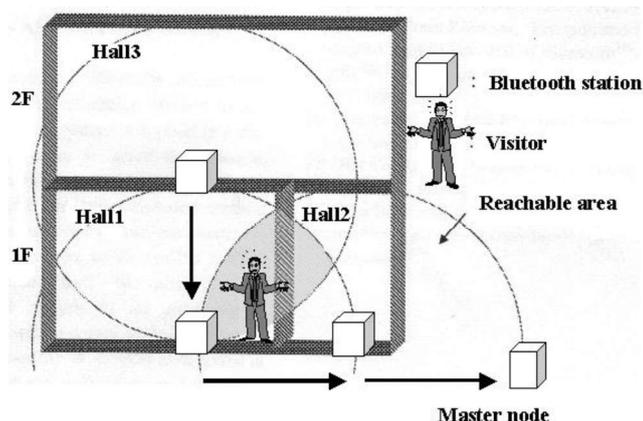


Figura 1.9. Sistema de localización *Bluetooth* [21]

1.3.2.7 Comparación de las tecnologías

WiFi, *ZigBee* y *Bluetooth* son tecnologías inalámbricas que comparten la misma banda de operación (ISM 24 GHz), además de propagar información de potencia a través del indicador RSSI; sin embargo, cabe recalcar que las capacidades de cada tecnología y su enfoque hacia la localización en interiores son diferentes. La Tabla 1.3 permite comparar las características de estas tres tecnologías.

Tabla 1.3. Comparación de Tecnologías aplicables a la localización en interiores

Característica	WiFi	ZigBee	BLE
Topología de Red	Estrella	Malla	<i>Piconet Scatternet</i>
Alcance	100 m	200 m	70 m
Tasa de datos	1 Mbps 7 Gbps	250 [kbps]	1 Mbps
Modulación	OFDM / DSSS	DSSS	GFSK
Corriente máxima en modo activo	60 mA Rx 200 mA Tx	19 mA Rx 35 mA Tx	< 15 mA
Corriente máxima en <i>standby</i>	< 100 μ A	5 μ A	< 2 μ A
Infraestructura	Dependiente	Independiente	Independiente
Soporte en <i>Smartphones</i>	Si	No	Si

Para el presente estudio se usa *Bluetooth Low Energy*; como evidencia la Tabla 1.3 existen características pertenecientes a esta tecnología que resaltan sobre las demás, las cuales se citan a continuación:

- **Consumo de energía:** En la actualidad, el consumo de energía es un factor determinante al momento de elegir una u otra tecnología. BLE es un estándar diseñado para un consumo de energía bajo, con la corriente pico en estado activo

más baja (< 15 mA). Este factor influye directamente en la elección de las baterías que alimenten los dispositivos, su duración y vida útil; en consecuencia, los costos por equipamiento y mantenimiento (cambio de baterías) serán mucho menores.

- **Soporte en Smartphones:** *ZigBee* es un estándar que no es soportado por ningún *Smartphone*, por lo contrario, BLE está disponible tanto *iOS* como en *Android*. Esto hace que esta tecnología esté ampliamente distribuida y evita que el usuario necesite dispositivos adicionales para acceder a los sistemas de localización disponibles, únicamente requiere la instalación de aplicaciones.
- **Infraestructura:** Los dispositivos BLE generalmente son de tamaño reducido, unos cuantos centímetros; su impacto dentro de la infraestructura es mínimo. Esto facilita modificaciones en la disposición física de transmisores y aumentar o disminuir su número. WiFi por otra parte está muy ligado a la infraestructura, realizar modificaciones en la disposición de los AP's es muy complejo, ya que su enfoque es brindar la cobertura y conectividad suficientes con la menor cantidad de dispositivos posibles.

En conclusión, *Buletooth Low Energy* como una tecnología para sistemas de localización en interiores, presenta considerables ventajas como: bajo costo, mínimo consumo de energía, alta disponibilidad, fácil implementación y modificación, escalabilidad, entre otras.

1.3.3 MÉTODOS DE LOCALIZACIÓN

Un sistema de posicionamiento busca principalmente estimar la posición de un dispositivo móvil dentro de un área de referencia determinada. Los niveles de precisión alcanzados por los LPS se convierten en el factor determinante a la hora de definir el tipo de aplicación que se va a implementar. Existen aplicaciones tolerantes a errores de decenas de metros; por lo contrario, existen otras que requieren conocer la ubicación exacta del móvil, además de que esta respuesta sea en tiempo real.

A lo largo del tiempo, se han estudiado, desarrollado e implementado una serie de técnicas capaces de calcular la posición de un objeto móvil, procurando responder a las necesidades de cada aplicación y siendo cada una más o menos adecuada para una situación u otra. Dentro de dichas técnicas se evalúan algunos parámetros de las señales de radiofrecuencia tales como: tiempos de propagación, potencia de la señal recibida, ángulo de llegada, entre otros.

A continuación, se analizan algunas de las técnicas más destacadas para sistemas de localización interiores.

1.3.3.1 Fingerprints (Huellas digitales)

La técnica de *Fingerprints* contrarresta los efectos causados por las múltiples trayectorias generadas en los ambientes interiores. Este algoritmo estima la localización de un dispositivo móvil al comparar un conjunto de características de las señales de radio frecuencia con una base de datos generada dentro del área de interés; los parámetros que pueden formar parte de la base de datos son la potencia, tiempos de vuelo o ángulos de incidencia, sin embargo como cita P. Romero [9] el indicador generalmente más usado es el RSSI.

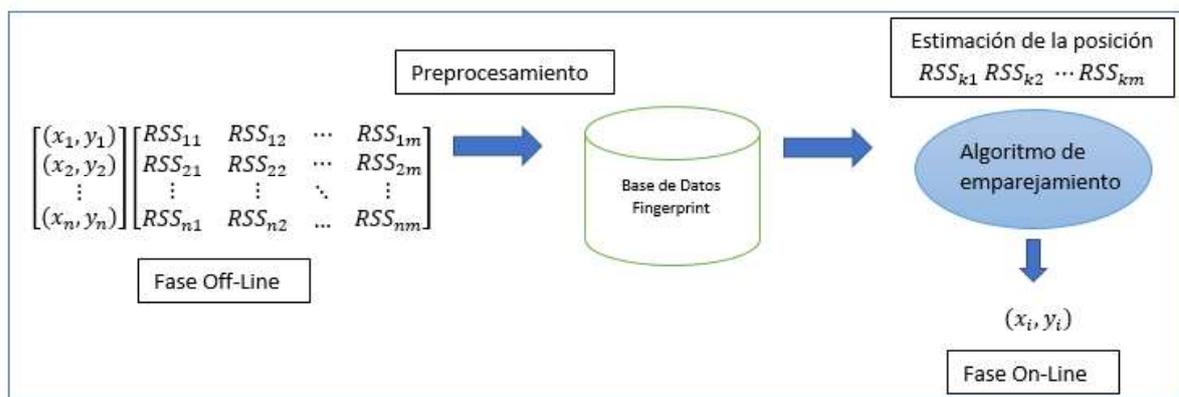


Figura 1.10. Posicionamiento con *Fingerprints* [9]

Fingerprints se ejecuta en dos etapas: fase de entrenamiento (*fase off-line*) y fase de pruebas (*fase on-line*) (Figura 1.10). En la fase de entrenamiento se mapea el área de interés ligando una posición a un patrón de potencias, formando así la base de datos y generando un mapa del área de prueba; por otro lado, en la fase de prueba se obtienen mediciones en tiempo real de los parámetros de potencia, las cuales se comparan con la base de datos existente para estimar la ubicación del dispositivo móvil asociándola con la posición almacenada que presente mayor coincidencia [7] [9].

1.3.3.2 Triangulación

Para determinar la ubicación del dispositivo móvil, este algoritmo se fundamenta en conceptos y propiedades geométricas del triángulo; en otras palabras, si se conocen la distancia entre dos puntos conocidos y los ángulos de llegada de dos señales a dichos

puntos, se puede determinar el tercer ángulo y con esta información calcular la distancia (ubicación) al punto emisor (Figura 1.11) [9] [7] [17]. La triangulación se puede realizar de dos maneras, por lateralización y una segunda forma llamada angulación. Para estimar las distancias y ángulos que requiere este algoritmo se pueden emplear varias técnicas de medición de la distancia, entre las cuales se destacan: tiempo de llegada (TOA), ángulo de llegada (AOA), diferencia de tiempo de llegada (TDOA) o intensidad de la señal recibida [8] [17].

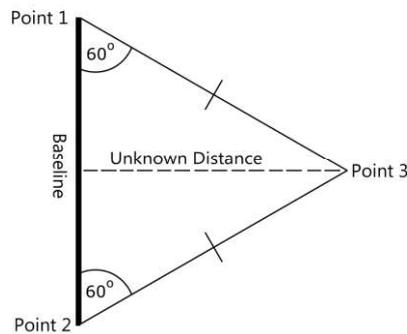


Figura 1.11. Triangulación [22]

1.3.3.3 Trilateración

Es un método análogo a la triangulación, con la cual comparte ciertos principios geométricos y algunas de las técnicas para estimar las distancias. Este algoritmo requiere conocer al menos tres puntos de referencia y la distancia de los mismos hacia el punto móvil para estimar su ubicación. Una alternativa para calcular la posición es ubicarla en el centro del triángulo de menor área que se forma al intersecar las tres circunferencias con centro en los puntos de referencia y radio la distancia entre las referencias y el móvil [7]. La Figura 1.12 ejemplifica el proceso de trilateración circular donde se tiene: un sistema con tres referencias (P1, P2, P3), un dispositivo móvil y las distancias medidas (r_1 , r_2 , r_3); las intersecciones A y B entre las dos primeras circunferencias son dos posibles ubicaciones, mientras que la posición B (intersección de las tres circunferencias) determina la ubicación del móvil.

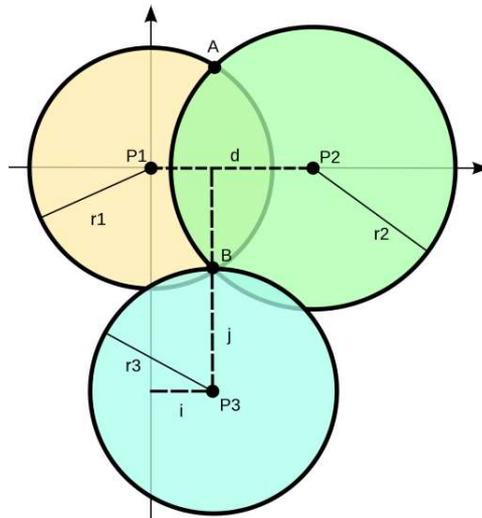


Figura 1.12. Trilateración circular [8]

Como se mencionó en el ejemplo, el método de trilateración puede hacer uso de geometría circular, sin embargo, existen variaciones en tres dimensiones como la trilateración esférica e hiperbólica. La trilateración esférica y circular se emplean en la implementación del algoritmo del presente proyecto y se explican con más detalle en la Sección 2.4.

1.3.4 TÉCNICAS DE MEDIDA DE DISTANCIA

Algunos de los métodos de localización citados en la sección anterior hacen uso de la información obtenida de la medida de distancias entre los dispositivos móviles y una serie de nodos, con ubicación fija y conocida, como referencia. Existen varias alternativas para realizar la medición de distancia entre las referencias y el equipo móvil, a continuación, se citan las técnicas que más se adecuan a ambientes interiores:

1.3.4.1 Tiempo de vuelo (*Time of Flight, TOF* o *Time of Arrival, TOA*)

Esta técnica calcula las distancias absolutas en función del tiempo que tardan las señales en viajar desde el dispositivo móvil hacia las referencias o viceversa y la velocidad de propagación en el medio, por lo cual se precisa una buena sincronización entre los equipos fijos y móviles. Se requiere al menos tres puntos de referencia para medir los tiempos y estimar la ubicación. La posición del móvil se puede determinar al calcular el punto de intersección de las circunferencias con centro en las referencias y radio igual a la distancia medida.

Cabe resaltar que para aumentar la precisión de esta técnica se podría filtrar las señales generadas por las múltiples trayectorias que pueden seguir las ondas de radiofrecuencia.

1.3.4.2 Diferencia en el tiempo de llegada (*Time Difference of Arrival, TDOA*)

TDOA al igual que TOA, es una técnica que mide tiempos de propagación, no obstante, la distancia se calcula en base a la diferencia entre los tiempos de llegada de las señales transmitidas. Se toma como fundamento el principio que establece que, por consecuencia de las diferentes distancias existentes entre las referencias y el punto de medición, los retardos medidos por el receptor son variados. En base a las medidas realizadas se trazan múltiples hipérbolas cuyo foco está ubicado en los equipos de referencia. Como muestra la Figura 1.13, la ubicación aproximada del móvil se determina por la intersección de dichas hipérbolas.

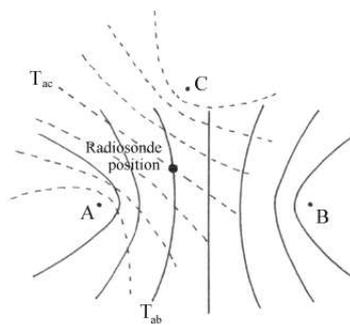


Figura 1.13. Trilateración hiperbólica [8]

1.3.4.3 Amplitud de la señal recibida

Esta técnica mide la pérdida potencia de la señal transmitida al pasar a través del medio de propagación. La distancia entre los puntos de referencia y el dispositivo móvil se calcula usando un modelo de propagación, el cual relaciona la distancia con las atenuaciones que sufre la señal, por consecuencia es necesario caracterizar el medio en el que se irradia la señal. Para estimar una ubicación con esta técnica es necesario conocer las distancias entre al menos tres puntos de referencia y el objetivo móvil, con la finalidad de aplicar cualquier método de triangulación o trilateración existente.

Generalmente se analiza una señal que presente ciertas características conocidas, la más común es el indicador de potencia RSSI (*Received Signal Strength Indicator*), que indica la potencia, usualmente en dBm, con la que llega la señal a un receptor fijo o móvil.

1.3.5 INDICADOR DE POTENCIA DE SEÑAL RECIBIDA RSSI

En el campo de las comunicaciones inalámbricas para poder medir la potencia de una señal de radio se usa el indicador RSSI, este indicador representa el nivel de potencia que puede

ser recibido por una antena [23] [24] y se mide en dBm. Como menciona V. Gao [23] RSSI es un porcentaje de intensidad de señal: cuanto mayor sea el número de RSSI, más fuerte será la señal; este valor es comúnmente definido por cada fabricante, en consecuencia, no existe una relación estandarizada entre algún parámetro físico con la lectura de RSSI. La Tabla 1.4 presenta ejemplos de los diferentes valores que puede presentar el RSSI según cada fabricante.

Tabla 1.4. Ejemplo de los valores de RSSI según fabricantes [25] [26] [27]

Fabricante	Modelo	RSSI a 1 metro [dBm]
<i>Alcatel-Lucent</i>	<i>OmniAccess Stellar</i>	-51
<i>Radioland</i>	<i>NRF51822-Beacon</i>	-54, -50, -89
<i>Nordic</i>	<i>nRF51822-Beacon</i>	-65

Considerando las variaciones que presentan las señales inalámbricas, se puede comprender el comportamiento de la medición de RSSI. De esta manera es posible conocer con relativa facilidad si un dispositivo móvil se acerca o aleja de un transmisor en función del incremento o disminución de la potencia de la señal, adicionalmente, si se conoce la relación entre el RSSI y la ubicación de un dispositivo, es factible estimar distancias con una precisión aceptable. La Figura 1.14 muestra la variación del indicador RSSI con la distancia.

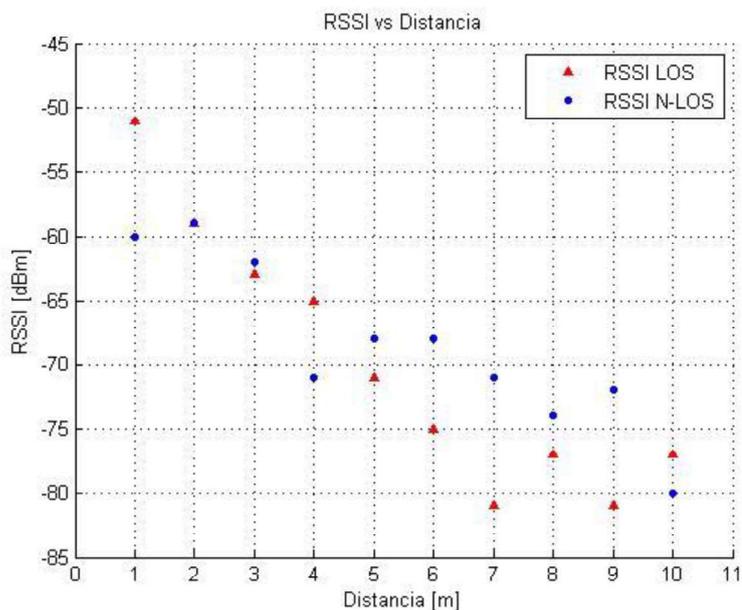


Figura 1.14. Variaciones de RSSI con la distancia

Como se ha citado anteriormente, debido a la naturaleza de las señales de radiofrecuencia, el RSSI es afectado por factores característicos de los ambientes interiores. Debido a la

influencia de dichos factores, las mediciones de potencias variaran dentro de un rango. Una manera de contrarrestar los efectos adversos del ambiente sobre las señales es diseñar, como parte del algoritmo de ubicación, un proceso de muestreo de la señal que contemple dentro de un periodo de tiempo únicamente los valores de RSSI que ocurren con mayor frecuencia, o presenten menor variación en base a un proceso estadístico de medición y calibración del ambiente [9] [23] [24] [28] [29].

Una forma de relacionar las pérdidas de potencia de las señales radioeléctricas con la distancia se define en la ecuación de Friis para transmisión en el espacio libre (Ecuación 1.1), la cual establece que la potencia de la señal disminuye con el cuadrado de la distancia [24] [30].

$$L_{bf} = 32.4 + 20 \log f + 20 \log d \quad (1.1)$$

Donde:

L_{bf} : Pérdida de transmisión en espacio libre [dB].

d : Distancia [km].

f : Frecuencia [MHz].

En diversos estudios del parámetro RSSI usando tecnologías WiFi, ZigBee y Bluetooth se ha logrado determinar de manera empírica su relación con la distancia [31]; dichos datos establecen que a una distancia d_i del nodo transmisor, la pérdida por propagación de la señal recibida es:

$$P_L(d_i) = P_L(d_o) + 10n \log_{10} \frac{d_i}{d_o} \quad (1.2)$$

Donde:

$P_L(d_o)$: Pérdida de propagación a una distancia de referencia $d_o = 1 m$.

n : Exponente de atenuación por trayectoria.

La exponente de atenuación para espacio libre es $n = 2$, sin embargo, en ambientes interiores copados de obstáculos la señal se verá afectada por reflexiones, difracciones y dispersiones, en consecuencia, se debe calcular este factor tomando como referencia un conjunto mediciones realizadas dentro del entorno. Despejando el exponente en la Ecuación 1.2, se tiene:

$$n = \frac{P_L(d_i) - P_L(d_o)}{10 \log_{10} \frac{d_i}{d_o}} \quad (1.3)$$

Entonces como establece E. Lau en [32], la potencia de la señal recibida será:

$$RSSI = -10n \log_{10} d + A \quad (1.4)$$

Donde:

n : Exponente de atenuación calculado.

d : Distancia entre el nodo transmisor y receptor [m]

A : $RSSI$ en la antena receptora a 1 metro de distancia del transmisor [dBm].

Despejando d de la Ecuación 1.4 se tiene que la distancia estimada entre nodos es:

$$d = 10^{\frac{RSSI-A}{-10n}} \quad (1.5)$$

1.3.6 ESPECIFICACIÓN *BLUETOOTH LOW ENERGY*

El *Bluetooth Special Interest Group* ha desarrollado la especificación *Bluetooth Low Energy* (BLE) 4.0 [33], 4.1 y 4.2, como una tecnología que garantice un consumo de energía bajo y menor tiempo durante el establecimiento de la conexión en comparación al *Bluetooth* clásico. El bajo consumo de energía característico de BLE se consigue gracias a que transmite cantidades pequeñas de datos, en consecuencia, sus tiempos de transmisión son muy pequeños; dentro de los objetivos de este estándar no se busca crear conexiones durante grandes periodos de tiempo ni transmitir datos a grandes velocidades, lo que permite que los dispositivos puedan estar activos únicamente cuando sea necesario.

1.3.6.1 Arquitectura

La principal topología de red que maneja esta tecnología es una *Piconet*. Una *Piconet* está formada por dos o más dispositivos que ocupan un mismo canal físico (un maestro y varios esclavos), esta topología define dos clases de dispositivos: *Master* y *Slave*. Los dispositivos *Master* son capaces de establecer varias conexiones a nivel de enlace con distintos periféricos, por lo contrario, un *Slave* solo puede mantener una conexión en capa de enlace con un único *Master*. Una gran ventaja de esta tecnología es que dispositivos configurados en estado *scanning* pueden captar datos (eventos *Advertising*) enviados por periféricos configurados en modo *broadcast* sin la obligación de crear una conexión *Master-Slave*.

El SIG de *Bluetooth* define una serie de protocolos para gestionar dispositivos, conexiones e interfaces con las aplicaciones. Como se presenta en la Figura 1.15, la pila de protocolos esta segmentada en tres grupos principales: *Controller*, *Host* y *Aplicaciones*.



Figura 1.15. Pila de protocolos BLE [33].

Controller posibilita la transmisión y recepción de las señales de radiofrecuencia, por otra parte, interpreta dichas señales como paquetes de información; está conformada por: la capa física (*Physical Layer*), modo de prueba directo (*Direct test mode*), la capa de enlace (*Link Layer*) y la interfaz de control de hardware (*Host Controller Interface*).

Host contiene los protocolos que administran la manera en la que los dispositivos se comunicaran entre sí. Esta parte de la pila de protocolos contiene: capa de control de enlace lógico y de protocolo de adaptación (*Logical Link Control and Adaptation Protocol*), administrador de seguridad (*Security Manager*), protocolo de atributo (*Attribute Protocol; ATT*), perfil de atributo genérico (*Generic Attribute Profile; GATT*) y perfil de acceso genérico (*Generic Access Profile; GAP*). Cabe resaltar que no se define ninguna interfaz superior al *Host* ya que cada entorno de desarrollo o sistema operativo maneja de manera independiente la forma de presentar la *API* (*Application Programing Interface*) de *host*.

A continuación, se describen las capas de la pila de protocolos agrupadas en *Controller*, haciendo énfasis en las capas que intervienen directamente en los experimentos y pruebas que se detallan más adelante.

1.3.6.1.1 Capa Física

La capa física es la encargada de convertir los bits en señales electromagnéticas que son irradiadas en el medio inalámbrico, transmite y recibe ondas de radio en la banda de 2,4 GHz (ISM), cuyas frecuencias van desde los 2400 MHz hasta los 2483.5 MHz. Los estándares *Bluetooth Low Energy* y *Bluetooth* clásico comparten la misma banda de operación, donde BLE define 40 canales y el *Bluetooth* clásico 79; además los dos hacen uso del mismo tipo de modulación GFSK (*Gaussian Frequency Shift Keying*) a 1 Mbps, aunque con índices de modulación diferentes. En consecuencia, estos dos estándares no se pueden comunicar entre sí, es decir son incompatibles. Sin embargo, se han desarrollado dispositivos *Dual Mode* que, al conmutar parámetros de modulación y canales de transmisión, logran trabajar con las dos tecnologías. La Tabla 1.5 hace una comparación entre *Bluetooth* clásico y BLE.

Tabla 1.5. Comparación entre *Bluetooth* clásico y BLE [33]

	<i>Bluetooth</i> clásico (BR)	<i>Bluetooth</i> (EDR)	<i>Bluetooth Low Energy</i>
Modulación	GFSK (0.28 – 0.35)	DQPSK / 8 DPSK	GFSK (0.45 – 0.55)
Velocidad de transmisión [Mbps]	1 Mbps	2 y 3 Mbps	1 Mbps
Nº de canales	79	79	40
Separación entre canales	1 MHz	1 MHz	2 MHz

La separación entre los 40 canales que utiliza el estándar BLE es de 2 MHz, los cuales se numeran del 0 al 39 y tienen un ancho de banda de 1 MHz. De los 40 canales disponibles, se usan 37 para la transmisión de datos durante la conexión y los tres restantes (canal 37, 38 y 39) transmiten únicamente paquetes de *Advertising*

RF Channel	RF Center Frequency	Channel Type	Data Channel Index	Advertising Channel Index
0	2402 MHz	Advertising channel		37
1	2404 MHz	Data channel	0	
2	2406 MHz	Data channel	1	
...	...	Data channels	...	
11	2424 MHz	Data channel	10	
12	2426 MHz	Advertising channel		38
13	2428 MHz	Data channel	11	
14	2430 MHz	Data channel	12	
...	...	Data channels	...	
38	2478 MHz	Data channel	36	
39	2480 MHz	Advertising channel		39

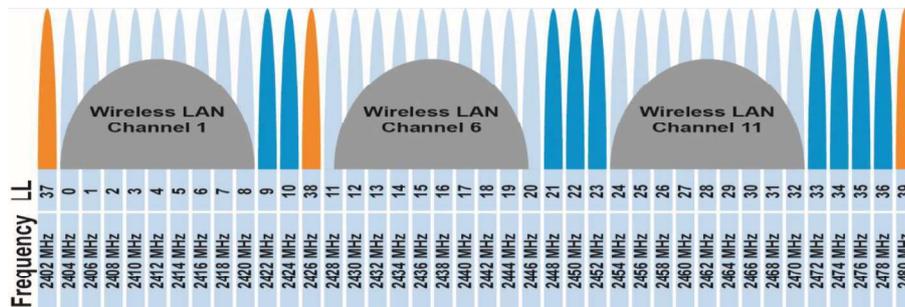


Figura 1.16. Distribución de canales BLE y espectro compartido [33]

Para evitar interferencias que podrían ser generadas por tecnologías como WiFi y ZigBee, las cuales comparten el mismo espectro, los tres canales de *Advertising* se han ubicado de una manera estratégica evitando solapamientos con canales de dichas tecnologías, como se muestra en la Figura 1.16. Por otra parte, BLE hace uso de *FHSS* (*Frequency Hopping Spread Spectrum*) para evitar de manera más eficiente las interferencias.

BLE usa una modulación *GFSK* con un producto de ancho de banda por tiempo de bit $BT=0.5$ y cuyo índice de modulación debe estar entre 0.28 y 0.35. Esta técnica representa un 1 lógico con una desviación positiva de frecuencia, y un 0 lógico con una desviación negativa de frecuencia [33]. Además, el filtro gaussiano suaviza las transiciones entre las frecuencias provocadas por la naturaleza binaria de la señal y en consecuencia disminuye el ancho de banda de la señal a transmitir [34].

La potencia de transmisión máxima y mínima definida por el estándar es de 10 mW (+10 dBm) y 0.01 mW (-20 dBm) respectivamente. El control de la potencia de salida puede ser definido localmente por el dispositivo, con el objetivo de optimizar el consumo o reducir

interferencias a otros equipos. Por otro lado, el nivel de sensibilidad debe ser menor o igual a -70 dBm.

1.3.6.1.2 Capa de enlace

Esta capa es la responsable de la creación y mantenimiento de las conexiones, de la estructura de los paquetes, y de los estados de *Scanning* y *Advertising*. Los dispositivos BLE pueden funcionar en diferentes estados, como muestra la Figura 1.17; a continuación, se describe cada estado:

- **Standby:** Este estado se asocia a un modo de ahorro de energía, durante el cual el dispositivo no transmitirá ni recibirá ningún tipo de dato. A este estado se puede acceder luego de que la conexión haya fallado o se ha perdido y no se recupere, posterior a un proceso de desconexión, al terminar un proceso de *scanning* o tras iniciar un proceso de *advertising*.
- **Initiating:** Un dispositivo central entra en este estado previo a establecer una conexión. Se mantiene a la escucha de paquetes de *advertising* provenientes de los periféricos y de ser el caso al recibir uno de estos paquetes tratará de crear una conexión con el *slave* correspondiente. Si se consigue la conexión Master - Slave, entran en un modo conectado durante el cual pueden intercambiar información a través de los canales de datos y lo harán periódicamente por medio de eventos de conexión.
- **Scanning:** Este es un estado exploratorio. Un dispositivo se mantiene buscando paquetes de *advertising* en los canales correspondientes.
- **Advertising:** Durante este estado un dispositivo periférico envía paquetes de *advertising* con cierta periodicidad dentro de los canales dedicados para las advertencias. Dependiendo de la configuración del dispositivo, este puede enviar advertencias sin solicitud alguna a todos los dispositivos adyacentes (*broadcast*) o por lo contrario solamente cuando recibe una petición por parte de un central. Este estado es crítico desde el punto de vista del consumo de energía, de manera que mientras menor sea el intervalo entre advertencias, mayor será el consumo de potencia y menor el tiempo de vida de las baterías.

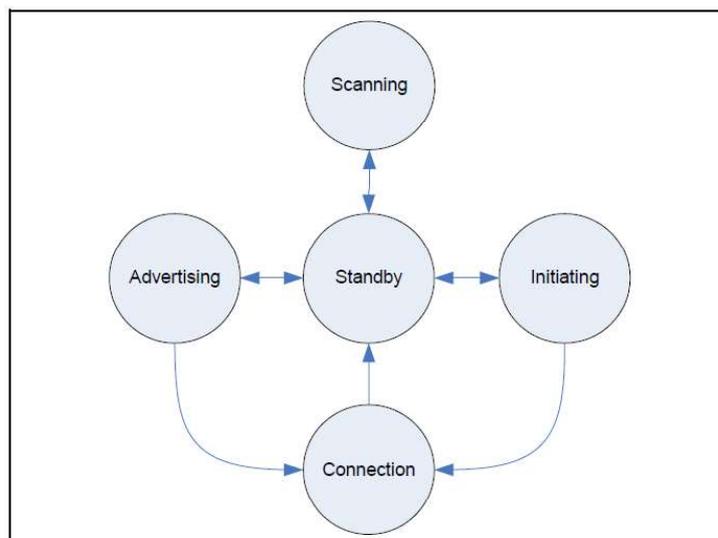


Figura 1.17. Estados de los dispositivos BLE [33]

Los estados de *advertising* y *scanning* manejan dos roles: *advertiser* y *scanner* respectivamente. *Advertiser* es el dispositivo que a través de los canales de *advertising* transmite paquetes que anuncian si es detectable y permite o no conexiones. A su vez un dispositivo con el rol de *scanner* se mantiene en los canales de *advertising* en búsqueda de paquetes de *advertising*. Dentro del modo *advertising* se pueden diferenciar cuatro tipos:

- **Connectable Undirected Advertising:** permite al *scanner* recibir anuncios o solicitar anuncios y establecer una conexión.
- **Connectable Directed Advertising:** facilita el proceso de conexión, al hacer más corto el intervalo entre advertencias. Se repite cada 3,75 milisegundos y con una estancia máxima en este estado de 1,28 s.
- **Nonconnectable Advertising:** se usa principalmente para dispositivos que únicamente desean transmitir datos (broadcast). No acepta conexiones ni recibe peticiones de información por parte de un *Master*, además el *advertiser* se mantiene como no detectable.
- **Discoverable Advertising:** en este caso el *advertiser* puede ser detectado por otros dispositivos y aceptar peticiones de información, sin embargo, no se puede establecer conexiones. Se usa también para emitir datos.

Uno de los estados más comunes en los que trabajan los dispositivos BLE configurados como *Beacons* es el de *Advertising*, razón por la cual en la Sección 1.3.6.2 se detalla su funcionamiento.

1.3.6.1.3 Host Controller Interface

En esta capa se definen las interfaces físicas que permiten la conexión entre las secciones Controller y Host. Algunas de las interfaces que se definen en la especificación son: UART (Universal Asynchronous Receiver-Transmitter), USB (Universal Serial Bus) y SDIO (Secure Digital Input Output) [33].

1.3.6.2 Estado de Advertising

En un dispositivo BLE existen dos formas para el intercambio de información, *Advertising & Scan Responses* y Conexiones.

- **Conexiones:** Cuando se ha establecido una conexión entre un dispositivo *Master* y un *Slave*, estos empiezan a intercambiar paquetes de datos de manera periódica durante intervalos de tiempo conocidos como *Connection Events*.
- **Advertising & Scan Responses:** Dentro de los paquetes de *Advertising* existen campos en los cuales se incluye la información; estos paquetes pueden ser enviados de manera periódica en modo *Broadcast* o bajo solicitud de un dispositivo central a través de un *Scan Request*, cuya respuesta puede ser un paquete de *Advertising* o un *Scan Response*, si se ha solicitado más información al *advertiser*.

1.3.6.2.1 Intervalo de Advertising

El intervalo de *Advertising* es un periodo de tiempo durante el cual los *Advertising PDU* son enviados periódicamente en los tres canales de *Advertising*. El tiempo entre el inicio de dos eventos de *Advertising* consecutivos ($T_{advEvent} = advInterval + advDelay$) [4].

Donde el retardo (*advDelay*) es un valor pseudo-aleatorio entre 0 ms y 10ms generado por la capa de enlace para cada evento de *Advertising*. La Figura 1.18 define los tiempos dentro de un evento de Advertising.

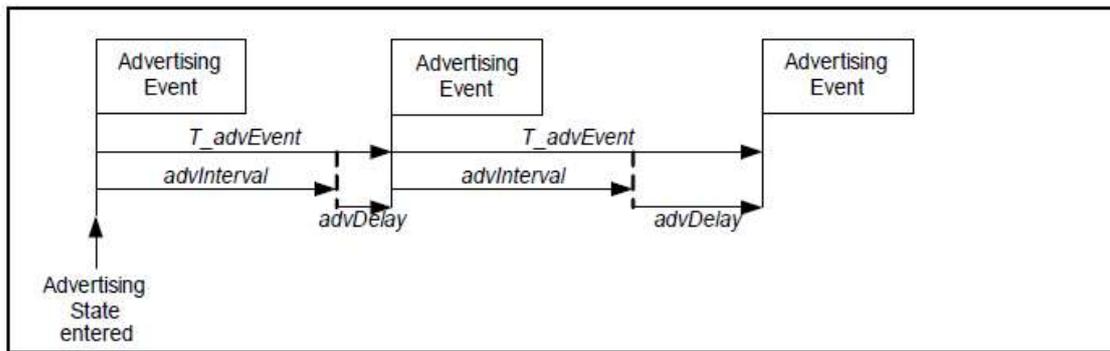


Figura 1.18. Intervalo de *Advertising* [33]

Este intervalo de *Advertising* (*advInterval*) es configurable entre 20 ms y 10.24 s, en pasos de 0.625 ms, para los modos *Connectable Undirected Advertising* y *Connectable Directed Advertising*. Por otro lado, los modos *Nonconnectable* y *Discoverable Advertising* requieren un intervalo mínimo de 100 ms. El retardo que es incluido automáticamente para reducir el número de colisiones entre *Advertisings* de diferentes dispositivos, presenta valores aleatorios entre 0 y 10 milisegundos. En consecuencia, la especificación BLE facilita a los *Scanners* el descubrimiento de *Advertisings*.

1.3.6.2.2 Paquete de datos

La capa de enlace define únicamente un formato usado tanto para paquetes de canales de datos o paquetes de canales de *Advertising*. La Figura 1.19 muestra el formato del paquete.

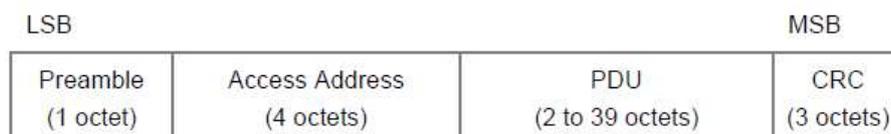


Figura 1.19. Formato del paquete de la capa de enlace [33]

- **Preámbulo:** ocho bits iniciales 01010101 o 10101010
- **Dirección de acceso:** este campo se puede usar para difundir datos o durante una conexión. Existen dos tipos: dirección de acceso de datos y dirección de acceso de *Advertising*. El primero es usado durante una conexión y tiene un valor diferente para cada conexión a nivel de la capa de enlace. El segundo es usado en los canales de *Advertising* para difundir datos, iniciar conexiones o escanear, su valor de ser siempre igual a 0x8E89BED6.

- **PDU:** *Protocol Data Unit* por sus siglas en inglés, contiene la información real que se transmite. Si el paquete se va a transmitir sobre el canal de datos es un *Data Channel PDU*, en cambio, si se transmiten por el canal de *Advertising* es un *Advertising Channel PDU*.
- **CRC:** *Cyclic Redundancy Check* de 3 bytes, es un campo que se usa para detectar cambios o errores en los datos.

1.3.6.2.3 Advertising Channel PDU

El *PDU* del canal de *Advertising* presenta una cabecera de 16 bytes y un *payload* de longitud variable, como se muestra en la Figura 1.20:

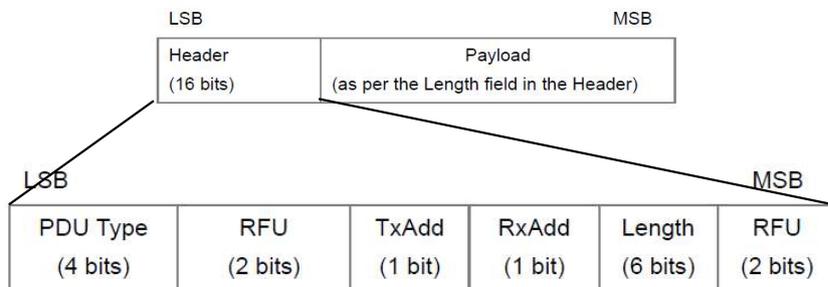


Figura 1.20. Advertising Channel PDU [33]

- **Tipo:** Define el tipo de *Advertising PDU* que se transmite.
- **RFU:** Campo reservado para uso futuro (*Reserve Future Use*).
- **TxAdd y RxAdd:** Contiene información específica sobre el tipo de PDU. Si no se define, este campo está reservado para uso futuro.
- **Longitud:** Indica la longitud de *payload* en octetos, debe ser un valor entre 6 y 37 octetos.

La Tabla 1.6 resume los tipos de paquetes de *Advertising* y sus principales características.

Tabla 1.6. Tipos de paquetes de *Advertising* y sus principales características [33]

Advertising	Escaneable	Conectable	Dirigido	Tipo de paquete
<i>Connectable Undirected</i>	Si	Si	No	ADV_IND
<i>Connectable Directed</i>	No	Si	Si	ADV_DIRECT_IND
<i>Non-Connectable Undirected</i>	No	No	No	ADV_NONCONN_IND
<i>Scannable Undirected</i>	Si	No	No	ADV_SCAN_IND

A continuación, se detalla con mayor profundidad el modo *Non-Connectable Advertising*; ya que, como se verá en secciones futuras, este modo contempla las características de funcionamiento requeridas para la configuración de los módulos BLE que se utilizan en el desarrollo de las pruebas del presente trabajo.

1.3.6.2.4 Non-Connectable Advertising

El objetivo de este evento de *Advertising* es que los dispositivos no permitan establecer conexiones entre sí, además no necesitan conocer si están siendo escaneados o no por uno o más dispositivos centrales (*Master*). Este modo básicamente transmite sus datos en *Broadcast*, su capa de enlace no puede recibir ninguna solicitud de *scanner* o inicialización de conexión. Un *Scanner* únicamente puede captar la información emitida, no puede hacer peticiones de más información ni crear conexiones.

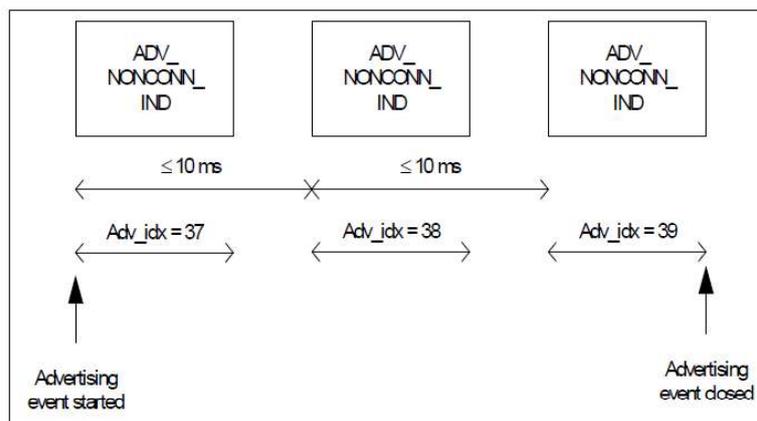


Figura 1.21. *Non-Connectable Advertising* [33]

El *advertiser* debe pasar al siguiente canal de *advertising* o cerrar el evento luego de enviar cada PDU. El tiempo entre el inicio de dos PDU *advertising non connectable* debe ser menor o igual a 10 ms, además el evento debe cerrarse dentro de este intervalo. La Figura 1.21 muestra el evento *Non-Connectable Advertising* para el caso en el que se usen los tres canales de advertencia.

1.3.7 BEACONS Y PROTOCOLOS BLE BEACON

Bluetooth Low Energy es una evolución del *Bluetooth* clásico con un enfoque hacia minimizar el consumo de energía, diseñado para desplegarse en IoT. Los dispositivos *Bluetooth LE* que apuntan al trabajo sobre esta tecnología se denominan *Bacons* BLE.

1.3.7.1 Beacon

Un *Beacon* funciona como una baliza que emite periódicamente señales de *Advertising*. El *payload* del *Advertising PDU* se crea siguiendo las pautas definidas por los protocolos *Beacon* (*iBeacon*, *Eddystone*, *AltBeacon*); dentro de estos datos se encuentra la información que permite filtrar a que protocolo pertenece la señal BLE de *Advertising* [35]. Los *Beacons* son dispositivos pequeños, baratos y generalmente su fuente de energía son baterías; pueden ser colocados de manera muy discreta en cualquier ambiente (autos, hospitales, centros comerciales, entre otros) gracias a su reducido tamaño [36].

Actualmente existe una gran variedad de empresas que desarrollan soluciones y *Beacons*, algunas de las más importantes son: *Estimote*, *kontakt*, *roximity*, *twocanoes*, *radius networks*, entre otras. La gran mayoría de estas empezaron desarrollando dispositivos que soporten el protocolo *iBeacon*, sin embargo, hoy en día el soporte está enfocado también a *Eddystone* [35]. La Figura 1.22 presenta algunos modelos de *Beacon* que se pueden conseguir en el mercado, iniciando en la esquina superior izquierda: *Gimbal*, *Kontakt.io*, *Estimote*, *Radius Networks*, *GPSshopper* y *Aruba*.



Figura 1.22. Beacons comerciales

Una gran ventaja de esta tecnología es que los dispositivos móviles, *smartphones* principalmente, vienen incorporados con el soporte para detectar los protocolos BLE.

1.3.7.2 Protocolos BLE Beacon

En esta sección se menciona previamente la existencia de protocolos *Beacon*, los cuales definen la información que se transmitirá dentro de un *Advertising PDU*. Los protocolos con mayor despliegue son: *iBeacon*, *Eddystone* y *AltBeacon*.

- iBeacon:** Protocolo desarrollado por *Apple* y presentado en el año 2013; por defecto tiene soporte de forma nativa en el sistema operativo iOS. El mayor desempeño de este protocolo, como es de esperarse, se presenta en dispositivos como *iPhones* e *iPads*, gracias a que este es su entorno de desarrollo; sin embargo, funciona de manera adecuada en otros sistemas operativos como *Android*. Además, presenta una gran ventaja por su fácil implementación y uso [35]. El intervalo óptimo de transmisión de cada *Advertising* es de 100 ms [35] [37] y el *payload* está conformado por cuatro los campos que se ilustran en la Figura 1.23:

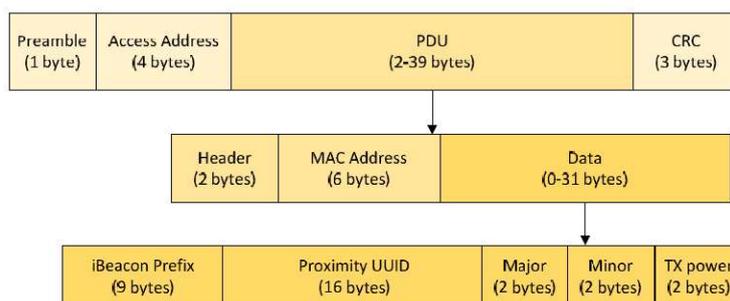


Figura 1.23. *Payload* de un *iBeacon Advertising* [28]

El uso de cada campo se resume en la Tabla 1.7.

Tabla 1.7. *Payload* de un *iBeacon Advertising* [37]

Campo	Bytes	Descripción
Prefijo iBeacon	9	Preámbulo de valor fijo que contiene información sobre el fabricante y el protocolo.
UUID	16	Identificador global del dispositivo. Definido por el desarrollador de cada aplicación. Ejemplo de un UUID de proximidad: D9B9EC1F-3925-43D0-80A9-1E39D4CEA95C
Major	2	Identificador de la zona o red local de <i>Beacons</i> . Definido por el desarrollador de cada aplicación.
Minor	2	Identificador del dispositivo dentro de la zona o red local de <i>Beacons</i> . Definido por el desarrollador de cada aplicación.
Tx Power	2	Indicador de potencia de la señal. RSSI

- Eddystone:** Google presento este protocolo a mediados del 2015, con un amplio crecimiento y aceptación en los años subsiguientes [35]. *Eddystone* presenta un funcionamiento similar a *iBeacon*, pero con la posibilidad de transmitir funcionalidades extendidas. Este protocolo presenta tres variantes: *eddytone-UID*, *eddytone-URL* y *eddytone-TLM*. UID trabaja de manera equivalente a *iBeacon*,

URL transmite una dirección URL aumentar y explicar bien y, TLM que difunde datos de telemetría sobre sensores conectados al dispositivo o información sobre el estado del mismo *Beacon* [35].

- **AltBeacon** es un protocolo abierto publicado por *Radio Networks*, con el fin de impulsar desarrollo de una industria abierta y competitiva. Al igual que *iBeacon* y *eddystone* define el formato del *advertising PDU* que difunde el dispositivo BLE. El mensaje que se transmite bajo este protocolo contiene información que permite al dispositivo receptor identificar el *Beacon* y estimar su distancia relativa [35] [38].

2. METODOLOGÍA

El presente estudio se ejecutó en dos fases. En la primera fase se emplea una metodología de tipo experimental, para determinar el comportamiento de las señales en los ambientes de prueba y de esta manera definir el modelo de propagación que permitirá establecer la relación entre la potencia de la señal y la distancia. Por otro lado, en la segunda fase se hace uso de una metodología de carácter descriptivo, para de esta manera determinar los alcances y limitaciones que presenta el algoritmo de ubicación. Además, para conocer la configuración más adecuada de los módulos BLE y que estos funcionen de tal manera que cumplan con un consumo de potencia mínimo, sin descuidar los requerimientos del algoritmo.

En este capítulo, inicialmente se indican las principales características y configuraciones de los módulos *Bluetooth LE* utilizados para la implementación de la red inalámbrica, y la transmisión y recepción de las señales. Posteriormente, se detalla el modelo de propagación, así como su influencia en la precisión final que se consigue con el algoritmo de ubicación. Finalmente se describe la implementación del algoritmo de ubicación, la interfaz gráfica y los esquemas de prueba en los que se evaluará su desempeño. El sistema descrito se ilustra en la Figura 2.1.

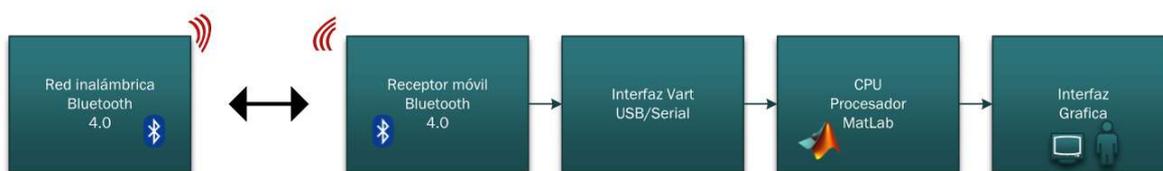


Figura 2.1. Diagrama de bloques del sistema para la implementación del Algoritmo de ubicación

2.1 DESCRIPCIÓN DEL MÓDULO *BLUETOOTH LOW ENERGY*

En la actualidad existen un sinnúmero de empresas dedicadas a la fabricación de dispositivos *Bluetooth LE*; los modelos disponibles en el mercado son muy variados, con distintas características, prestaciones y costos. Con el presente estudio se busca que el algoritmo sea capaz de funcionar de manera adecuada dentro de una red con costos de implementación y mantenimiento mínimos; en consecuencia, se ha elegido el módulo *Bluetooth 4.0 BLE module HM-10* para el desarrollo del proyecto. La Tabla 2.1 resume una comparación de algunos módulos existentes en el mercado local y su precio actual.

Tabla 2.1. Comparación de módulos BLE existentes en el mercado local

Módulo	Precio Local [USD]	Disponibilidad Local
HM-10	8	Alta
ESP-32	13	Media
AT-09	10	Media

2.1.1 MÓDULO HM-10



Figura 2.2. Módulo HM-10

Este módulo es fabricado por la empresa China *JN Huamao Technology Co., Ltd.* [39] y basa su funcionamiento en el *chip* CC2540/1 de *Texas Instrumen* [40]. Puede configurarse como *Scanner* o *Advertiser (Beacon)*, soporta los roles periférico y central, permite transmisión de datos, control remoto y adquisición de estados PIO.

Las principales características del módulo HM-10 (Figura 2.2) se detallan en la Tabla 2.2.

Tabla 2.2. Características módulo HM-10 [39]

Parámetro		Valor
Versión <i>Bluetooth</i>		V 4.0 BLE
Frecuencia		2.4 GHz
Modulación		GFSK
Potencia de transmisión		-23 dBm, -6 dBm, 0 dBm, 6 dBm
Sensibilidad		~ -90 dBm
Velocidad	Sincrónica	2-6 kbps
	Asincrónica	2-6 kbps
Seguridad		Autenticación y encriptación
Voltaje		+2.5 ~ +3.5 VDC
Corriente estado activo		8.5 mA
Corriente en ahorro de energía		50 ~ 200 μ A
Temperatura		-20 ~ +105 °C
Tamaño		27mm x 13mm x 2.2mm
<i>Chip</i>		TI CC2540/41
Memoria <i>Flash</i>		256 kbits
Protocolo <i>Beacon</i>		<i>iBeacon</i>

Los parámetros de funcionamiento del módulo son configurables por medio de comandos AT a través la interfaz UART, que por defecto presenta la siguiente configuración (*Firmware V7xx*): velocidad de transmisión 115200 bps, sin paridad, 8 bits de información y un bit de parada.

Los comandos AT son un conjunto de instrucciones que permiten consultar o configurar los parámetros del módulo a través de un microcontrolador, computador o cualquier dispositivo que disponga de comunicación serial en sus interfaces y un monitor serial (*Arduino IDE, Matlab, Hyperterminal, etc.*). El formato de estos comandos y algunos ejemplos se presentan en la Tabla 2.3 y se ilustran en la Figura 2.3.

Tabla 2.3. Comandos AT módulo HM-10 [39]

Formato		
Comando		Respuesta
Configuración	AT+COMANDO[PARAMETRO]	OK+Set: [PARAMETRO]
Consulta	AT+COMANDO?	OK+Get: [PARAMETRO]
Ejemplo		
Comando		Respuesta
Configuración	AT+NAMEEPN1	OK+Set:EPN1
	AT+ROLE1	OK+Set:1
Consulta	AT+NAME?	OK+Get:EPN1
	AT+ROLE?	OK+Get:1



Figura 2.3. Comandos AT módulo HM-10 (Monitor serial Arduino IDE)

Todos los comandos, parámetros y configuraciones disponibles se detallan en el *datasheet* del fabricante [39].

2.1.2 CONEXIÓN

Para poder configurar y actualizar los módulos es necesario conectarlos a un dispositivo que permita establecer comunicación serial en alguna de sus interfaces. Actualmente la mayoría de dispositivos únicamente vienen equipados con puertos USB, por este motivo es obligatorio el uso de un módulo que facilite la conversión UART a USB. La Figura 2.4 muestra el esquema de conexión al usar un módulo Arduino nano para comunicar al dispositivo BLE con el puerto USB de un computador.

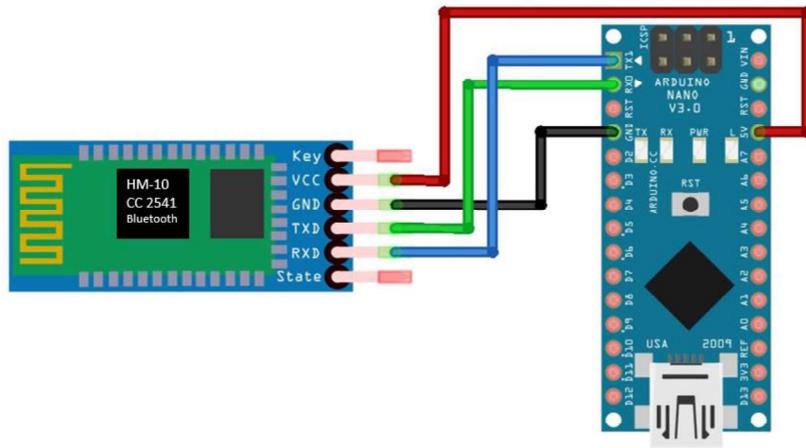


Figura 2.4. Conexión módulo HM-10 con Arduino Nano

Este esquema de conexión se implementa para la configuración y actualización de todos los módulos BLE (*Beacons* y central). Es importante mencionar que el dispositivo BLE central, al ser el encargado de captar las señales que provienen de cada *Beacon* y transmitir las al algoritmo, mantiene esta conexión de manera permanente. Por otro lado, los *Beacons* una vez configurados, únicamente necesitan ser energizados (Figura 2.5).

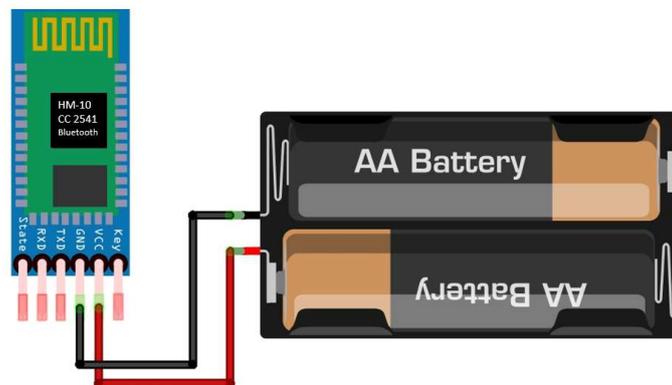


Figura 2.5. Conexión *Beacon*

Cada módulo configurado como *Beacon* se alimenta con un par de pilas convencionales triple A, garantizando de esta manera que el costo de implementación sea bajo. Las Tablas

Tabla 2.4 y Tabla 2.5 resumen el consumo de corriente del módulo con diferentes potencias e intervalos de transmisión, tanto para estado activo como inactivo; por otra parte, la

Tabla 2.6 establece una comparación del costo y tiempo de duración de diferentes tipos de baterías, lo que respalda la elección de las pilas AAA como fuente de alimentación de los *Beacons*. A pesar de que existen baterías con mayor capacidad y duración, su costo también es elevado, además que debido al mínimo consumo que representa cada módulo, estas se verían subutilizadas presentando un desperdicio de recursos.

Tabla 2.4. Consumo de corriente en estado activo módulo HM-10

Corriente Estado Activo [mA]		Potencia [dBm]	
		-23	6
Intervalo de advertencia [ms]	100	8.9775	9.1625
	546.25	8.845	9.225
	1022.5	8.9525	9.0875
	7000	8.75	9.0975

Tabla 2.5. Consumo de corriente en estado inactivo módulo HM-10

Corriente Estado Inactivo [uA]		Potencia [dBm]	
		-23	6
Intervalo de advertencia [ms]	100	150	157
	546.25	153	154
	1022.5	148	152
	7000	151	132

Tabla 2.6. Comparación de baterías

Batería	Capacidad nominal [mAh]	Costo [USD] c/u	Duración media [días]	
			Activo	Inactivo
AAA	2000	0,5	9,19	542,89
BL-5CB	800	8	3,69	217,16
Litio 18650	4200	4	19,40	1166,67
Lipo Turnigy	1200	14	5,60	353,36

2.1.3 ACTUALIZACIÓN *FIRMWARE*

Los módulos por defecto vienen cargados con una versión de *Firmware*, la cual no necesariamente es la misma en todos los casos. Para garantizar que las configuraciones y funcionamiento de todos los dispositivos son iguales y compatibles, se debe verificar que

el *Firmware* cargado en cada módulo esté en la última versión disponible V700; en el caso de que esto no se cumpla, es necesario actualizar el dispositivo. La consulta de la versión se realiza a través del comando: AT+VERR?, cuya respuesta será la versión actual disponible en el módulo (Figura 2.6) [41].



Figura 2.6. Consulta de versión de *Firmware*

El proceso de actualización de *Firmware* está definido por el mismo fabricante [41] y describe los siguientes pasos:

- Comprobar que el módulo trabaja normalmente, tras lo cual se debe terminar cualquier conexión que se haya establecido con el mismo.
- Conectar el módulo a un computador y verificar que existe comunicación a través del monitor serial.
- Activar el modo de actualización en el módulo. A través del monitor serial enviar el comando AT+SBLUP, si el módulo responde OK+SBLUP ha entrado en modo actualización (Figura 2.7).



Figura 2.7. Módulo en modo de actualización

- Descargar los archivos de actualización directamente de la página oficial del fabricante. Estos archivos son HMSoft.bin (*Firmware*) y HMSoft.exe (asistente de actualización).
- Ejecutar el archivo HMSoft.exe, donde se debe seleccionar el puerto al que está conectado el módulo e ingresar el directorio donde está ubicado el archivo HMSoft.bin (Figura 2.8). Finalmente cargar la imagen.

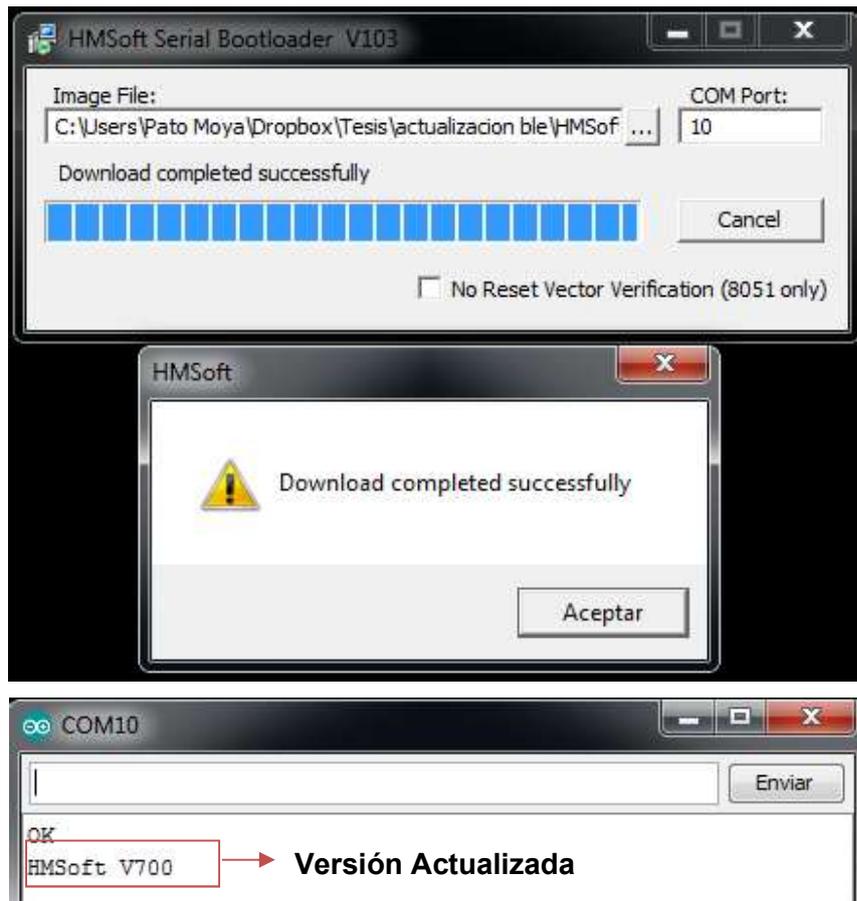


Figura 2.8. Actualización de *Firmware*

Una vez actualizados los módulos, se pueden configurar con las características de funcionamiento que demandan la red inalámbrica y el algoritmo.

2.1.4 CONFIGURACIÓN DE LOS MÓDULOS

Los módulos se configuran en dos modos: BLE central (*scanner*) y *Beacon (advertiser)*. El BLE central está encargado de buscar y recopilar los paquetes de *advertising* generados por los *Beacons*, los cuales a su vez transmitirán periódicamente dichos paquetes.

2.1.4.1 Configuración del dispositivo BLE central

En este modo el dispositivo será capaz de buscar los paquetes de dentro de los canales de *advertising*. Una vez que recibe los datos, estos se transmiten al algoritmo de ubicación, el cual procesa y convierte la información en valores potencia y distancia. Dentro de la red inalámbrica, este módulo es el dispositivo móvil del cual se estimará la ubicación. El proceso de configuración de este modo se detalla en la Tabla 2.7.

Tabla 2.7. Configuración del modo BLE central [39]

Comando	Respuesta	Descripción
AT	OK	Verifica la conexión serial
AT+RENEW	OK+RENEW	Restablece todos los valores de fabrica
AT+RESET	OK+RESET	Reinicia el módulo
AT	OK	Verifica que el módulo reinició correctamente
AT+NAMECENTRAL	OK+Set:CENTRAL	Define el nombre del módulo. Nombre con el cual se podrá encontrar en otros dispositivos
AT+ROLE1	OK+Set:1	Define el rol del módulo. 0: Periférico 1: Central
AT+IMME1	OK+Set:1	Define el modo de escaneo. 0: Escaneo automático 1: Escaneo por comando
AT+RESET	OK+RESET	Reinicia el módulo para efectuar cambios en las configuraciones.

Cuando el módulo es configurado como *central-scanner*, este puede trabajar de dos maneras: escaneo automático y por comando. En el modo automático, una vez el módulo enciende, este revisa las direcciones de los dispositivos con los cuales tuvo conexiones exitosas. Si existen, el módulo intentara conectarse con estos dispositivos para siempre. Si no tiene direcciones almacenadas, entonces tratara de conectarse a los dispositivos que encontró durante el proceso de escaneo. Por su parte en el modo de escaneo por comando, el dispositivo espera por la orden para buscar dispositivos de *advertising (Beacons)* y no es necesario establecer conexiones para acceder a la información. Los comandos para iniciar un proceso de búsqueda son: AT+DISC?, AT+DISI? o AT+DISA?. El comando que se usa en la implementación del algoritmo es: AT+DISI?, debido a que su respuesta contiene el formato completo del advertising PDU definido por el protocolo *iBeacon* [39].

En este caso el módulo se configura como central con escaneo por comando (AT+IMME1), ya que con el fin de disminuir el consumo de potencia en los dispositivos no se requiere establecer conexiones entre ellos. La Figura 2.9 presenta la configuración de un módulo en modo BLE central.

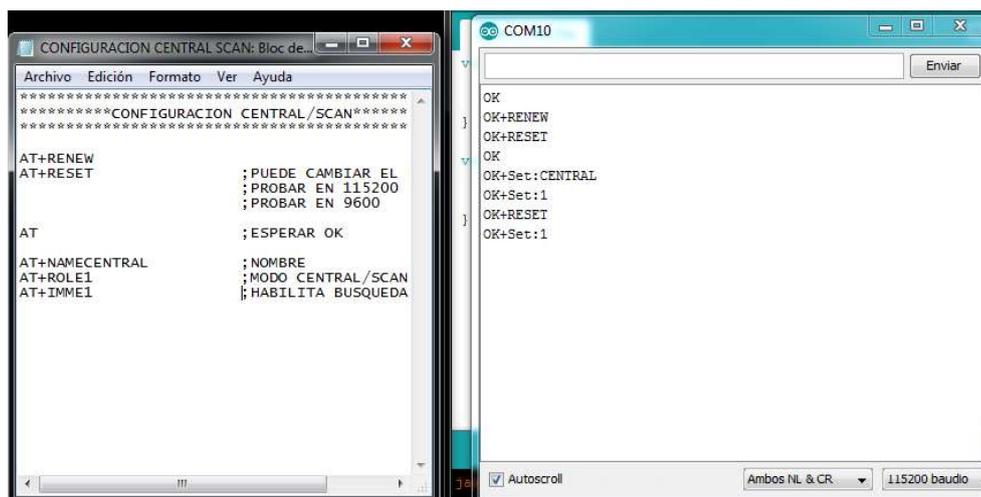


Figura 2.9. Configuración de un módulo en modo BLE central

2.1.4.2 Configuración del dispositivo *Beacon*

En esta configuración los dispositivos transmitirán paquetes de *advertising* de manera periódica cada cierto tiempo, sin esperar por una petición de información o establecer conexiones con un módulo central. El formato del PDU de *advertising* está definido por el protocolo *iBeacon*, que es el cual manejan estos módulos. En este modo de funcionamiento los parámetros más importantes son la potencia de transmisión y el intervalo de *advertising*. Esta configuración se detalla en la Tabla 2.8.

Tabla 2.8. Configuración del modo *Beacon* [39]

Comando	Respuesta	Descripción
AT	OK	Verifica la conexión serial
AT+RENEW	OK+RENEW	Restablece todos los valores de fabrica
AT+RESET	OK+RESET	Reinicia el módulo
AT	OK	Verifica que el módulo reinició correctamente
AT+NAMEEPN1	OK+Set: EPN1	Define el nombre del módulo. Nombre con el cual se podrá encontrar en otros dispositivos
AT+MARJ0x0001	OK+Set:0x0001	Define el identificador de Zona. Zona 1
AT+MINO0x0001	OK+Set:0x0001	Define el identificador dentro de la Zona. Dispositivo 1 de la zona 1
AT+ADVI0	OK+Set:0	Define el intervalo en el que se generan las advertencias en milisegundos. 0: 100ms; 1: 252.5ms; 2: 211.25ms; 3:318.75ms; 4: 417.50ms; 5:546.25ms; 6: 760.00ms; 7:852,50ms; 8: 1022.5ms; 9:1285ms; A: 2000ms; B: 3000ms C: 4000ms; D: 5000ms; E: 6000ms; F: 7000ms
AT+POWE0	OK+Set:0	Define la potencia de tx: 0: -23 dBm; 1: -6 dBm; 2: 0 dBm; 3: 6 dBm
AT+ADTY3	OK+Set:3	Establece el modo de <i>advertising</i> : 0: Connectable undirected

		1: Connectable directed 2: Discoverable 3: Non-Connectable
AT+IBEA1	OK+Set:1	Habilita el protocolo <i>iBeacon</i> : 0: Desactivado 1: Activado
AT+DELO2	OK+DELO2	Define el modo de despliegue: 0: Broadcast y scanning 2: Solamente broadcast
AT+PWRM0	OK+Set:0	Modo de ahorro de energía: 0: Activado 1: Desactivado
AT+RESET	OK+RESET	Reinicia el módulo para efectuar cambios en las configuraciones.

Esta configuración busca hacer que el dispositivo consuma la mínima cantidad de energía posible. El modo ahorro de energía hace que el módulo transmita *advertisings* únicamente cuando es necesario, además, al desplegar únicamente *broadcast* se evita iniciar procesos de *scanning* innecesarios. Como se define en la sección 1.3 el modo *non connectable advertising*, garantiza que el dispositivo únicamente transmitirá información sin establecer conexiones ni responder a solicitudes de *scanning*, minimizando aún más el consumo de energía. Los parámetros de potencia de transmisión e intervalos de *advertising* se modificarán para evaluar el desempeño del algoritmo en conjunto con la red inalámbrica. La Figura 2.10 presenta la configuración del módulo HM-10 como un *Beacon* BLE.

```

CONFIGURACION iBEACON: Bloc de notas
Archivo Edición Formato Ver Ayuda
AT+RENEW
AT+RESET          ; PUEDE CAMBIAR EL
                  ; PROBAR EN 115200
                  ; PROBAR EN 9600

AT                ; ESPERAR OK

AT+NAMEEPN1      ; NOMBRE
AT+MARJ0x0001    ; ID DE ZONA
AT+MINO0x0001    ; ID DENTRO DE ZONA

AT+ADVIO         ; t(ADVERTENCIA) 10
AT+ADVI5         ; t(ADVERTENCIA) 54
AT+ADVIF         ; t(ADVERTENCIA) 70

AT+POWE0         ; POTENCIA TX -23dB
AT+POWE2         ; POTENCIA TX 0dBm
AT+POWE3         ; POTENCIA TX 6dBm

AT+ADTY3         ; NO CONECTABLE
AT+IBEA1         ; ACTIVAR MODO iBEA
AT+DELO2         ; SOLO BROADCAST
AT+PWRM0         ; ACTIVAR AUTO-SLEE
AT+RESET

COM10
Enviar
OK+RENEW
OK+RESET
OK
OK+Set:EPN1
OK+Set:0x0001
OK+Set:0x0001
OK+Set:0
OK+Set:0
OK+Set:3
OK+Set:1
OK+DELO2
OK+Set:0
OK+RESET
Autoscroll Ambos NL & CR 115200 baudio

```

Figura 2.10. Configuración de un módulo en modo *Beacon* BLE

2.2 MODELO DE PROPAGACIÓN

Para definir la relación entre la distancia y la potencia de la señal existen diversos modelos de pérdidas por propagación. Como se cita en [29] y según lo planteado en la Sección 1.3.5, el modelo de pérdidas de propagación *log-distance* (Ecuación 1.2) es el que más se ajusta a las características de las señales transmitidas por los dispositivos BLE. La Ecuación 2.1 se presenta como una variante al modelo definido en la Ecuación 1.2 e incorpora la variable X_σ , ruido gaussiano aleatorio con media 0 y distribución estándar σ [29]; esta variable ayuda a describir de una manera empírica la inestabilidad de las mediciones de RSSI afectadas por los efectos de reflexión, difracción y múltiples trayectorias [29].

$$RSSI = RSSI_0 - 10n \log_{10} d + X_\sigma \quad (2.1)$$

Entonces, la distancia en función de la medida de potencia se define como:

$$d = 10^{\frac{RSSI_0 - RSSI + X_\sigma}{10n}} \quad (2.2)$$

Es importante establecer que las mediciones de potencia se deben realizar a distancias $d > d_0$ (1 metro), de tal manera que la diferencia entre la potencia recibida ($RSSI$) y la potencia medida a un metro ($RSSI_0$) sea diferente de cero.

2.2.1 ADQUISICIÓN DE DATOS

Con el objetivo de definir un modelo relativamente preciso, es fundamental realizar una amplia adquisición de datos y su análisis estadístico. Las variables que se miden para determinar el modelo son la distancia (d) y la potencia recibida ($RSSI(d)$); este conjunto de datos permite, a través de un algoritmo de ajuste de curvas no lineales, encontrar los valores óptimos para la potencia de referencia ($RSSI_0$) y el exponente de pérdida por trayectoria (n).

La adquisición de datos se realiza en dos ambientes de prueba, que se detallan con más precisión en la descripción de los esquemas de prueba (Sección 2.6). El primero es un ambiente sin obstáculos que afecte en lo más mínimo a las señales, es decir, se elimina cualquier obstáculo con el objetivo de reducir reflexiones, difracciones y múltiples trayectorias. Por otra parte, el segundo ambiente es un espacio interior real con la presencia de paredes, ventanas y mueblería. Para ambos casos se toman 50 mediciones de potencia por cada posición y con cada *Beacon*, los puntos de medición se posicionan

en línea recta con una separación de un metro entre cada uno de ellos a lo largo de 10 metros (Figura 2.11).

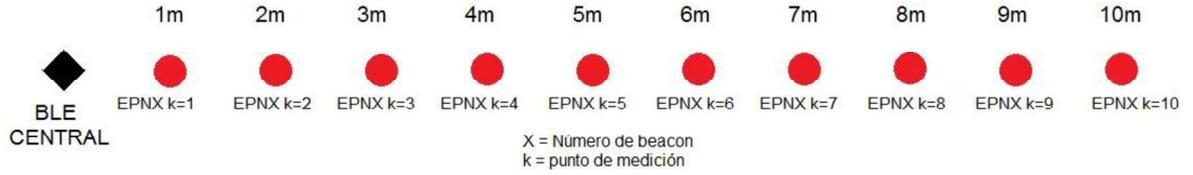


Figura 2.11. Distribución de los puntos de medición

El conjunto de datos obtenido por cada medición se define como:

$$RSSI_EPNX_K = [RSSI_1, RSSI_2, \dots, RSSI_{50}] \quad (2.3)$$

Todos los datos de cada posición K se reducen a un solo valor de potencia; de esta manera se genera los conjuntos *RSSIM* (mediana; Ecuación 2.4), *RSSIP* (media; Ecuación 2.5) y *RSSIMD* (moda; Ecuación 2.6) en función de la medida estadística de tendencia central que se aplique a cada conjunto de medidas, esto con el fin de evaluar la manera más viable de procesar la información para determinar los coeficientes del modelo de propagación.

$$RSSIM_EPNX = [RSSIM_{K=1}, RSSIM_{K=2}, \dots, RSSIM_{K=10}] \quad (2.4)$$

$$RSSIP_EPNX = [RSSIP_{K=1}, RSSIP_{K=2}, \dots, RSSIP_{K=10}] \quad (2.5)$$

$$RSSIMD_EPNX = [RSSIMD_{K=1}, RSSIMD_{K=2}, \dots, RSSIMD_{K=10}] \quad (2.6)$$

Los valores agrupados de *RSSIM*, *RSSIP* y *RSSIMD* con respecto a cada posición K, son presentados en la Figura 2.12.

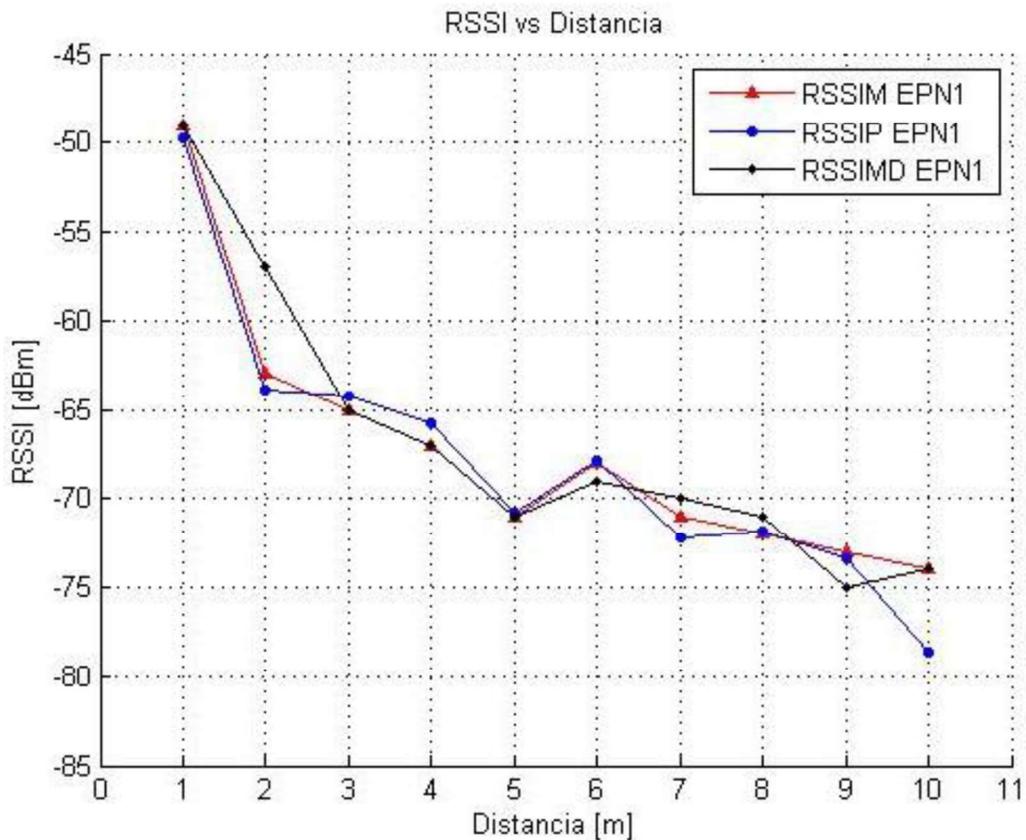


Figura 2.12. Mediana, media y moda de los valores RSSI medidos

2.2.2 AJUSTE DEL MODELO DE PROPAGACIÓN

Un factor preliminar para definir qué conjunto de datos es el más adecuado para relacionar las mediciones de potencia con la distancia, es la correlación existente entre estas variables. Para determinar este valor, se hace uso del coeficiente de correlación de Pearson (Ecuación 2.7). Este coeficiente se define dentro de un rango entre -1 y 1, siendo respectivamente valores de correlación máxima inversa y directa, conforme este coeficiente se acerca a cero la correlación existente entre las variables tiende a ser nula.

$$r_{XY} = \frac{cov(XY)}{\sigma_X \sigma_Y} \quad (2.7)$$

Donde:

$cov(XY)$: es la covarianza entre X ($RSSIM$, $RSSIP$ o $RSSIMD$) y Y (distancia entre el Beacon y el punto k).

$\sigma_X \sigma_Y$: son las desviaciones estándar de las variables X y Y

En la Tabla 2.9 se indican los coeficientes calculados para cada conjunto de datos.

Tabla 2.9. Coeficientes de correlación.

Conjunto de datos	Coeficiente de Correlación		
	Media	Mínimo	Máximo
Mediana	-0,851	-0,774	-0,888
Promedio	-0,867	-0,779	-0,923
Moda	-0,812	-0,569	-0,900
Desviación estándar	0,352	-0,063	0,843

Con los coeficientes calculados es evidente que la distancia entre los dispositivos (*Beacons*-BLE central) tiene el mayor impacto en la variación de la potencia de las señales, además que el promedio de las mediciones presenta la mayor correlación con la distancia (-0,923); por otra parte, a pesar de la existencia de una correlación (0,352) entre la distancia y la desviación estándar de las mediciones, este valor denota una dependencia débil entre dichas variables. Con estos precedentes, se decidió optimizar el factor exponencial de pérdida de trayectoria n y el valor referencial $RSSI_0$ en el proceso de ajuste del modelo de propagación (Ecuación 2.1) y considerando como datos de entrada el promedio de las mediciones de potencia y la distancia.

Para poder contrastar los resultados del proceso de ajuste de curva se recurre a dos métodos diferentes existentes en Matlab: *lsqcurvefit* y *fminsearch*.

Lsqcurvefit resuelve problemas de optimización no lineales por mínimos cuadrados a través de dos algoritmos *trust region reflective* y *Levenberg-Marquardt*. Por su parte, la función *fminsearch* emplea el método simple de búsqueda directa *Nelder-Mead* (conocido también como método simple *downhill*). Ambos métodos buscan los valores más óptimos en función de los datos adquiridos experimentalmente. La Tabla 2.10 muestra los resultados del ajuste con los dos algoritmos.

Tabla 2.10. Valores de n y $RSSI_0$.

Beacon	Constante	Lsqcurvefit		fminsearch
		trust region	Levenberg	Nelder-Mead
EPN1	n	2,66	2,66	2,66
	$RSSI_0$	-50,32	-50,32	-50,32
EPN2	n	2,08	2,08	2,08
	$RSSI_0$	-51,29	-51,29	-51,29
EPN3	n	2,71	2,71	2,71
	$RSSI_0$	-48,64	-48,64	-48,64
EPN4	n	2,01	2,01	2,01
	$RSSI_0$	-50,17	-50,17	-50,17

EPN5	n	2,2	2,2	2,2
	<i>RSSI</i> ₀	-50,10	-50,10	-50,10

Con base en los resultados obtenidos, se definen los valores para las constantes del modelo de propagación (Ecuación 2.1) y estos son: $n = 2.37$ y $RSSI_0 = -50,10$. La Figura 2.13 presenta la comparación entre el modelo ajustado y los datos adquiridos. Con estos valores la Ecuación 2.1 da lugar a la Ecuación 2.2 y la distancia estimada en función del *RSSI* se presenta en la Figura 2.14.

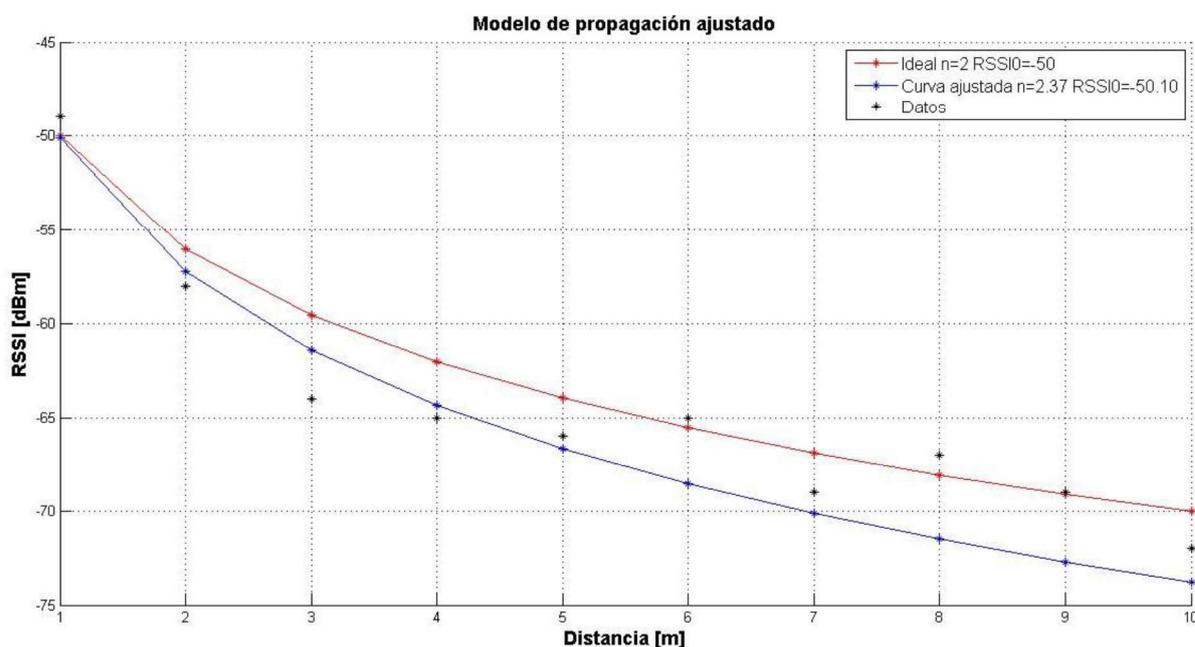


Figura 2.13. Modelo de propagación ajustado

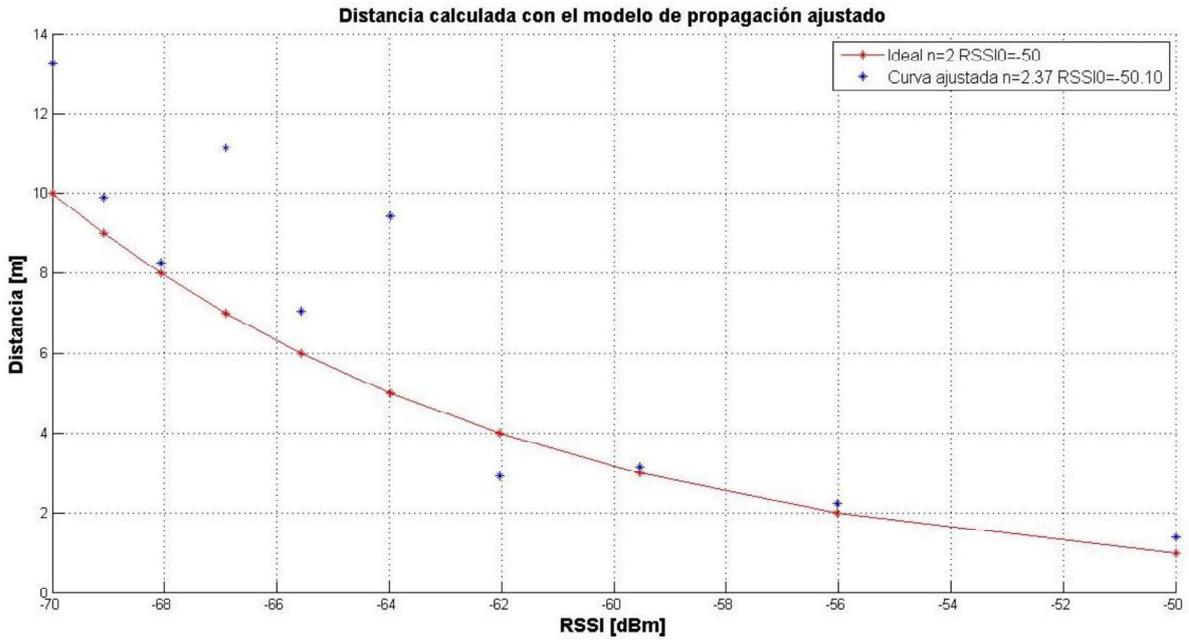


Figura 2.14. Distancia estimada en función del *RSSI*

El error en la estimación de la distancia se calcula como:

$$error(d) = |d_{real} - d_{estimada}| \quad (2.8)$$

Donde la distancia real es:

$$d_{real} = \sqrt{(x_r - x_p)^2 + (y_r - y_p)^2} \quad (2.9)$$

Donde

r : Posición del *Beacon*

p : Posición del BLE central

El cálculo del error presenta valores aceptables para el rango de 1 a 9 metros (Figura 2.15), generando valores entre 0 y 3 metros para el 82,5% de las mediciones (Tabla 2.11). Considerando que el dispositivo central no estará a distancias mayores a 9 metros de cada *Beacon*, el margen de error es manejable para el algoritmo; sin embargo, su presencia influye directamente en la estimación final de la ubicación. El incremento del error al calcular la distancia disminuye la precisión final con la cual se ubica el dispositivo móvil [28] [29].

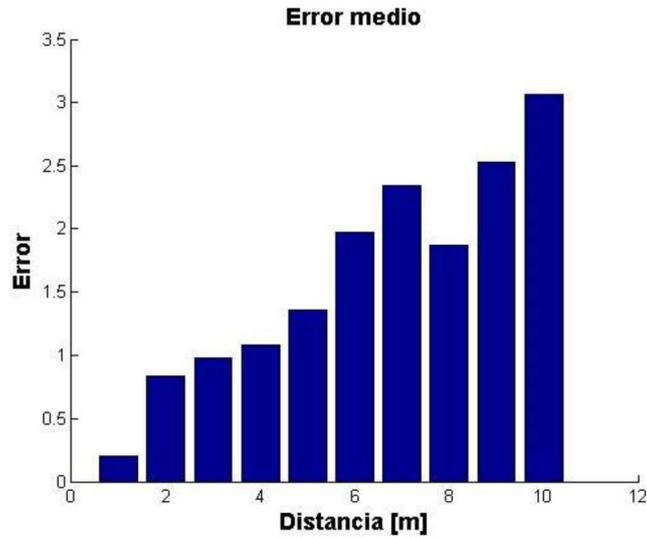


Figura 2.15. Error en la estimación de la distancia

Tabla 2.11. Porcentajes de error

Error	%
< 1 m	45
1m<error<3m	37,5
3m<	17,5
Total	100

La Tabla 2.12 presenta los resultados del cálculo del error al usar mediciones de potencia para estimar la distancia.

Tabla 2.12. Errores en la estimación de la distancia

	Distancia	1	2	3	4	5	6	7	8	9	10
Error	EPN1	0,04	1,81	0,97	0,60	2,51	0,36	1,55	0,26	0,57	6,00
	EPN2	0,33	0,04	1,04	1,62	0,74	0,82	4,27	2,82	0,58	1,27
	EPN3	0,26	0,38	1,70	0,26	0,70	6,42	1,30	3,30	4,69	1,27
	EPN4	0,19	1,10	0,18	1,84	1,49	0,29	2,28	1,07	4,30	3,71
	EPN5	0,17	0,56	0,63	0,94	1,05	3,2	1,45	2,01	0,43	4,21
Error medio		0,20	0,83	0,97	1,08	1,36	1,97	2,35	1,86	2,54	3,06

Se puede concluir que, utilizando el promedio de las mediciones para definir las constantes del modelo de propagación y estimar la distancia, el error obtenido para distancias entre 1 y 10 metros se encuentra dentro de un rango manejable para el algoritmo. Sin embargo, es preciso establecer un proceso de filtrado de las mediciones que permita reducir el error medio al calcular separación entre los dispositivos *LE*.

2.2.3 MEJORA EN LA ADQUISICIÓN DE DATOS

La variabilidad en la medición de potencia en un mismo punto a lo largo del tiempo (Figura 2.16) por efecto de las características del ambiente cerrado, hace que la estimación de la distancia presente errores; en consecuencia, es necesario un proceso de filtrado de las señales recibidas.

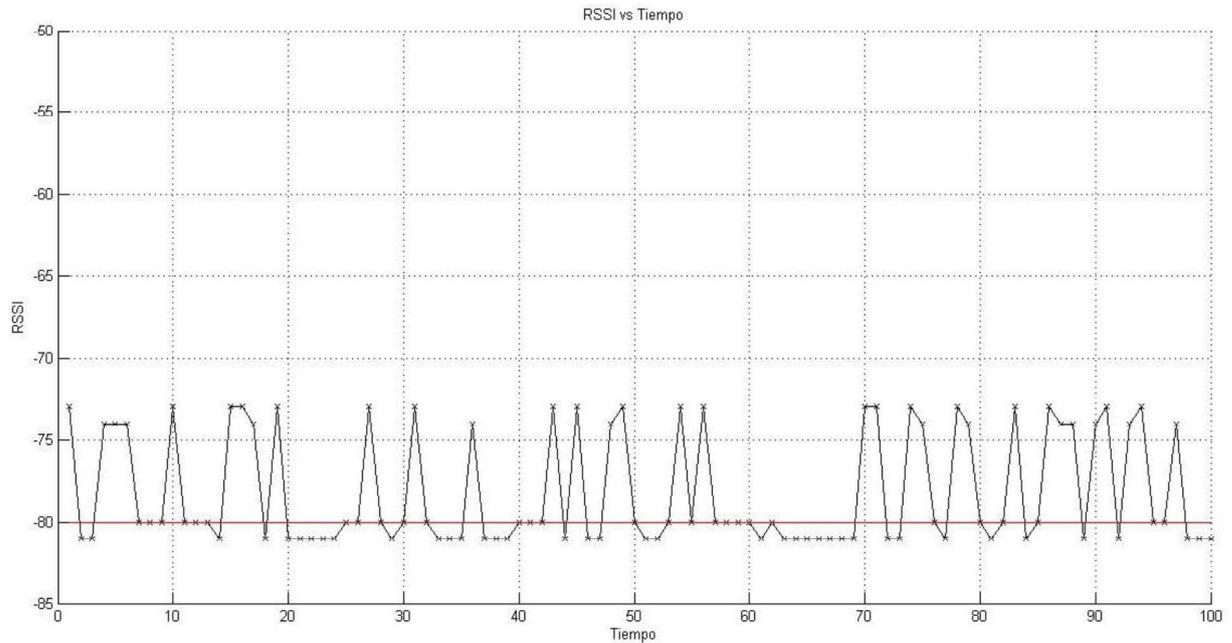


Figura 2.16. Variación de la potencia en el tiempo d=10m

Para el proceso de análisis se toman 5 grupos de 10 mediciones cada uno por cada distancia en el mismo espacio de adquisición de datos descrito anteriormente (Figura 2.11).

De estas muestras se analizan los siguientes casos:

1. Promedio directo.

$$RSSI = \frac{1}{n} \sum_{i=1}^n RSSI_i \quad (2.10)$$

2. Promedio eliminando el 10% de valores máximos y mínimos.

$$RSSI = \frac{1}{n-2} \sum_{i=1}^{n-2} RSSI_i \quad (2.11)$$

3. Promedio eliminando el 20% de valores máximos y mínimos.

$$RSSI = \frac{1}{n-2} \sum_{i=1}^{n-2} RSSI_i \quad (2.12)$$

4. Promedio de los datos luego de pasar por un filtro de media móvil.

$$RSSI = \frac{1}{n} \sum_{i=1}^n \left(\frac{1}{M} \sum_{k=0}^M RSSI_{(i-k)} \right) \quad (2.13)$$

Los procesos mencionados permiten obtener una única medición de *RSSI* a una distancia específica del *Beacon*. Los resultados del cálculo del error en la estimación de la distancia se presentan en Figura 2.17, Figura 2.18, Figura 2.19 y Figura 2.20.

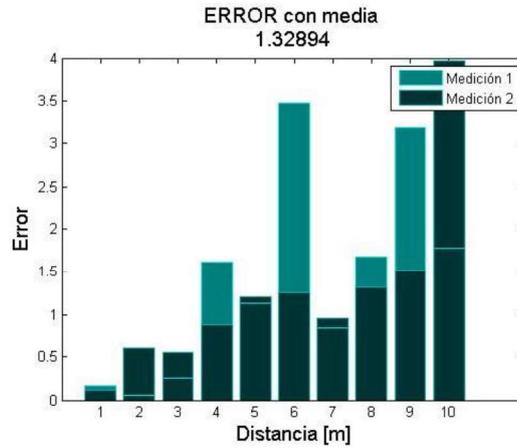


Figura 2.17. Error en la estimación de la distancia (Promedio Directo)

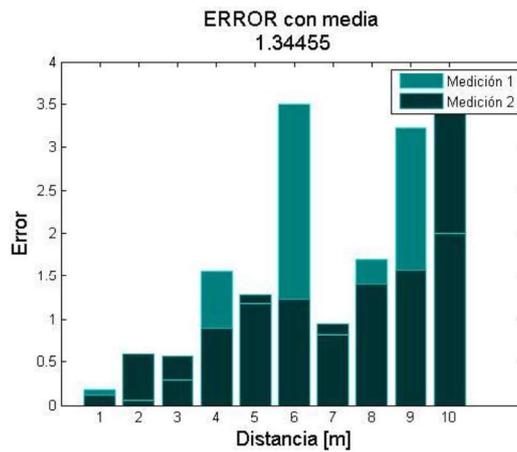


Figura 2.18. Error en la estimación de la distancia (Promedio eliminando el 10% de valores máximos y mínimos)

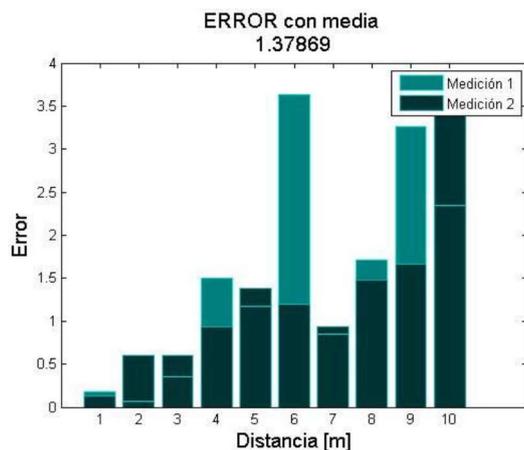


Figura 2.19. Error en la estimación de la distancia (Promedio eliminando el 20% de valores máximos y mínimos)

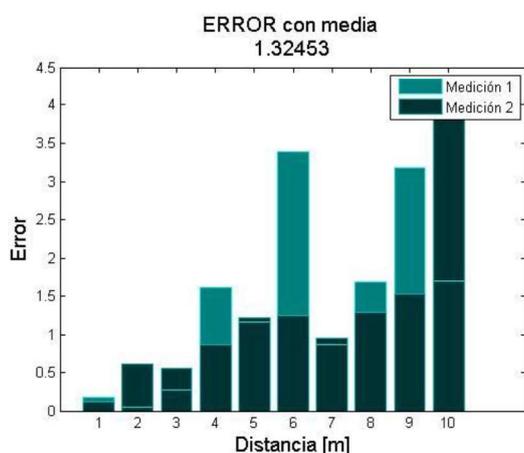


Figura 2.20. Error en la estimación de la distancia (Promedio de ventana móvil 3)

La Tabla 2.13 compara los valores de error medio obtenidos con cada proceso, por otro lado, la Tabla 2.14 resume en porcentaje el número de mediciones con errores menores a un metro, entre uno y tres metros, y mayores a tres metros, para cada método de procesamiento de datos.

Tabla 2.13. Error en la estimación de la distancia para los procesamientos de datos propuestos

Distancia [m]	Error medio para cada distancia									
	1	2	3	4	5	6	7	8	9	10
Promedio Directo	0,17	0,05	0,26	1,61	1,13	3,47	0,85	1,68	3,19	1,77
Promedio -10%	0,18	0,06	0,29	1,57	1,18	3,50	0,83	1,70	3,22	2,01
Promedio -20%	0,18	0,07	0,35	1,51	1,17	3,63	0,85	1,71	3,27	2,35
Promedio VM 3	0,17	0,05	0,27	1,61	1,16	3,39	0,87	1,68	3,18	1,69

Tabla 2.14. Porcentaje de error

Rango	Porcentaje de Error			
	Promedio Directo	Promedio - 10%	Promedio - 20%	Promedio MA 3
<1m	51,5	51,25	50,75	51,75
1m-3m	32,5	32	32,75	32,75
3m<	16	16,75	16,5	15,5
Error medio Total	1,329	1,345	1,379	1,325

Con los resultados obtenidos se evidencia que los cuatro procesos generan errores similares; sin embargo, el cuarto proceso denota valores menores de error frente a los otros, además su porcentaje de valores estimados de distancia menores a tres metros es mayor. Por otro lado, el proceso previo de filtrado ayuda a suavizar la variación de las mediciones, como se evidencia en la Figura 2.21. Con estos precedentes se decide utilizar el cuarto proceso para la adquisición de datos que usara el algoritmo.

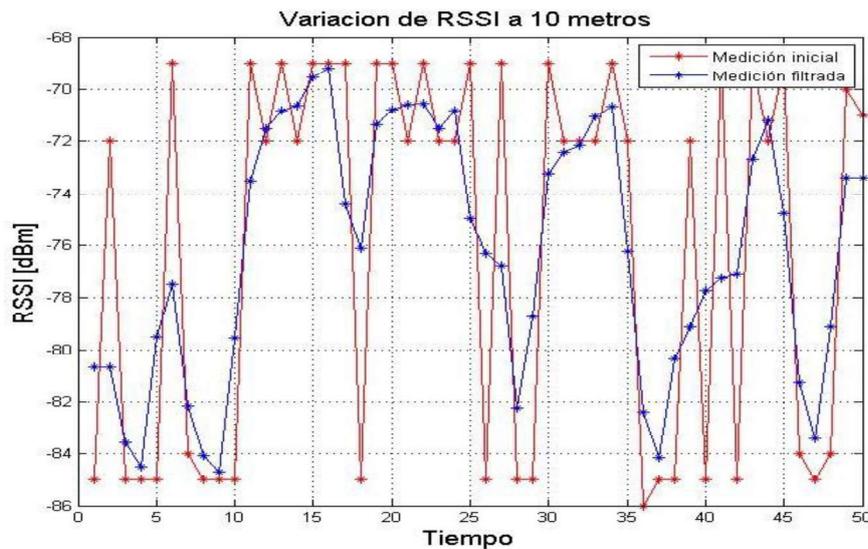


Figura 2.21. Variación de las mediciones con proceso de filtrado

Con base en los valores expuestos en la Tabla 2.13 y Tabla 2.14, y la Figura 2.17, Figura 2.18, Figura 2.19 y Figura 2.20, se aprecia que el error aumenta considerablemente al estimar ciertas distancias (6, 9 y 10 metros), lo cual hace que el error medio se incremente también. En consecuencia, es necesario analizar el comportamiento de las mediciones en esas distancias para determinar la manera de reducir el error.

Uno de los efectos que afecta en gran medida la adquisición de datos, son las múltiples trayectorias que se generan cuando las señales rebotan contra paredes, haciendo que muchas veces la señal que es recibida presente valores atípicos de potencia. Como se

puede observar en la Figura 2.21 existen variaciones demasiado pronunciadas donde el RSSI medido llega hasta -86 dBm; si se considera que en el modelo de propagación ajustado el valor de RSSI a 10 metros es de -73,77 dBm y que el valor promedio de las mediciones a 10 metros en el ambiente de prueba es -74,3 dBm, una medición por debajo de estos valores se puede considerar como atípica. La Tabla 2.15 presenta una comparación de los valores de RSSI esperados para cada distancia.

Tabla 2.15. Valores de RSSI esperados para cada distancia

Distancia	1	2	3	4	5	6	7	8	9	10
Modelo ajustado	-50,1	-57,2	-61,4	-64,4	-66,6	-68,5	-70,1	-71,5	-72,7	-73,8
Modelo ideal	-50,0	-56,0	-59,5	-62,0	-64,0	-65,6	-66,9	-68,1	-69,1	-70,0
Promedio ambiente de prueba	-51,0	-58,3	-62,4	-64,6	-65,1	-67,7	-70,3	-71,5	-72,4	-74,3

Si se toma en cuenta que la separación máxima entre el dispositivo central y un *Beacon* no supera los 11 metros y que a esta distancia el RSSI estimado es de -74,5 dBm, entonces cualquier medición por debajo de -75 dBm se considera un valor atípico generado por una de las múltiples trayectorias. Para evitar que estas mediciones afecten la estimación de la distancia, se deben descartar; la medición descartada se reemplaza por la anterior medición válida. De esta manera el proceso de adquisición de medidas se define en la Figura 2.22.

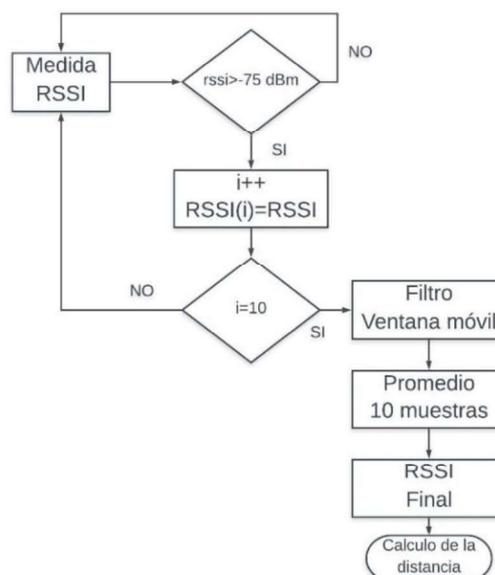


Figura 2.22. Proceso de adquisición de medidas

La Figura 2.23 y Tabla 2.16 presentan los errores obtenidos al aplicar el proceso de adquisición de medidas.

Tabla 2.16. Error al aplicar el proceso de adquisición de medidas

	Error medio									
Distancia	1	2	3	4	5	6	7	8	9	10
Medida 1	0,17	0,05	0,27	1,61	0,53	1,32	0,73	1,70	1,72	2,03
Medida 2	0,12	0,60	0,54	0,87	1,21	0,79	0,67	1,35	1,74	1,12
Rango	Porcentaje									
< 1 m	56,25									
1m-3m	33,5									
3m<	10,25									
Error medio	0,96									

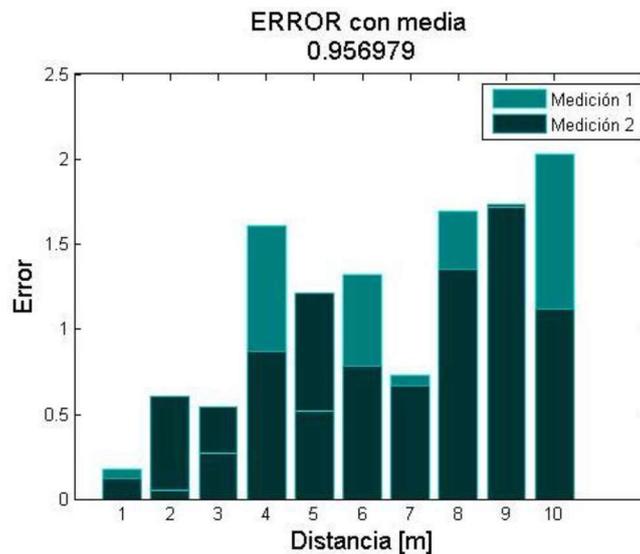


Figura 2.23. Error en la estimación de la distancia (Proceso de adquisición de medidas)

Los resultados obtenidos presentan una mejora considerable al disminuir el error medio en la estimación de la distancia de 1.32 a 0.96, además, el porcentaje de estimaciones con errores menores a 3 metros aumentó de 84.5% a 89.75%. En consecuencia, se puede concluir que este proceso es adecuado para el cálculo de la distancia.

2.3 IMPLEMENTACIÓN DE LA RED *BLUETOOTH LOW ENERGY*

La red que se implementa es una *Piconet* con un solo dispositivo BLE central (maestro) y un conjunto desde tres hasta máximo cinco *Beacons* (periférico) (Figura 2.24); para lo cual se configura un módulo HM-10 como central y cinco como *Beacon* (Tabla 2.17).

Tabla 2.17. Dispositivos de la red *Bluetooth LE*

	BLE central	Beacon 1	Beacon 2	Beacon 3	Beacon 4	Beacon 5
Nombre	CENTRAL	EPN1	EPN2	EPN3	EPN4	EPN5
Mayor	-	1	1	1	1	1
Minor	-	1	2	3	4	5

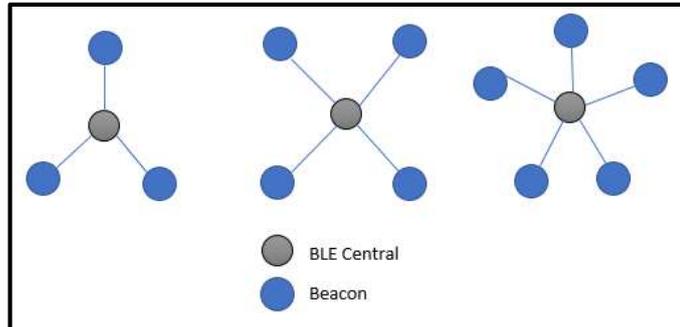


Figura 2.24. Piconets con 3, 4 y 5 Beacons

El dispositivo central es capaz de recibir los *advertisings* de cada *Beacon* y procesar las señales para obtener el *advertising PDU* del protocolo *iBeacon* (Figura 2.25). Los *Beacons* a su vez transmiten periódicamente los paquetes, todos con el mismo intervalo de tiempo y la misma potencia de transmisión. Es importante destacar que no se establece ningún tipo de conexión entre el dispositivo central y los *Beacons*.

rsi_1		readData	
1x3 cell			
1	2	3	
1 OK+DISIS	OK+DISC:4C000215:10F86430134611E491910800200C9A66:00010001C8:74F22E94933B:-042	OK+DISCE	

Figura 2.25. *Advertising PDU* recibido por el dispositivo central

La distribución física de los *Beacons* se define según cada prueba (Sección 2.5) dentro de un área de 10 x 5 metros y cada uno se coloca sobre un parante a 1.5 metros de altura desde el piso (Figura 2.26).

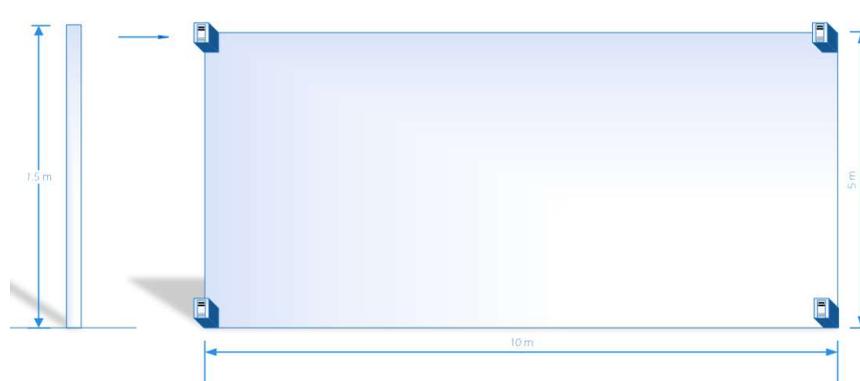


Figura 2.26. Distribución física de los *Beacons*

Las dimensiones del área se definen por el tamaño promedio de una oficina, local comercial, aulas de clase, entre otros ambientes interiores [42]. Por otro lado, la altura de los parantes donde se ubican los *Beacons* está condicionada por la altura promedio de una persona [29] con el fin de mantener línea de vista con el dispositivo móvil; además, la potencia de referencia ($RSSI_0$) se mide a una distancia de 1 metro [37], en consecuencia, se debe colocar los dispositivos a una altura mayor a este valor.

2.4 IMPLEMENTACIÓN DEL ALGORITMO DE TRILATERACIÓN

La trilateración es una técnica que permite estimar la posición de un objeto en función de medidas simultáneas de la distancia desde un mínimo de tres estaciones con ubicaciones conocidas [43]. Existen diversos métodos de trilateración: circular, esférica e hiperbólica; cada uno definido por la geometría que se utiliza para resolver el problema de ubicación. A continuación, se describen la trilateración esférica y circular, técnicas que se aplicaran en el desarrollo e implementación del algoritmo para la ubicación de objetos o personas en ambientes interiores.

2.4.1 TRILATERACIÓN ESFÉRICA

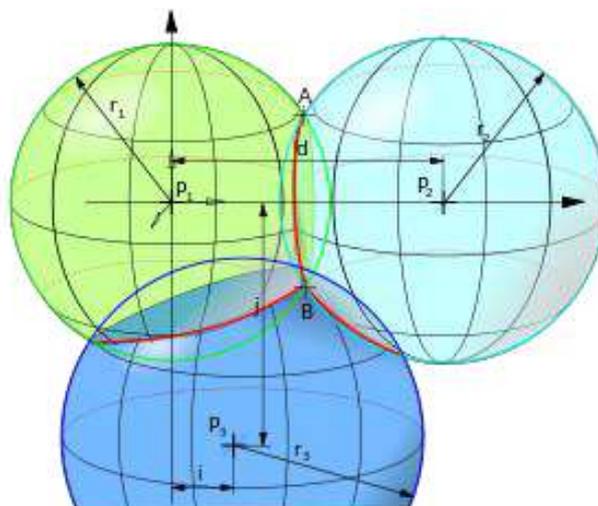


Figura 2.27. Trilateración esférica [22]

La trilateración esférica se puede expresar como el problema para encontrar la intersección de tres esferas (Figura 2.27), en otras palabras, encontrar la solución al siguiente sistema de ecuaciones cuadráticas [8] [43]:

$$\left. \begin{aligned} (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 &= l_1^2 \\ (x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 &= l_2^2 \\ (x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 &= l_3^2 \end{aligned} \right\} \quad (2.14)$$

Donde:

$p_i = (x_i, y_i, z_i) \quad i = 1, 2, 3$: Son las coordenadas de las estaciones conocidas.

l_i : Es la distancia entre la estación y el punto a ubicar.

La solución se puede hallar fácilmente al encontrar la intersección de una línea y una esfera. El anterior sistema de ecuaciones (Ecuación 2.14) se puede simplificar a un sistema de dos ecuaciones lineales, cuya solución es una línea, y una ecuación cuadrática [43] (Figura 2.28):

$$\left. \begin{aligned} (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 &= l_1^2 \\ x(x_2 - x_1) + y(y_2 - y_1) + z(z_2 - z_1) &= \frac{l_1^2 - l_2^2 - a^2}{2} \\ x(x_3 - x_1) + y(y_3 - y_1) + z(z_3 - z_1) &= \frac{l_1^2 - l_3^2 - b^2}{2} \end{aligned} \right\} \quad (2.15)$$

Donde:

$$a = \|P_2 - P_1\| \quad y \quad b = \|P_3 - P_1\|$$

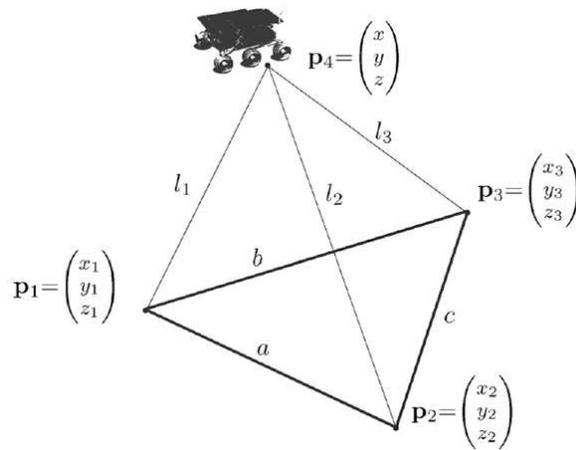


Figura 2.28. Sistema de trilateración esférica [43]

Una de las formas para obtener las dos ecuaciones lineales en la Ecuación 2.15 es restar en Ecuación 2.14, la segunda y la tercera de la primera, eliminando así los términos cuadráticos. Esta solución se puede simplificar más aun dependiendo de donde se ubiquen las referencias dentro de un sistema de coordenadas, sin embargo, si se cambia la referencia o el sistema de coordenadas las limitantes y complicaciones para encontrar una solución aumentan [8] [43]. Por estos motivos se prefiere usar soluciones que no estén ligadas a un sistema de coordenadas en particular.

Existen algunos métodos numéricos para hallar una solución al problema de la trilateración esférica, los cuales estiman la mejor posición de un objeto a través de la linealización de las ecuaciones de distancia. De esta manera el sistema puede ser formulado como un problema no lineal de mínimos cuadrados para identificar la mejor aproximación de la ubicación [43].

2.4.2 TRILATERACIÓN CIRCULAR

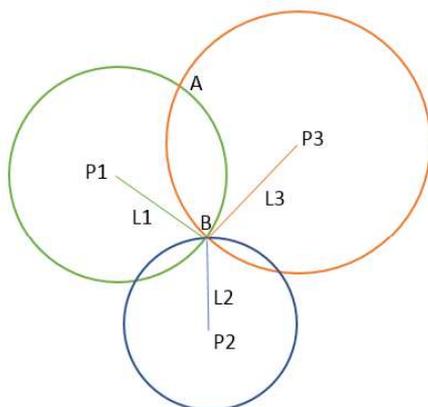


Figura 2.29. Trilateración circular

La trilateración circular es un método derivado de la trilateración esférica con ciertas simplificaciones. Se considera un sistema bidimensional, donde el plano z es una constante en el cual los puntos de referencia y el objetivo están ubicados. De aquí que, para encontrar la ubicación del objetivo, se busca la intersección de las tres circunferencias (en la forma más simplificada) o el centro del triángulo de menor área que se forma con dichas intersecciones [8]. La Ecuación 2.16 define el sistema de ecuaciones para encontrar la intersección de las tres circunferencias.

$$\left. \begin{aligned} (x - x_1)^2 + (y - y_1)^2 &= l_1^2 \\ (x - x_2)^2 + (y - y_2)^2 &= l_2^2 \\ (x - x_3)^2 + (y - y_3)^2 &= l_3^2 \end{aligned} \right\} \quad (2.16)$$

Donde:

$p_i = (x_i, y_i) \quad i = 1, 2, 3$: Son las coordenadas de las estaciones conocidas.

l_i : Es la distancia entre la estación y el punto a ubicar.

La intersección de las dos primeras circunferencias genera dos posibles soluciones A y B, es ahí donde la intersección con la tercera circunferencia determina donde se encuentra ubicado el objeto (B) (Figura 2.29).

2.4.3 MÉTODO DE GAUSS NEWTON APLICADO A LA TRILATERACIÓN ESFÉRICA

El algoritmo de Gauss-Newton fue desarrollado por Carl Friedrich Gauss al modificar el método de optimización de Newton. Es un método iterativo para resolver problemas no lineales de mínimos cuadrados. Este al ser aplicado a trilateración permite encontrar la

posición de un objeto móvil mediante un proceso que va llevando la ubicación del móvil desde una posición inicial P_0 , que normalmente es asumida, hacia la posición real del mismo [8] (Figura 2.30).

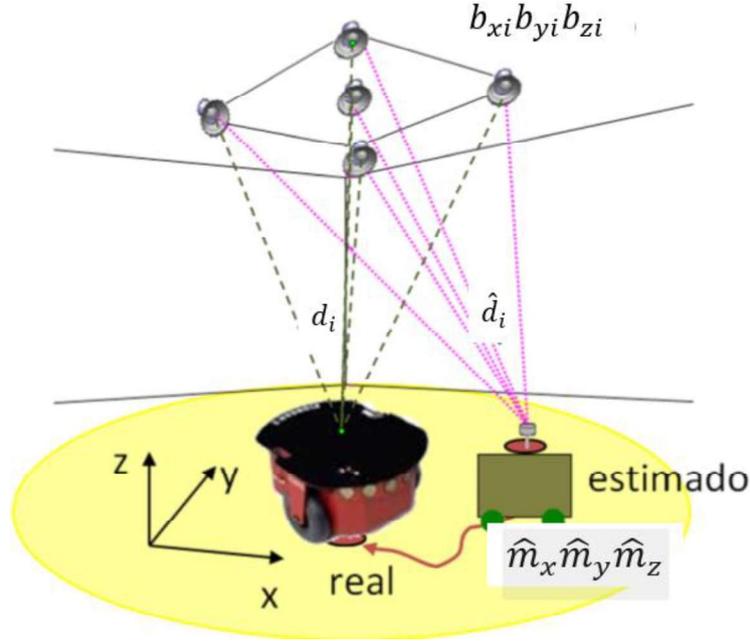


Figura 2.30. Estimación de la posición basado en el método de Gauss-Newton [8]

En un sistema de ubicación en interiores compuesto por B *Beacons*, la posición de cada *Beacon* está definida como $b_i = [b_{xi} b_{yi} b_{zi}]$ y la posición del objeto móvil $m = [m_x m_y m_z]$, entonces la distancia entre ellos se define como:

$$d_i = \sqrt{(b_{xi} - m_x)^2 + (b_{yi} - m_y)^2 + (b_{zi} - m_z)^2} \quad (2.17)$$

Además, considerando que el punto estimado se define como $\hat{m} = [\hat{m}_x \hat{m}_y \hat{m}_z]$, la distancia entre cada *Beacon* y la posición estimada es:

$$\hat{d}_i = \sqrt{(b_{xi} - \hat{m}_x)^2 + (b_{yi} - \hat{m}_y)^2 + (b_{zi} - \hat{m}_z)^2} \quad (2.18)$$

Este método trata de minimizar la suma de los errores cuadráticos de la distancia (Ecuación 2.19), por lo que la función de error de la distancia se define como:

$$f_i(m_x m_y m_z) = \hat{d}_i - d_i = \sqrt{(b_{xi} - \hat{m}_x)^2 + (b_{yi} - \hat{m}_y)^2 + (b_{zi} - \hat{m}_z)^2} - d_i \quad (2.19)$$

$$F(m_x m_y m_z) = \sum_{i=1}^B (\hat{d}_i - d_i)^2 = \sum_{i=1}^B (f_i(m_x m_y m_z))^2 \quad (2.20)$$

El algoritmo define a la Ecuación 2.21 como la relación que permite encontrar cada nueva estimación de la posición:

$$\hat{m}_{k+1} = \hat{m}_k - \Delta X_k \quad (2.21)$$

Donde:

\hat{m}_k : Posición estimada previa.

\hat{m}_{k+1} : Nueva posición estimada.

ΔX_k : Variación en la posición.

La variación de la posición se calcula como:

$$\Delta X = (A^T A)^{-1} A^T B \quad (2.22)$$

Donde:

ΔX : Variación de la posición (Ecuación 2.22).

A : Jacobiano de las funciones de error (f_i) (Ecuación 2.24).

B : Matriz de las funciones de error (f_i) (Ecuación 2.26).

$$\Delta X = \begin{bmatrix} \Delta \hat{m}_x \\ \Delta \hat{m}_y \\ \Delta \hat{m}_z \end{bmatrix} \quad (2.23)$$

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial \hat{m}_x} & \frac{\partial f_1}{\partial \hat{m}_y} & \frac{\partial f_1}{\partial \hat{m}_z} \\ \frac{\partial f_2}{\partial \hat{m}_x} & \frac{\partial f_2}{\partial \hat{m}_y} & \frac{\partial f_2}{\partial \hat{m}_z} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_B}{\partial \hat{m}_x} & \frac{\partial f_B}{\partial \hat{m}_y} & \frac{\partial f_B}{\partial \hat{m}_z} \end{bmatrix} \quad (2.24)$$

Donde las derivadas parciales se calculan como:

$$\frac{\partial f_i}{\partial \hat{m}_x} = \frac{(\hat{m}_x - b_{xi})}{\hat{d}_i} ; \frac{\partial f_i}{\partial \hat{m}_y} = \frac{(\hat{m}_y - b_{yi})}{\hat{d}_i} ; \frac{\partial f_i}{\partial \hat{m}_z} = \frac{(\hat{m}_z - b_{zi})}{\hat{d}_i} ; i = 1 \dots B \quad (2.25)$$

$$B = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_B \end{bmatrix} \quad (2.26)$$

Cada nueva posición del móvil (Ecuación 2.21) se calcula una vez se hayan obtenido las nuevas variaciones (Ecuación 2.23). Este proceso se repite k veces hasta que las variaciones ΔX sean inferiores a un umbral definido al establecer el algoritmo. Al ser un

método iterativo, la rapidez con la que la función converge y se encuentra la mejor aproximación a la solución, depende de la posición inicial (P_0) que se asume para iniciar el algoritmo.

2.4.4 IMPLEMENTACIÓN DEL ALGORITMO

En esta fase, utilizando un modelo de programación determinístico se implementa el algoritmo conforme a las características de la red de *Beacons* implantada. Además, se selecciona como herramienta de desarrollo Matlab ya que facilita el análisis de grandes volúmenes de datos, su método de programación se ajusta a los requerimientos y planteamientos del proyecto, no es necesario el uso de hardware adicional al previamente mencionado, y permite un desarrollo rápido del algoritmo.

El algoritmo contempla los siguientes procesos: inicialización de la comunicación serial, búsqueda de *Beacons* activos, procesamiento de las señales, algoritmo de trilateración básica, algoritmo de trilateración basado en el método de Gauss Newton, presentación de resultados. La Figura 2.31 muestra la lógica de programación del algoritmo.

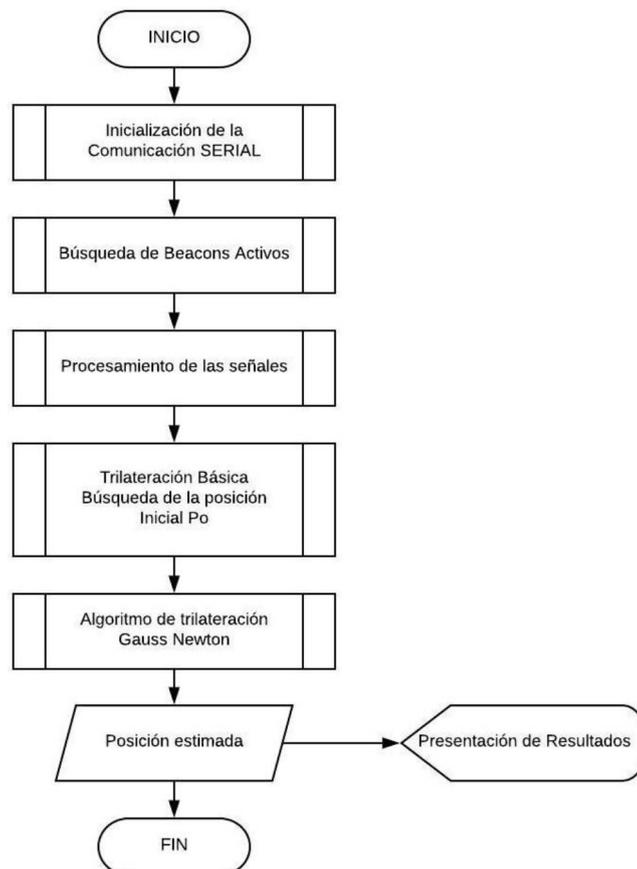


Figura 2.31. Diagrama de flujo del algoritmo de trilateración

A continuación, se analiza y detalla cada proceso que forma parte del algoritmo.

2.4.4.1 Inicialización de la comunicación serial

La lógica de programación de este proceso se ilustra en la Figura 2.32.

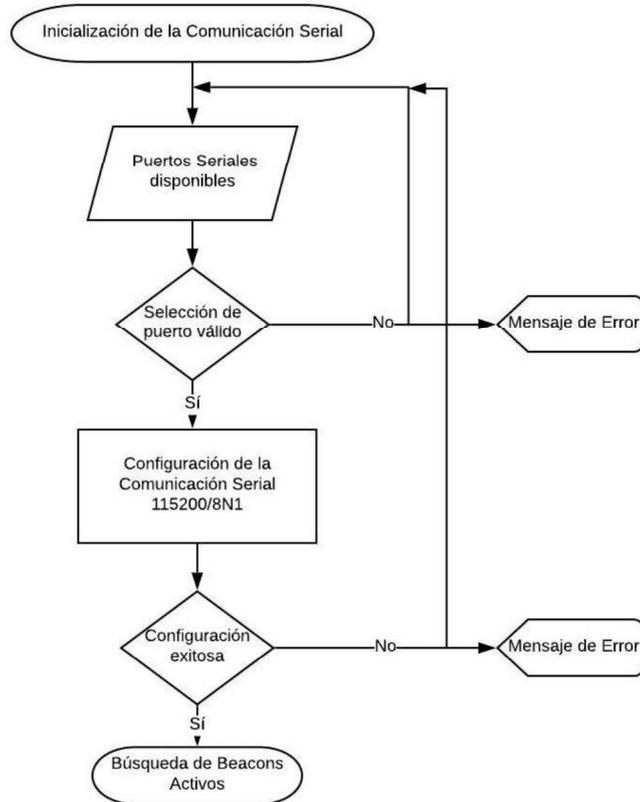


Figura 2.32. Diagrama de flujo del proceso de Inicialización de la comunicación serial

La información que proviene de la red inalámbrica es recogida por el dispositivo BLE central, el cual a su vez transmite esta información al algoritmo a través de la interfaz serial. Esta se configura en base al puerto disponible (Figura 2.33) y los parámetros definidos en la Sección 2.1.



```
serialPorts = instrhwinfo('serial');
```

Figura 2.33. Código para buscar puertos disponibles

El comando `instrhwinfo('serial')` devuelve la información sobre los puertos seriales disponibles. Si se selecciona un puerto invalido se despliega un mensaje de error, indicando que se debe seleccionar un puerto valido (Figura 2.34).



Figura 2.34. Mensaje de error por puerto invalido

Si se selecciona un puerto valido, esta información se usa para configurar los parámetros de la comunicación serial (Figura 2.35).

```
serPort = contents(get(hObject,'Value'));
serConn = serial(serPort,'BaudRate',115200,...
'DataBits',8,'StopBits',1,'Terminator','CR/LF');
```

Diagram annotations: A red box labeled "Puerto" points to the `serPort` variable. A blue box labeled "Parámetros" points to the list of serial parameters: `'BaudRate',115200,... 'DataBits',8,'StopBits',1,'Terminator','CR/LF'`.

Figura 2.35. Parámetros de configuración de la interfaz serial

El comando `contents = cellstr(get(hObject,'String'))` devuelve el valor del puerto seleccionado, por otro lado, el comando `Serial (Parámetros)` permite establecer los valores con los que se establece la comunicación. Una vez definidos el puerto y los parámetros se intenta abrir el puerto e iniciar la comunicación; si el proceso es exitoso el programa está listo para empezar al ordenarle al dispositivo BLE central que inicie la búsqueda de *Beacons*, caso contrario se despliega un mensaje de error (Figura 2.36).



Figura 2.36. Mensaje de error por falla al abrir el puerto y establecer la comunicación serial

2.4.4.2 Búsqueda de *Beacons* activos

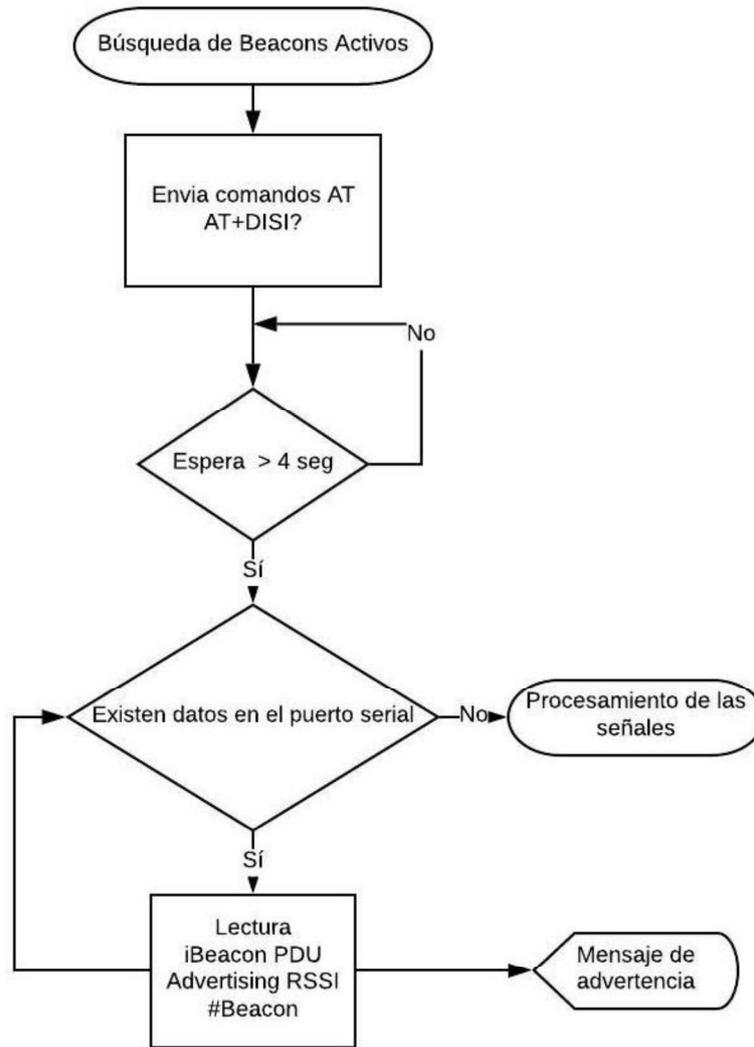


Figura 2.37. Diagrama de flujo del proceso de búsqueda de *Beacons* activos

Considerando que la inicialización de la comunicación serial fue exitosa, se inicia un proceso de búsqueda de los *Beacons* pertenecientes al sistema que se encuentren activos (Figura 2.38). La lógica de programación de este proceso se ilustra en la Figura 2.37.

```
fprintf(serConn, '%s\n', 'AT+DISI?');  
tic;  
while toc<=4  
    disp('Wait for incoming data...');  
end
```

Comando AT

Bucle de espera

Figura 2.38. Comando AT para descubrir *Beacons*

El comando `fprintf(serConn,'%s\n','AT+DISI?')` envía desde la interfaz serial al módulo central el comando AT para inicializar la búsqueda de los módulos BLE *Beacon*, entonces

se inicializa un bucle de espera (tic; toc<=4) de cuatro segundos (tiempo mínimo de ejecución de la búsqueda [39]) para recibir la información encontrada durante el escaneo. La Figura 2.39 muestra el proceso de adquisición de los datos escaneados. Mientras existan datos que leer en el puerto, estos se van almacenando en la variable tipo cell denominada readData, cuyo contenido se muestra en la Figura 2.40.

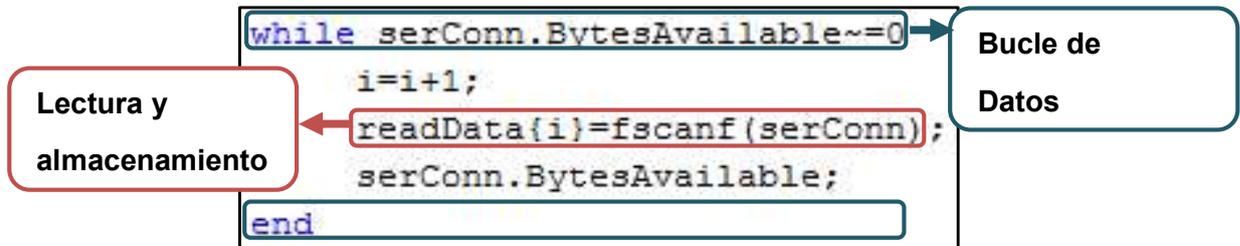


Figura 2.39. Lectura de datos enviados desde el módulo central

La imagen muestra una ventana de MATLAB con la variable `readData` seleccionada. El contenido de la variable es una matriz de celdas de 1x3. Una etiqueta "Inicio y fin de búsqueda" apunta a los índices 1 y 3 de la matriz. El contenido de la celda en el índice 2 es un UUID.

1	2	3
OK+DISIS	UUID	OK+DISCE

Figura 2.40. Variable readData.

En el caso donde no se encuentre ningún *Beacon* activo, se despliega un mensaje de advertencia como se presenta en la Figura 2.41.



Figura 2.41. Mensaje de advertencia de Beacons inactivos

2.4.4.3 Procesamiento de las señales

Una vez obtenida la información por parte de la red inalámbrica, esta debe ser procesada para definir la información que usará el algoritmo de trilateración. El flujo del procesamiento que se da a las señales se presenta en la Figura 2.42.

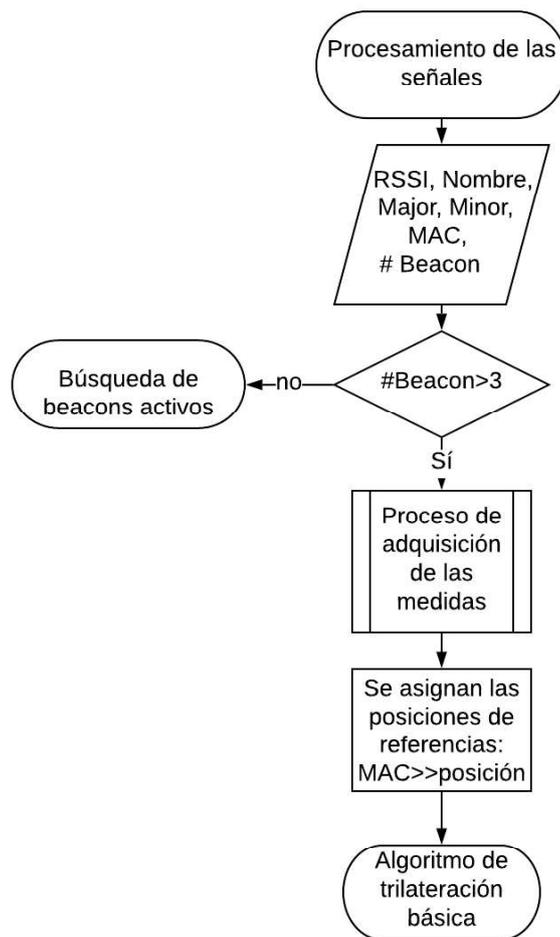


Figura 2.42. Diagrama de flujo del proceso de procesamiento de señales

En esta etapa es importante comprobar que se tiene al menos tres *Beacons* activos (requerimiento mínimo del algoritmo), si se cumple con esta condición se procesan las mediciones de RSSI de acuerdo a lo definido en la Sección 2.2, proceso de adquisición de medidas (Figura 2.22). Además, se asigna una posición de referencia en el plano a cada *Beacon* en base a su dirección MAC (*Media Access Control*). La Figura 2.43 y 2.44 muestra el código implementado para esta etapa.

```

N_beacons_inicial=N_beacons;
mediciones=zeros(10,5);indicador=0;
for n_medicion=1:10
    while N_beacons<3 || indicador==0
        beacons=[0 0 0 0 0];
        fprintf(serConn,'%s\n','AT+DISI?'); %write data
        tic;
        while toc<=4
            end
            i=0;          readData={};
            while serConn.BytesAvailable~=0
                i=i+1;
                readData{i}=fscanf(serConn);
                serConn.BytesAvailable;
                if length(readData{i})>32
                    beacon_id=readData{i}(1,62:73);
                    switch beacon_id
                        case '44EAD82A8ADE' beacons(1)=1;
                        case '44EAD82AACC8' beacons(2)=1;
                        case 'D436395E19E6' beacons(3)=1;
                        case '44EAD82A69C5' beacons(4)=1;
                        case '44EAD82A93C5' beacons(5)=1;
                    end
                end
            end
            N_beacons=sum(beacons);
            if N_beacons<=2
                disp('Adquiriendo datos... Beacons insuficientes');
            elseif N_beacons==N_beacons_inicial
                indicador=1; disp('Adquiriendo datos...');
            end
            if N_beacons>=3
                disp('Adquiriendo datos...');
            end
        end
        for dato=1:i
            if length(readData{dato})>32
                beacon_id=readData{dato}(1,62:73);
                switch beacon_id
                    case '44EAD82A8ADE'
mediciones(n_medicion,1)=str2num(readData{dato}(1,75:78));beacons(1)=1;
                    case '44EAD82AACC8'
mediciones(n_medicion,2)=str2num(readData{dato}(1,75:78));beacons(2)=1;
                    case 'D436395E19E6'
mediciones(n_medicion,3)=str2num(readData{dato}(1,75:78));beacons(3)=1;
                    case '44EAD82A69C5'
mediciones(n_medicion,4)=str2num(readData{dato}(1,75:78));beacons(4)=1;
                    case '44EAD82A93C5'
mediciones(n_medicion,5)=str2num(readData{dato}(1,75:78));beacons(5)=1;
                end
            end
        end
        indicador=0;
    end
end

```

Figura 2.43. Código implementado para el procesamiento de las señales

2.4.4.4 Algoritmo de trilateración básica

Como se mencionó anteriormente, una de las características más importantes que se debe definir en el algoritmo de trilateración esférica con el método de Gauss Newton, es el valor P_0 (posición inicial) que se asume para la primera iteración del algoritmo. Bajo esta premisa y considerando las ventajas de establecer una posición inicial adecuada, el objetivo de este algoritmo de trilateración básica es encontrar dicha posición (en lugar de asumirla a priori) para posteriormente usarla en el arranque del método de trilateración que estimara la ubicación final del objeto. Cabe resaltar que este algoritmo se fundamenta en el método de trilateración circular en su forma más básica. El proceso que sigue el algoritmo se detalla en la Figura 2.44.

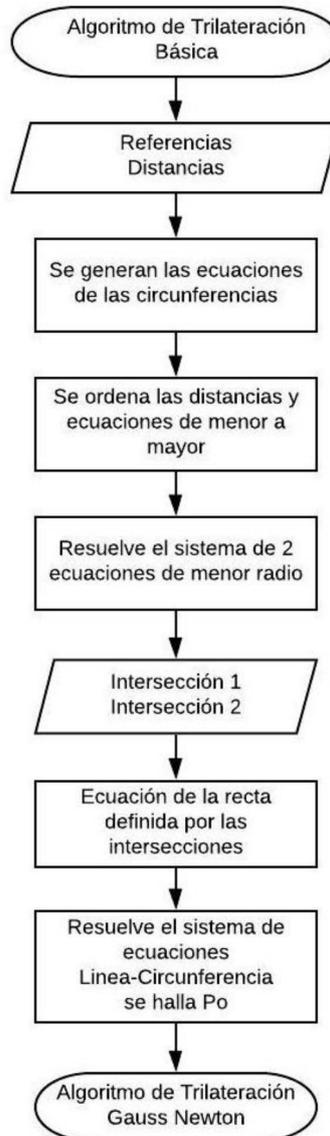


Figura 2.44. Diagrama de flujo del algoritmo de trilateración básica

Dentro de este proceso inicialmente se ordenan las distancias de mayor a menor, a continuación, se generan las ecuaciones de las dos circunferencias de menor radio (Ecuación 2.16) con centro en las referencias y radio las distancias medidas desde cada *Beacon* al dispositivo central (Figura 2.45).

```
ecuacion_1=(x-datos_trilateracion(1,2))^2+...  
            (y-datos_trilateracion(1,3))^2==radio_2(1);  
ecuacion_2=(x-datos_trilateracion(2,2))^2+...  
            (y-datos_trilateracion(2,3))^2==radio_2(2);
```

Ecuación 2.16

Figura 2.45. Ecuaciones de las circunferencias

A continuación, se hallan los puntos donde estas se intersecan (Figura 2.46).

```
cruce=solve(ecuacion_1,ecuacion_2);  
cruce_x=double(cruce.x);           %Valores en x donde intersecan las  
circunferencias  
cruce_y=double(cruce.y);           %Valores en y donde intersecan las  
circunferencias
```

Figura 2.46. Distancias ordenadas y cálculo de la intersección de las dos circunferencias de menor radio

Con los dos puntos donde se intersecan las dos circunferencias, se genera la ecuación de una recta, tras lo cual se busca el punto de cruce entre la recta y la tercera circunferencia de menor radio dentro del área de prueba. En consecuencia, este punto se convierte en la primera estimación de la posición del dispositivo central, es decir P_0 . Este proceso se presenta en la Figura 2.47.

```

if isreal(cruce_x) && isreal(cruce_y)
    %Ecuacion de la recta
    if (cruce_y(2,1)-cruce_y(1,1))==0
        recta=y==cruce_y(2,1);
    elseif (cruce_x(2,1)-cruce_x(1,1))==0
        recta=x==cruce_x(2,1);
    else
        recta=(x-cruce_x(1,1))/(cruce_x(2,1)-cruce_x(1,1))==...
            (y-cruce_y(1,1))/(cruce_y(2,1)-cruce_y(1,1));
    end
    %Ecuacion tercera circunferencia
    ecuacion_3=(x-datos_trilateracion(3,2))^2+...
        (y-datos_trilateracion(3,3))^2==radio_2(3);
    S=solve(recta,ecuacion_3);
    ble_x=double(S.x);
    ble_y=double(S.y);
    if isreal(ble_x) && isreal(ble_y)
        d_1=sqrt((ble_x(1,1)-cruce_x(1,1))^2+((ble_y(1,1)-
cruce_y(1,1))^2));
        d_2=sqrt((ble_x(2,1)-cruce_x(1,1))^2+((ble_y(2,1)-
cruce_y(1,1))^2));
        if d_1<d_2
            p_0=[ble_x(1,1) ble_y(1,1) -1];
        else
            p_0=[ble_x(2,1) ble_y(2,1) -1];
        end
    else
        if cruce_x(1,1)>=0 && cruce_y(1,1)>=0
            p_0=[cruce_x(1,1) cruce_y(1,1) -1];
        elseif cruce_x(2,1)>=0 && cruce_y(2,1)>=0
            p_0=[cruce_x(2,1) cruce_y(2,1) -1];
        end
    end
end

```

Figura 2.47. Proceso para el cálculo de P_0

2.4.4.5 Algoritmo de trilateración basado en el método de Gauss Newton

La lógica de programación que sigue el algoritmo se muestra en la Figura 2.48.

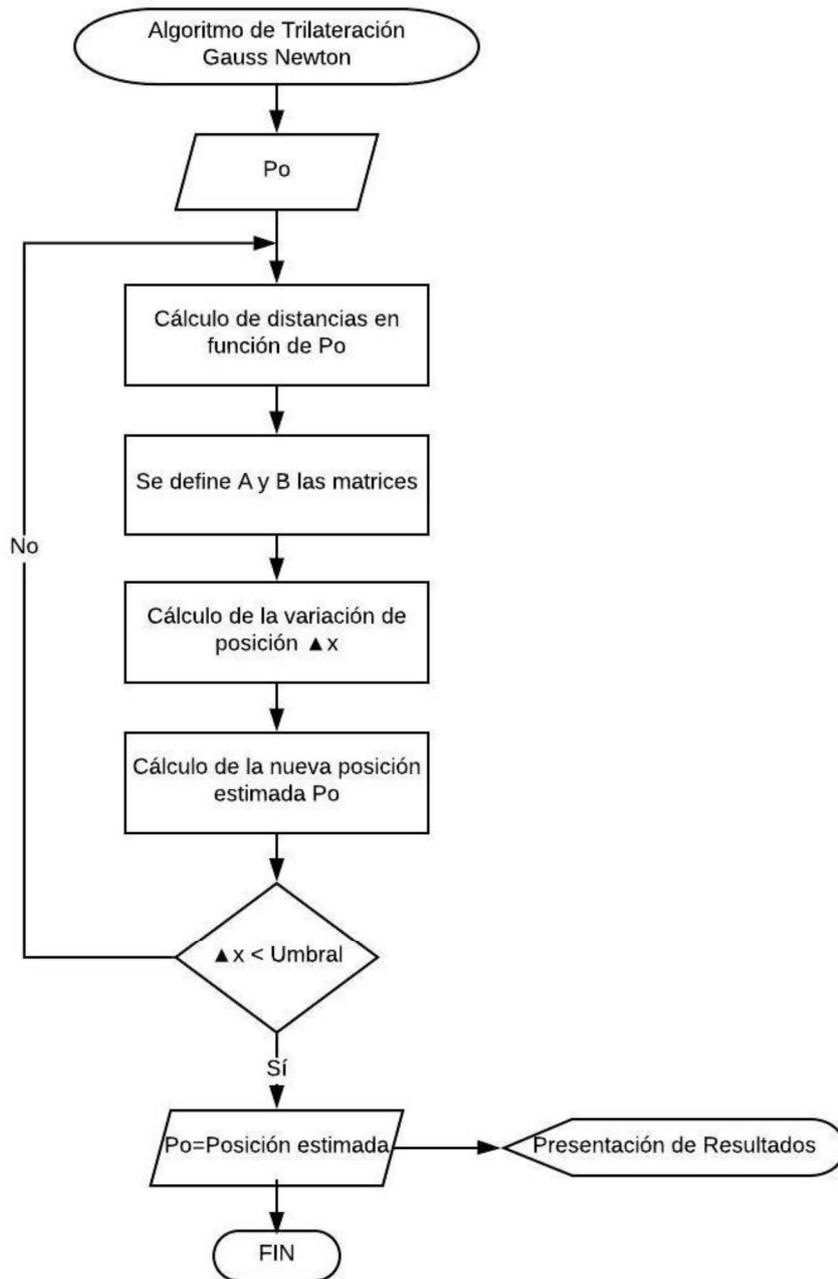


Figura 2.48. Diagrama de flujo del proceso del Algoritmo de trilateración basado en el método de Gauss Newton

Este algoritmo se define como una función de Matlab denominada `gaussnewton` y cuyos argumentos de entrada son: la ubicación de cada *Beacon*, las distancias, la posición inicial y el número de *Beacons* activos, y su salida es la posición final estimada del dispositivo central en dos dimensiones. La invocación de esta función se muestra en la Figura 2.49.

```

posicion_final=gaussnewton(referencias,distancias,p_0,N_beacons);
  
```

Figura 2.49. Función de trilateración `gaussnewton`.

Los pasos que sigue esta función se describen a continuación y se presentan en la Figura 2.50:

1. Para la primera iteración se calcula las distancias entre la posición inicial P_0 y la ubicación de los *Beacons*
2. Se definen las matrices A (Ecuación 2.24) y B (Ecuación 2.26)
3. Se calcula la matriz de variación de la posición ΔX (Ecuación 2.23)
4. Se calcula la nueva posición estimada (Ecuación 2.21).
5. Si la variación de la posición es menor al umbral ($\Delta X < 0.2 * 10^{-15}$) el proceso finaliza y la posición estimada en el paso 4 se transforma en la ubicación del dispositivo central. Por otro lado, si la variación de la posición es mayor al umbral ($\Delta X > 0.2 * 10^{-15}$) el proceso continúa, la posición estimada en el paso 4 se convierte en la nueva posición inicial (P_0) y se inicia una nueva iteración (Paso 1).

El umbral de ΔX es un valor que permite definir hasta donde va a ejecutarse el algoritmo, si es demasiado grande el número de iteraciones será menor, sin embargo, la precisión en la ubicación será pobre; por otro lado, si el umbral es muy pequeño el algoritmo ejecutara demasiadas iteraciones tratando de alcanzar el umbral, a pesar de que la solución más optima ya haya sido encontrada. Experimentalmente se definió que un umbral óptimo para las condiciones en las que se devuelve el algoritmo es de $\Delta X < 0.2 * 10^{-15}$ o un máximo de 100 iteraciones. Superando estos límites el algoritmo no presenta mejoras considerables en la precisión.

```

function [BLE_GN]=gaussnewton(referencias,distancias,BLE_i,N)

for interacion=1:100                                % # de
interaciones max
    for i=1:N
        1      di(i)=sqrt(((referencias(i,1)-BLE_i(1))^2)+... %Distancia
estimada
                ((referencias(i,2)-BLE_i(2))^2)+...
                ((referencias(i,3)-BLE_i(3))^2));
        2      A(i,1)=(BLE_i(1)-referencias(i,1))/di(i);      %Matriz de
derivadas parciales
                A(i,2)=(BLE_i(2)-referencias(i,2))/di(i);
                A(i,3)=(BLE_i(3)-referencias(i,3))/di(i);
        3      B(i,1)=di(i)-distancias(i);                    %Matriz de
error
    end
        4      AT=A';                                          % A' = A^T
                ATA=AT*A;                                     % A' * A
                ATA_1=ATA^-1;                                % (A' * A)^-1
                ATB=AT*B;                                     % A'*B
                %Calculo de la variacion de posicion en x, y, z
                DX=ATA_1*ATB;                                % (A' * A)^-1 * A'*B = Delta(x,y,z)
                %Nueva posicion estimada
                BLE_i=BLE_i-DX';                             % (X Y Z)n+1=(Xn Yn Zn)+Delta(X,Y,Z)
    end
        5      if abs(DX)<0.2*1.0e-15 % Tolerancia de variacion en la posicion
                break
            end
    end
BLE_GN=BLE_i;      SOLUCIÓN      %Ubicacion estimada del movil

```

Figura 2.50. Función gaussnewton

2.4.4.6 Presentación de resultados

Los resultados del algoritmo de ubicación se muestran dentro de un plano en dos dimensiones que hace referencia al espacio donde se realizan las mediciones (Figura 2.51).

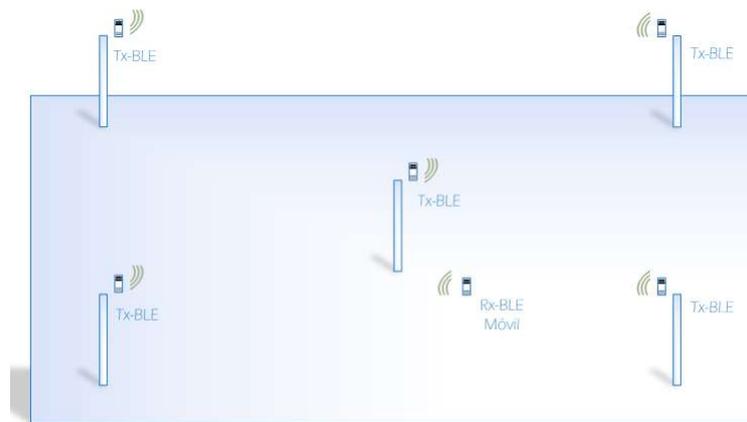


Figura 2.51. Espacio de mediciones

La Figura 2.52 muestra los resultados del proceso de ubicación del dispositivo BLE central. En esta se detallan:

- La ubicación de cada *Beacon* activo, en color azul y con identificador.
- Las circunferencias de menor radio de color rojo, amarillo y verde, en función de la distancia de menor a mayor respectivamente.
- La recta que cruza por las dos intersecciones entre las circunferencias de menor radio, en color negro.
- La posición inicial P_0 calculada con el algoritmo de trilateración básico, en color verde.
- La ubicación final estimada del dispositivo móvil, en color rojo.
- La ubicación real (pruebas) del dispositivo móvil, en color negro.

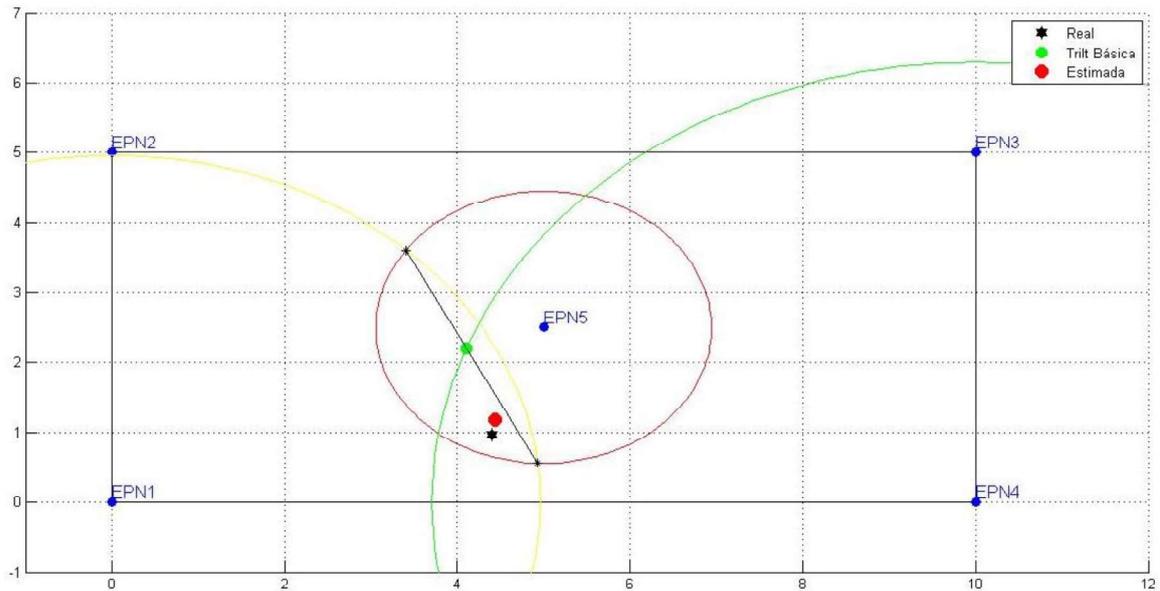


Figura 2.52. Resultados del proceso de ubicación

El segmento de código que genera la imagen de resultados se presenta en la Figura 2.53 y Figura 2.54.

```

%Area de prueba
rectangle('Position',[0 0 10 5],'LineWidth',1);
%Ubicación de los beacons referenciales
labels = {'EPN1','EPN2','EPN3','EPN4','EPN5'};
referencias=[0 0 1.5;0 5 1.5;10 5 1.5;10 0 1.5;5 2.5 1.5];
colores=['r' 'y' 'g' 'b' 'c'];
for i=1:5
    plot(referencias(i,1),referencias(i,2),...
        'Marker','o',...
        'MarkerFaceColor','b',...
        'MarkerEdgeColor','b');
    text(referencias(i,1),referencias(i,2),labels(i),...
        'FontSize',12,'Color','b',...
        'VerticalAlignment','bottom');
End
%Ubicación de los circulos con distancias mas cortas al dispositivo
central
for i=1:3
    viscircles([datos_trilateracion(i,2) datos_trilateracion(i,3)],...
        datos_trilateracion(i,1),...
        'LineStyle','-','...
        'EdgeColor',colores(i),...
        'LineWidth',0.25);
end
end

```

Figura 2.53 Código de presentación de resultados 1

```

%Puntos de cruce entre las circunferencias mas cercanas
plot(real(cruce_x),real(cruce_y),...
      'black','Marker','*',...
      'MarkerFaceColor','black',...
      'MarkerEdgeColor','black');
%Ubicacion generada por trilateracion basica
plot(p_0(1),p_0(2),...
      'w','Marker','o',...
      'MarkerFaceColor','g',...
      'MarkerEdgeColor','g','MarkerSize',8)
%Ubicacion generada por trilateracion Gauss Newton
plot(posicion_final(1),posicion_final(2),...
      'w','Marker','o',...
      'MarkerFaceColor','r',...
      'MarkerEdgeColor','r','MarkerSize',10)
legend('Posición inicial (Po)','Posición Estimada');

```

Figura 2.54. Código de presentación de resultados 2

2.5 IMPLEMENTACIÓN DE LA INTERFAZ GRÁFICA

La interfaz gráfica se implementa con el editor de interfaces graficas de usuario, GUI, de Matlab (Figura 2.55), y presenta dos componentes: una interfaz de presentación y la interfaz de usuario como tal.

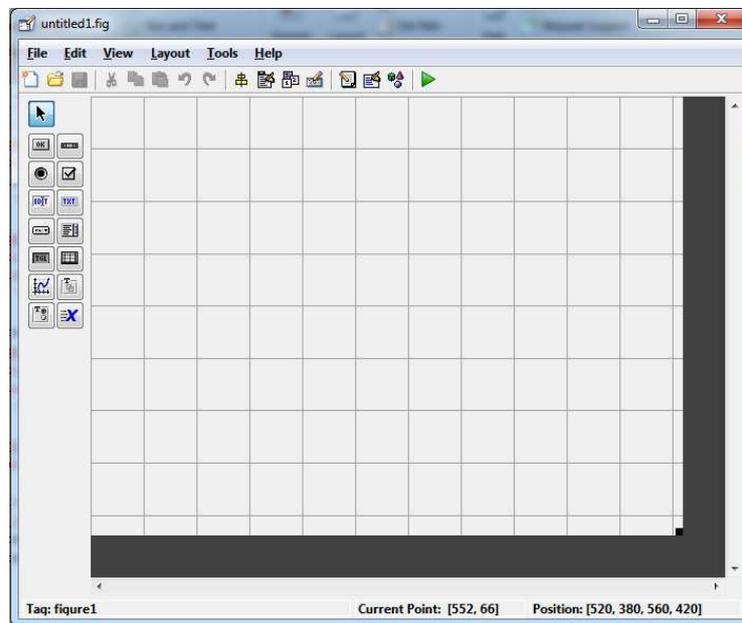


Figura 2.55. Editor de interfaces graficas de usuario, GUI

2.5.1 PORTADA

Esta interfaz presenta una lógica de programación simple basada en la presentación de información y dos botones (Figura 2.56).

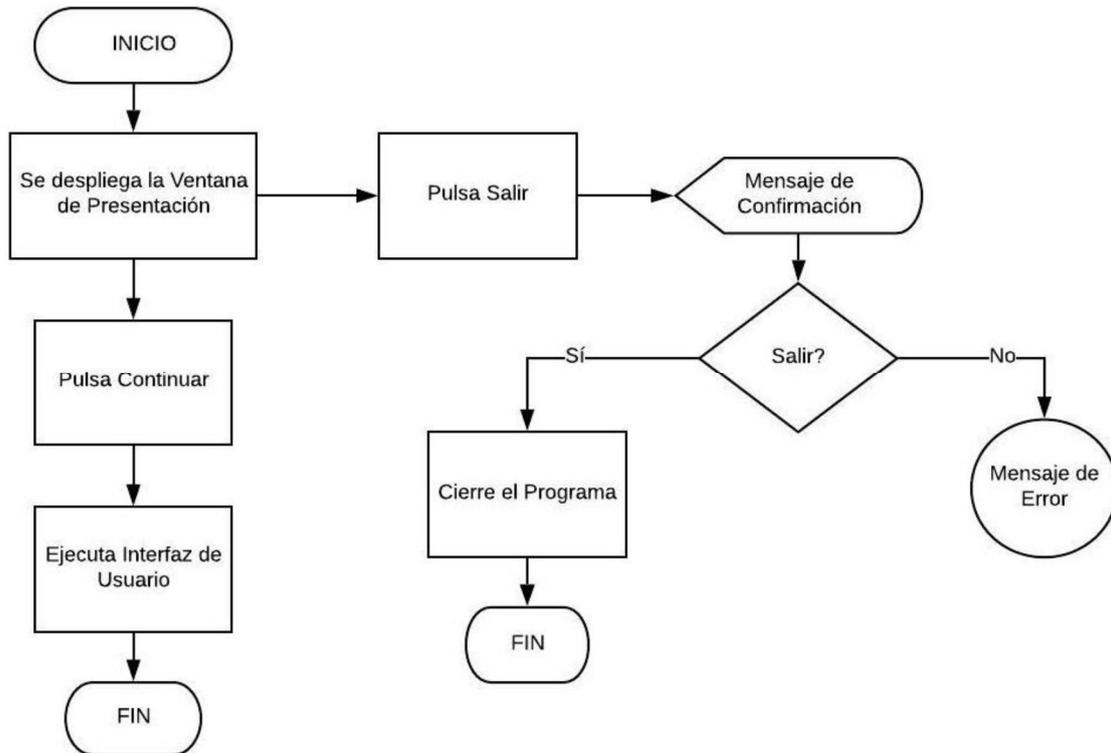


Figura 2.56. Diagrama de flujo de la interfaz de presentación

La disposición de los elementos de esta interfaz se muestra en la Figura 2.57.



Figura 2.57. Elementos de la interfaz de presentación

La ejecución de la presentación permite elegir entre dos opciones (botones): continuar y salir. Pulsar el botón continuar permite desplegar la interfaz de usuario, por otra parte, al pulsar el botón salir se despliega un mensaje de confirmación (Figura 2.58) si se presiona si, se cierra el programa, si se presiona no, se cierra el mensaje de confirmación y regresa a la pantalla de presentación. La Figura 2.59 muestra la ejecución de esta interfaz ya implementada.



Figura 2.58. Mensaje de confirmación para salir

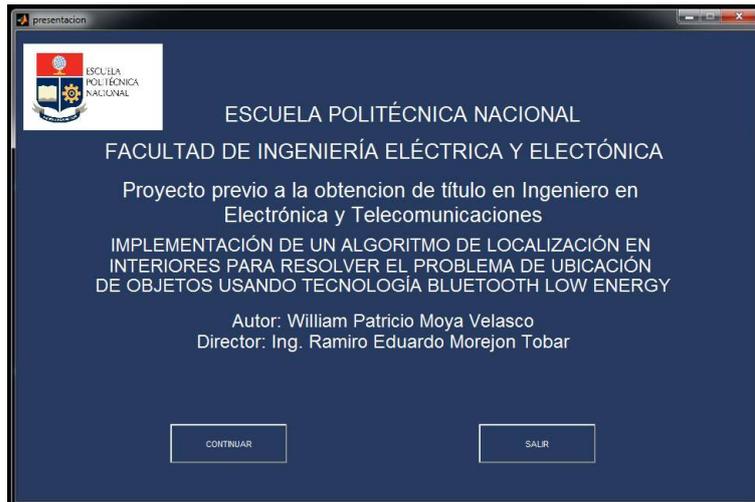


Figura 2.59. Pantalla de presentación

2.5.2 INTERFAZ DE USUARIO

El funcionamiento de esta interfaz es más complejo, ya que permite trabajar con el algoritmo y los dispositivos BLE. Su lógica de programación se presenta en la Figura 2.60.

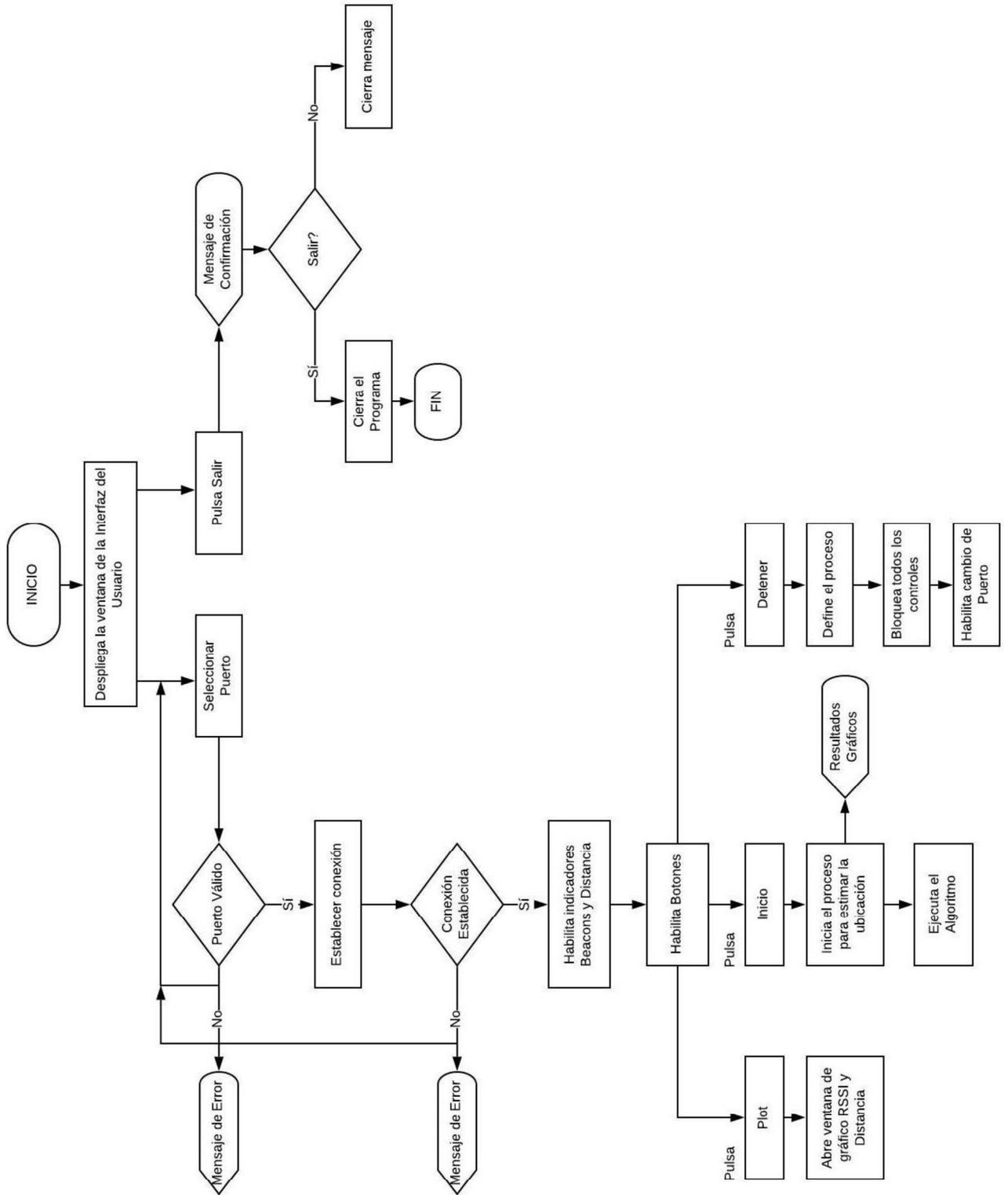


Figura 2.60. Diagrama de flujo de la interfaz de usuario

La disposición de los elementos de esta interfaz se muestra en la Figura 2.61.



Figura 2.61. Elementos de la interfaz de usuario

El funcionamiento y efecto en el programa de cada componente de la interfaz de usuario se resume en la Tabla 2.18.

Tabla 2.18. Funcionamiento y efecto en el programa de cada componente de la interfaz de usuario

Elemento	Funcionamiento	Efecto
Indicadores <i>Beacons</i>	Cambian de color en función de la actividad (Verde) o inactividad (Rojo) de su <i>Beacon</i> correspondiente.	Visualizar actividad (Figura 2.63)
Indicadores Distancia	Muestra la distancia actual estimada entre el dispositivo BLE central y el <i>Beacon</i> activo correspondiente.	Visualizar distancia (Figura 2.63)
Ingreso referencias	Permite cambiar manualmente los puntos de referencia en los cuales se ubicarán los <i>Beacons</i> dentro del área de prueba.	Ingreso de texto (Figura 2.63)
Botón Plot	Permite visualizar una gráfica de las últimas mediciones de RSSI y la estimación de la distancia, del <i>Beacon</i> activo correspondiente.	Despliega una ventana emergente que presenta el grafico de RSSI y distancia. (Figura 2.64)
Ubicación aproximada	Muestra el resultado de la estimación de la posición del dispositivo móvil obtenido con el algoritmo.	Visualizar resultado (Figura 2.65)

Resultado gráfico	Muestra el resultado del proceso de ubicación, de manera gráfica sobre un plano en dos dimensiones.	Visualizar resultado (Figura 2.65)
Combo box puerto	Despliega los puertos disponibles para establecer la comunicación serial con el módulo BLE central.	Si el puerto está disponible y se establece la conexión, habilita los indicadores de <i>Beacons</i> y distancia, los botones de plot, inicio y detener, y la ubicación aproximada. Caso contrario despliega mensajes de error al no seleccionar un puerto correcto o en el caso que no se logre establecer la comunicación serial. Una vez establecida la comunicación este elemento se bloquea. (Figura 2.63)
Botón iniciar	Inicia el proceso de estimación de la ubicación.	Ejecuta el algoritmo. Despliega los resultados gráficos y la ubicación aproximada. (Figura 2.65)
Botón detener	Detiene el proceso de estimación de la ubicación.	Inhabilita todos los controles y permite cambiar de puerto. (Figura 2.66)
Botón salir	Cierra completamente la interfaz.	Despliega un mensaje de confirmación para salir del programa. Si se presiona si, se cierra el programa, si se presiona no, se cierra el mensaje de confirmación y regresa a la interfaz de usuario. (Figura 2.67)

La Figura 2.62 muestra la interfaz de usuario al ejecutar el programa.



Figura 2.62. Interfaz de usuario



Figura 2.63. Pantalla al seleccionar un puerto y establecer la comunicaci3n de manera exitosa



Figura 2.64. Ventana del grafico de RSSI y distancia.

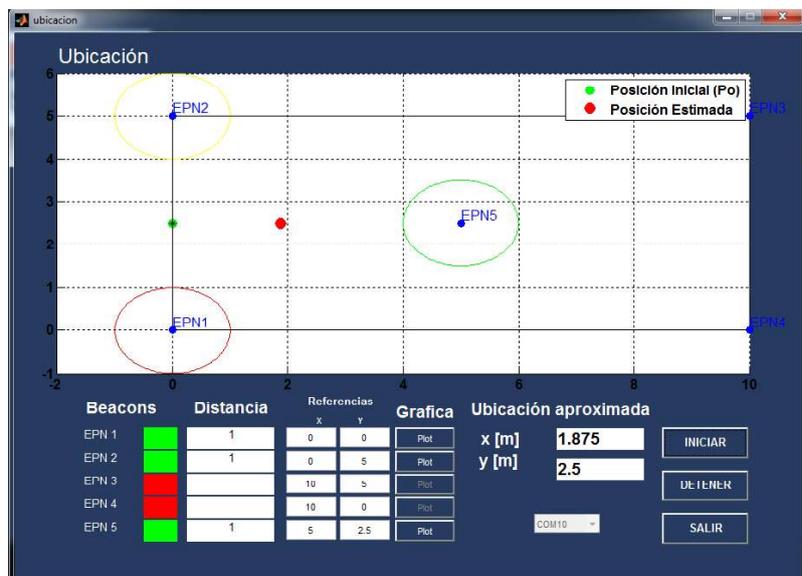


Figura 2.65. Pantalla al iniciar el proceso de ubicación y los resultados obtenidos

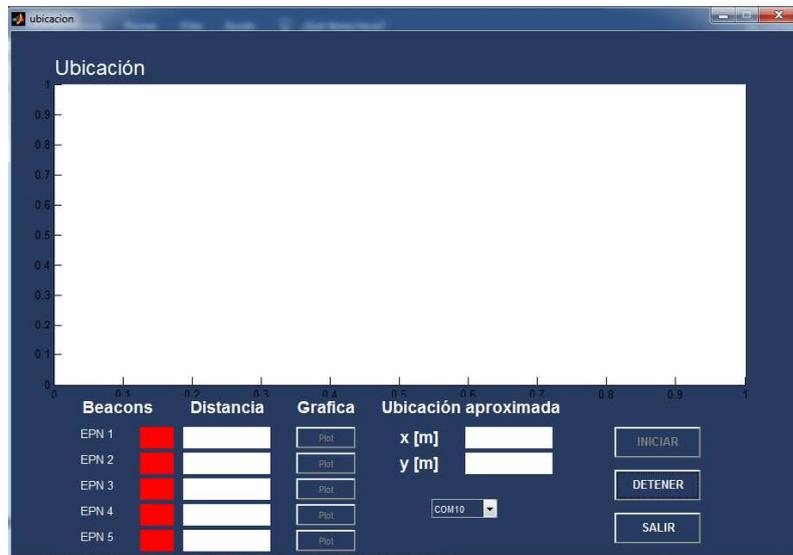


Figura 2.66. Pantalla al detener el proceso de ubicación



Figura 2.67. Mensaje de confirmación para salir del programa.

2.6 DESCRIPCIÓN DE ESQUEMAS DE PRUEBA

Todos los esquemas de prueba evaluarán el desempeño del algoritmo bajo un conjunto común de puntos de referencia. Se ubicará el dispositivo BLE central (móvil) en cada una de las posiciones descritas en la Figura 2.68 dentro de cada ambiente (simulación, ambiente de prueba y ambiente interior). Con lo cual se podrá contrastar los resultados obtenidos en cada situación y para cada configuración de la red inalámbrica.

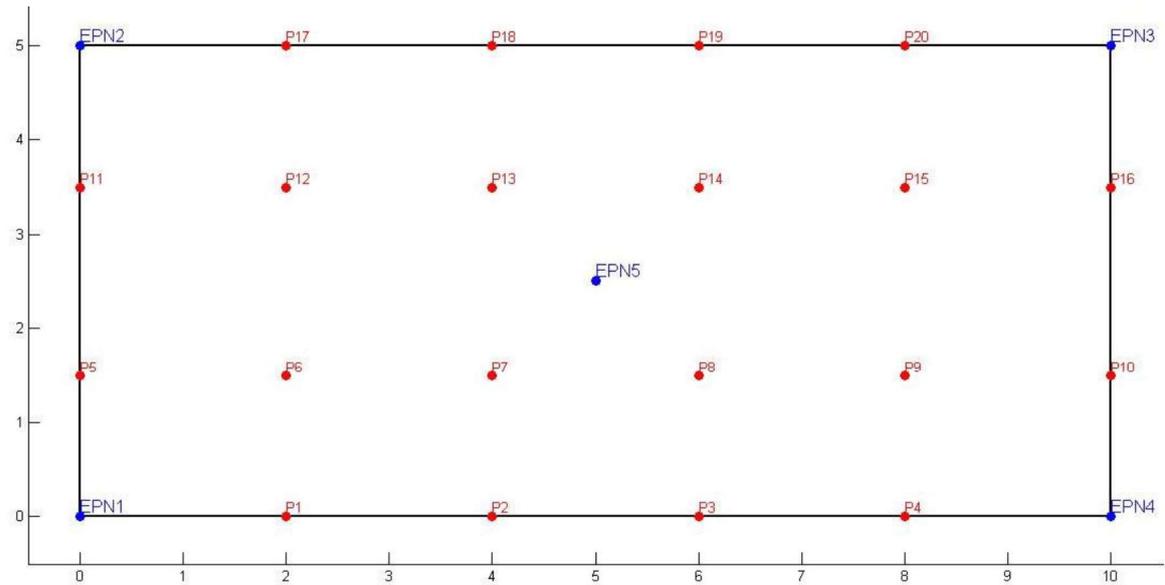


Figura 2.68. Distribución de puntos de prueba para evaluar el algoritmo (EPNX *Beacon*, P_i punto de prueba)

Las pruebas que se establecen para valorar la respuesta del algoritmo definen el número de *Beacons* que conformaran la red y la configuración de potencia e intervalo de transmisión de cada uno (Tabla 2.19).

Tabla 2.19. Pruebas para evaluar el algoritmo

Prueba	N° de <i>Beacons</i>	Potencia de tx [dBm]	Intervalo de tx [ms]
1	3 (EPN1 EPN2 EPN5)	6	100
2	4 (EPN1- EPN4)	6	100
3	5 (EPN1- EPN5)	6	100

Contrariamente a lo propuesto al inicio, se definen únicamente estas tres pruebas. Debido a que, conforme se incrementa el intervalo de tiempo entre cada *advertising* transmitido por los *Beacons*, el número de *Beacons* encontrados en cada búsqueda no corresponde al número de *Beacons* activos, es decir, se pierden mediciones simultaneas de todos los dispositivos disponibles. Esto se presenta debido a la diferencia entre el tiempo de búsqueda (parámetro no modificable menor a 4 segundos) definido por el fabricante y el tiempo máximo entre *advertisings* (7 segundos) [39] [40]. La Tabla 2.20 resume los resultados en los procesos de búsqueda configurando tiempos máximos y mínimos entre *advertisings*, donde se observa que al configurar al *Beacon* EPN1 con el intervalo máximo de transmisión, únicamente se consigue capturar 3 mediciones.

Tabla 2.20. Procesos de búsqueda configurando tiempos máximos y mínimos entre advertisings

Búsqueda	RSSI [dBm] de <i>Beacons</i> activos		
	EPN1	EPN3	EPN4
	t=7s	t=100ms	t=100ms
1	0	-49	-73
2	-59	-49	-69
3	0	-49	-71
4	-58	-49	-68
5	0	-48	-68
6	0	-48	-68
7	0	-48	-70
8	0	-49	-69
9	-58	-49	-75
10	0	-48	-69

Por otra parte, se plantean las pruebas únicamente con potencia de transmisión máxima (6 dBm), esto debido a que en las mediciones de corriente y consumo de energía presentadas en la Sección 1 no existe una diferencia considerable entre los consumos con una u otra potencia. Además, de los resultados obtenidos al definir el modelo de propagación (Sección 2.2) se encontró que las estimaciones de distancia entre dispositivos no difieren tanto entre sí para cada potencia; por otro lado, las señales de radiofrecuencia tienden a ser más estables con potencia de transmisión máxima.

Los tres escenarios que se presentan a continuación permitirán evaluar el comportamiento del algoritmo y establecer una comparativa de resultados.

2.6.1 ESCENARIO DE PRUEBAS EN SIMULACIÓN

La simulación permitirá determinar el comportamiento del algoritmo sin la necesidad de obtener los datos desde la red inalámbrica. Inicialmente se harán pruebas ingresando valores de distancias sin errores, a continuación, considerando la Ecuación 2.8 se adicionará a cada distancia un error aleatorio de acuerdo a los valores establecidos en la

Tabla 2.21. Rangos de variación de la distancia para pruebas. Los rangos de variación de la distancia se consideran en función de los valores obtenidos en la Sección 2.2.

Tabla 2.21. Rangos de variación de la distancia para pruebas

Prueba	Variación de las distancias [m]
1	0
2	0 - 1
3	1 - 3
4	1 - 7

Cabe destacar que en este esquema no se consideran las configuraciones de la Tabla 2.19, evidentemente por que las medidas no dependen de la red *Bluetooth*.

2.6.2 ESCENARIO DE PRUEBAS EN AMBIENTE DE PRUEBA

Se considera como ambiente de prueba una locación que minimice lo máximo posible los efectos que afectan a las señales de radio en los ambientes interiores, es decir, múltiples trayectorias, reflexiones y difracciones de la señal, entre otros. Por este motivo se eligió como espacio de prueba una sección de la pista de aterrizaje del antiguo aeropuerto de Quito (Figura 2.69), actual parque bicentenario, ya que no presenta ningún elemento reflectivo a parte del piso.



Figura 2.69. Ambiente de prueba en el parque bicentenario de Quito

Como se estableció en secciones anteriores el espacio de prueba será de 5 x 10 metros, con cada *Beacon* ubicado a una altura de 1.5 metros y el dispositivo móvil dispuesto en cada uno de los puntos de prueba (Figura 2.68). En este ambiente se realizarán mediciones en base a todas las pruebas descritas en la Tabla 2.19.

2.6.3 ESCENARIO DE PRUEBAS EN AMBIENTE INTERIOR

El ambiente interior definido para las pruebas es la casa comunal del barrio la Esperanza ubicado en San Juan de Morán, Quito. Esta infraestructura está compuesta por: paredes de bloque recubiertas con cemento y pintura, ventanas con marco de aluminio, puertas de vidrio, metal y madera, piso de baldosa y techo de concreto (Figura 2.70). De la misma manera que en el escenario anterior, se selecciona un espacio de 5 x 10 metros, con cada *Beacon* ubicado a una altura de 1.5 metros y el dispositivo móvil dispuesto en cada uno de los puntos de prueba (Figura 2.68). En este ambiente también se realizarán mediciones en base a todas las pruebas descritas en la Tabla 2.19.



Figura 2.70. Ambiente de prueba en la casa comunal

3. RESULTADOS Y DISCUSIÓN

El algoritmo de ubicación en interiores implementado basa su funcionamiento en las mediciones de potencia obtenidas de la red inalámbrica de tecnología *Bluetooth Low Energy*. En esta sección se presenta los resultados obtenidos al implementar la red *LE* y se valida el funcionamiento del algoritmo siguiendo las especificaciones de los escenarios de prueba descritos en la Sección 2.6. Adicionalmente, se analiza la relación existente entre las diferentes configuraciones de los dispositivos BLE en la red y la precisión final del algoritmo.

3.1 RED INALÁMBRICA IMPLEMENTADA

Los requerimientos que deben cumplir los módulos BLE y la red *Bluetooth Low Energy* se detallan en las secciones 1 y 2.3 respectivamente, a continuación, se presentan los resultados obtenidos al configurar e implementar los módulos y la red.

3.1.1 BEACONS



Figura 3.1. Módulo HM-10 como *Beacon*

El módulo HM-10 configurado y conectado como *Beacon* se presenta en la Figura 3.1, además sus parámetros de configuración final se presentan en la Figura 3.2.

```
COM10
OK
OK+NAME:EPN1 Nombre
OK+Get:0x0001 MAJOR
OK+Get:0x0001 MINOR
OK+Get:0 Intervalo de advertencia
OK+Get:3 Potencia de transmisión
OK+Get:3 Modo Non-Connectable
OK+Get:1 iBeacon activado
OK+Get:0 Modo ahorro de energía
```

Figura 3.2. Configuración final del módulo HM-10 como *Beacon* (Tabla 2.8)

Para verificar que el dispositivo está configurado como *Beacon*, se hace uso de la aplicación dartle.io disponible para iOS, la cual permite escanear *Beacons* desde un smartphone. La Figura 3.3 muestra el resultado al buscar *Beacons* activos con la aplicación y con el dispositivo BLE central, comprobando que la configuración de cada módulo es correcta y que sus identificadores son los mismos en ambos casos.

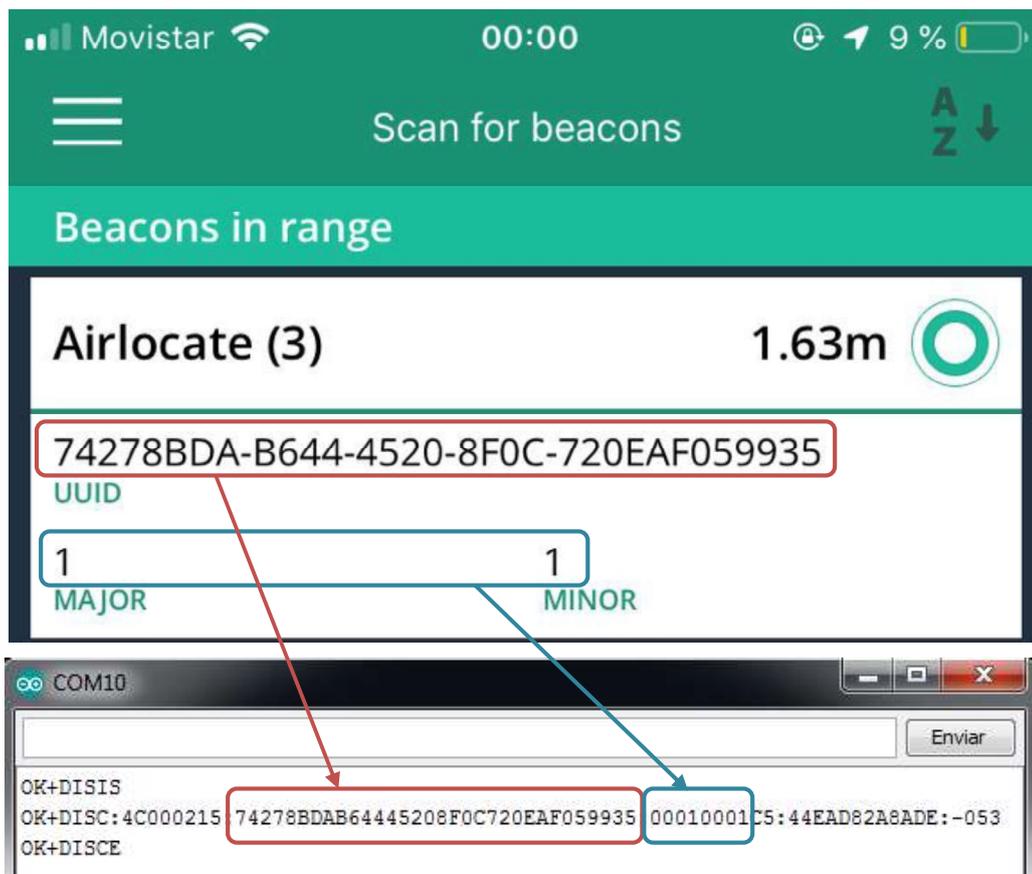


Figura 3.3. Búsqueda de *Beacons* activos (App y BLE central).

3.1.2 BLE CENTRAL

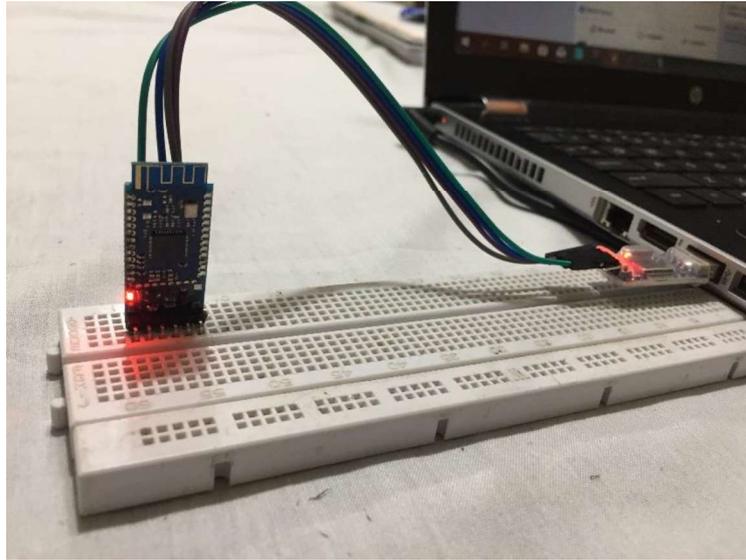


Figura 3.4. Módulo HM-10 como BLE central.

El módulo HM-10 configurado y conectado como BLE central se presenta en la Figura 3.4, además sus parámetros de configuración final se presentan en la Figura 3.5.

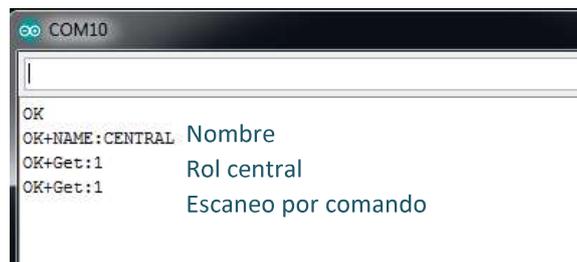


Figura 3.5. Configuración final del módulo HM-10 como BLE central (Tabla 2.7)

Para verificar que el dispositivo está configurado como BLE central, se inicia un proceso de búsqueda de *Beacons* activos y se compara con los resultados obtenidos en la búsqueda efectuada por la aplicación dartle.io disponible para iOS (Figura 3.3).

3.1.3 RED BLUETOOTH LOW ENERGY

La red implementada con el máximo número de *Beacons* disponible y el dispositivo central, se presenta en Figura 3.6, donde además se detallan las dimensiones del área de cobertura y la elevación de cada *Beacon* (Sección 2.3).



Figura 3.6. Red *Bluetooth Low Energy*

El resultado de un proceso de búsqueda de *Beacons* activos con la red funcionando se muestra en la Figura 3.7.

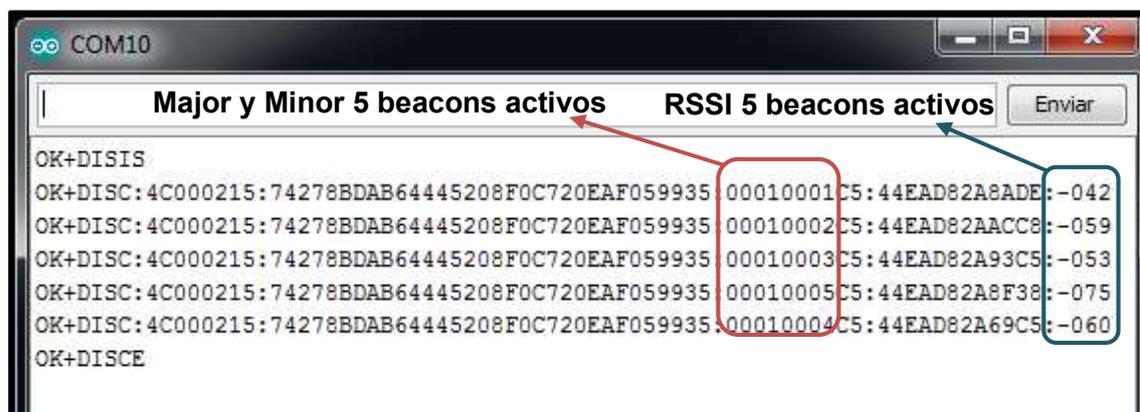


Figura 3.7. Búsqueda de *Beacons* activos con la red funcionando

La Figura 3.8 muestra los resultados al obtener las mediciones de potencia en el punto P_6 (Figura 2.68), además, hace una comparación gráfica entre las distancias reales calculadas y las distancias estimadas por el algoritmo. En esta se muestran las circunferencias de color rojo creadas con la distancia real entre el dispositivo central y cada *Beacon*, por su parte las circunferencias de color azul indican la distancia estimada con las mediciones de

potencia, además, la diferencia Δd entre las mismas. De igual manera, se presentan el punto real donde está ubicado el módulo central (P_6) y la ubicación estimada.

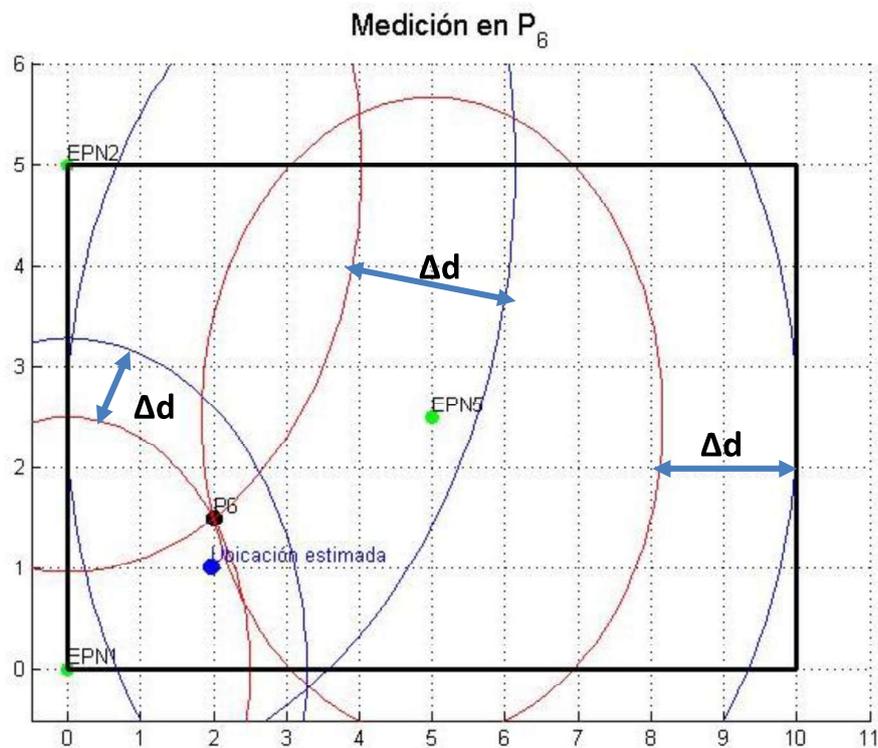


Figura 3.8. Estimación de la distancia y la ubicación en el punto P_6

De esta manera se valida el funcionamiento correcto de la red inalámbrica implementada. Sin embargo, como se mencionó en la Sección 2.2 y como se observa en la figura anterior, es importante destacar que el error al estimar la distancia influye directamente en el algoritmo, al disminuir la precisión final en el cálculo de ubicación del dispositivo central.

3.2 RESULTADOS DE SIMULACIÓN DEL ALGORITMO

Como se ha mencionado en el marco teórico y la implementación del algoritmo de ubicación, la técnica de trilateración presenta grandes ventajas, sin embargo, existen múltiples limitaciones. Es por esta razón, que las pruebas de simulación descritas a continuación tienen la finalidad de establecer la relación entre las variaciones que presenta el cálculo de la distancia y la precisión final con la cual se ubica el móvil.

3.2.1 PRUEBA 1: ESTIMACIÓN DE LA POSICIÓN SIN ERROR EN LAS DISTANCIAS.

Para realizar la Prueba 1 se calcula directamente la distancia entre el dispositivo central ubicando en un punto de prueba P_i (Figura 2.68) y la ubicación de cada uno de los cinco *Beacons* (Ecuación 2.9). Una vez que se calculó las cinco distancias, se procedió a estimar la posición del dispositivo central. En el Anexo C se recogen los resultados obtenidos para cada posición P_i . Cabe mencionar que se realizaron tres simulaciones por cada punto de prueba.

A continuación, en la Figura 3.9 se presenta la gráfica obtenida a partir de la ejecución del algoritmo, donde se muestra la ubicación de cada punto de prueba y la ubicación estimada del mismo punto. Además, la Figura 3.10 presenta el error en la estimación final de la posición para cada punto de prueba.

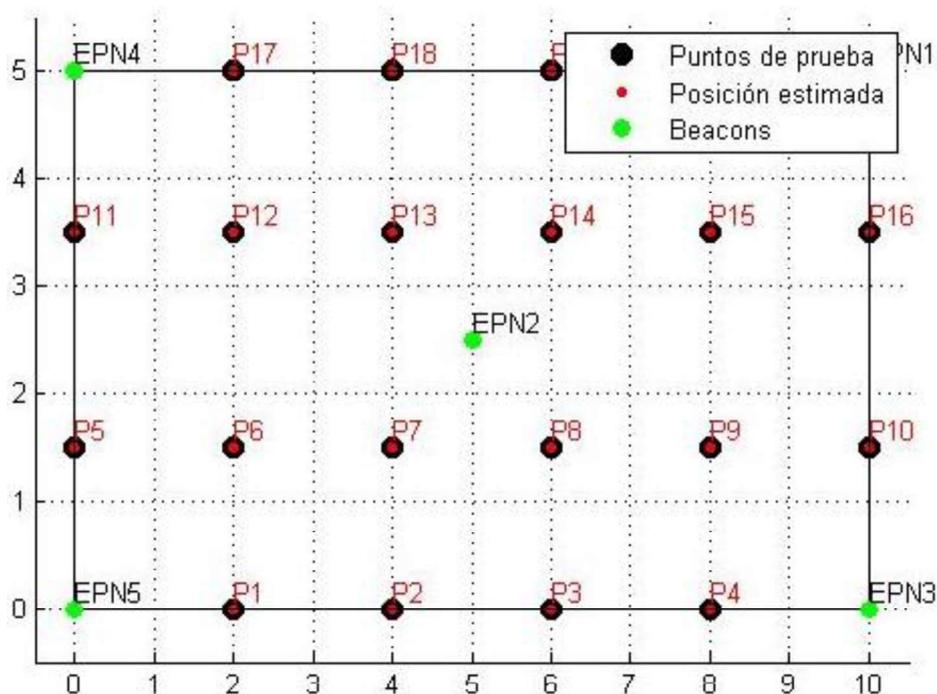


Figura 3.9. Ubicaciones estimadas Prueba 1 (Simulación)

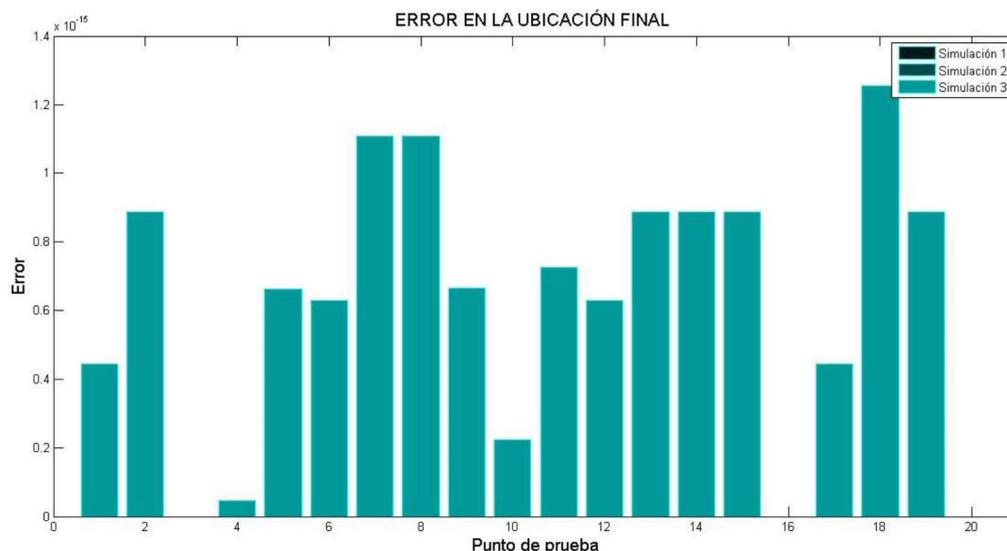


Figura 3.10. Error en la ubicación Prueba 1 (Simulación)

Los resultados obtenidos muestran que, si las distancias no presentan ningún error el algoritmo estima la posición con un error medio igual a 6.19×10^{-16} ; sin embargo, este comportamiento es ideal, ya que las mediciones de potencia obtenidas de una red inalámbrica real siempre van a tener errores y en consecuencia las distancias calculadas también incluirán un error.

3.2.2 PRUEBAS 2, 3 Y 4: ESTIMACIÓN DE LA POSICIÓN CON ERRORES EN LAS DISTANCIAS.

Para realizar las Pruebas 2, 3 y 4 se calcula directamente la distancia entre el dispositivo central ubicando en un punto de prueba P_i (Figura 2.68) y la ubicación de cada uno de los cinco *Beacons* (Ecuación 2.9), adicionalmente a cada distancia se le suma un error aleatorio dentro uno de los rangos definidos en la Tabla 2.21 para cada prueba. Una vez que se calculó las cinco distancias, se procedió a estimar la posición del dispositivo central. En el Anexo C se recogen los resultados obtenidos para cada posición P_i . Cabe mencionar que se realizaron tres simulaciones por cada punto de prueba y por cada Prueba.

A continuación, en la Figura 3.11, Figura 3.12 y Figura 3.13 se presentan las gráficas obtenidas a partir de la ejecución del algoritmo, donde se muestra la ubicación de cada punto de prueba y la ubicación estimada del mismo punto para las Pruebas 2, 3 y 4 respectivamente. Además, la Figura 3.14, Figura 3.15 y Figura 3.16 presentan el error en la estimación final de cada posición y por cada Prueba.

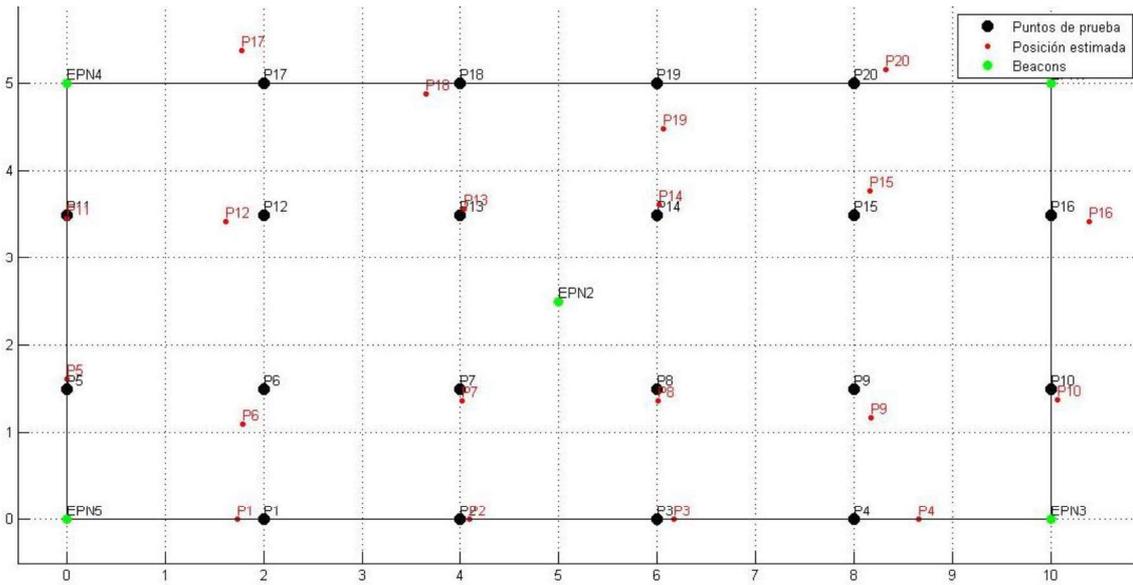


Figura 3.11. Ubicaciones estimadas Prueba 2 (Simulación)

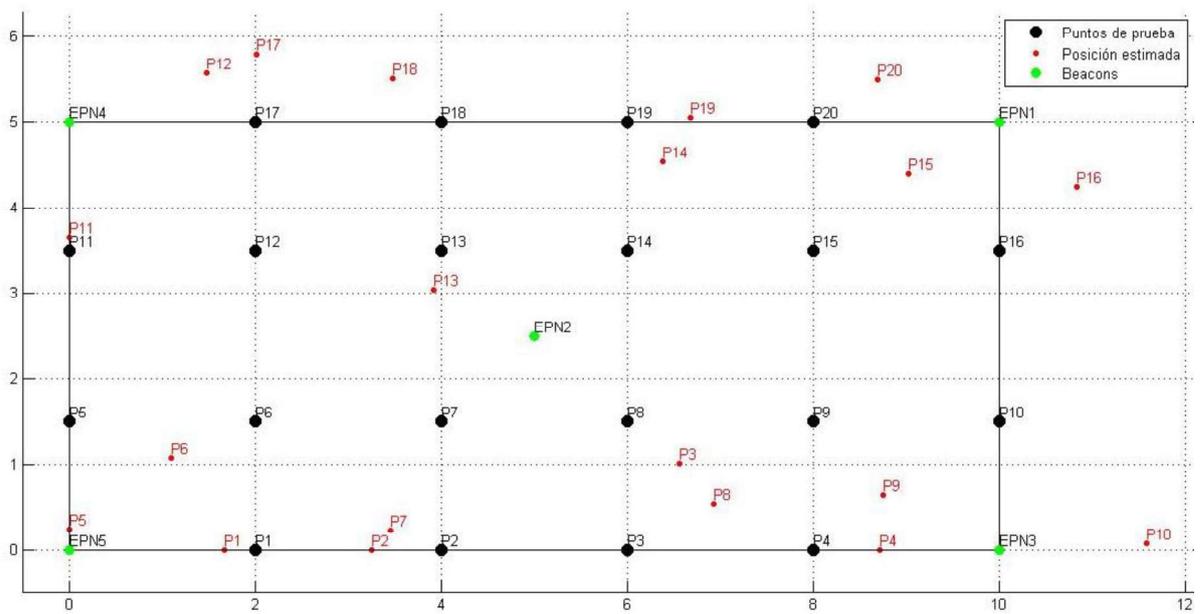


Figura 3.12. Ubicaciones estimadas Prueba 3 (Simulación)

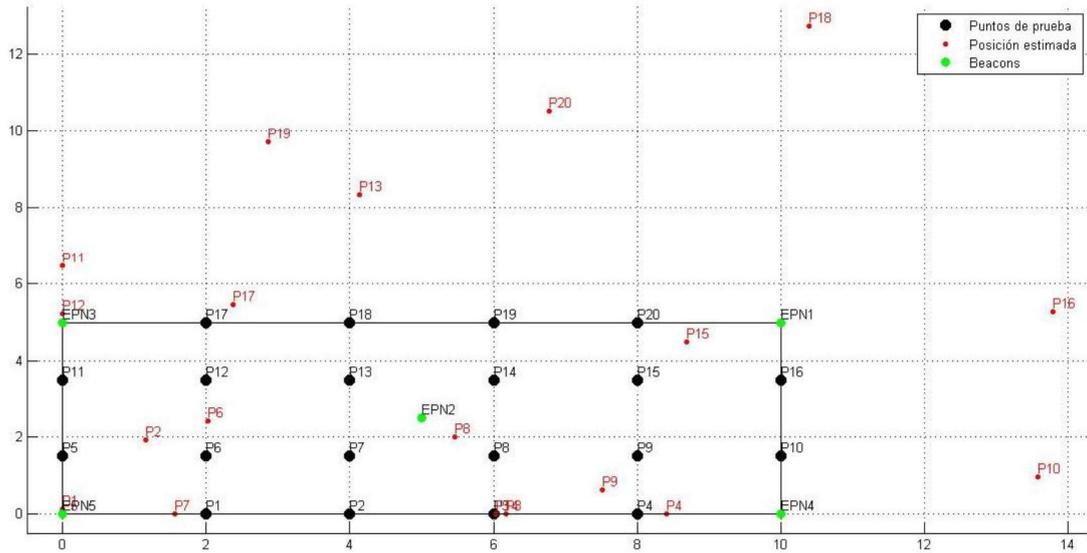


Figura 3.13. Ubicaciones estimadas Prueba 4 (Simulación)

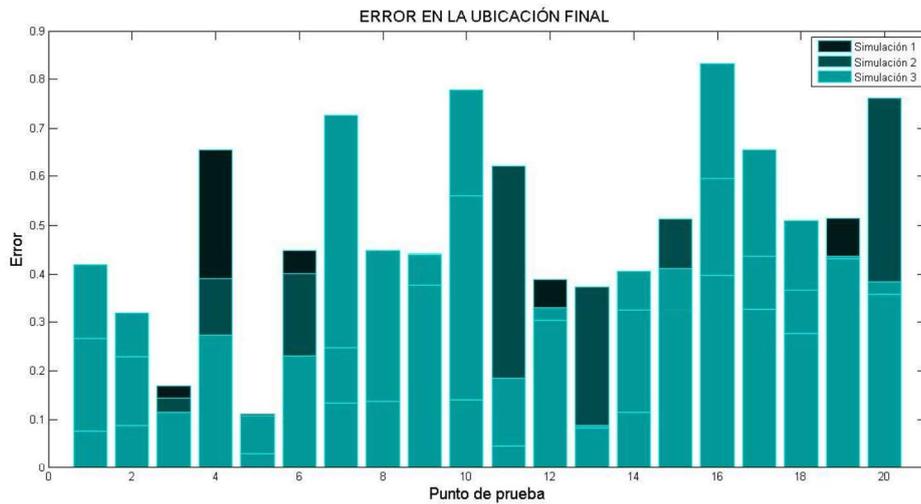


Figura 3.14. Error en la ubicación Prueba 2 (Simulación)

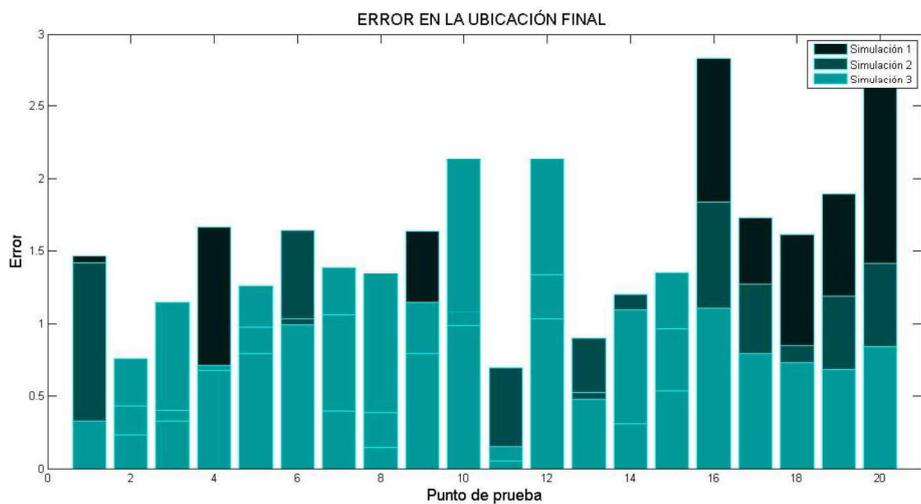


Figura 3.15. Error en la ubicación Prueba 3 (Simulación)

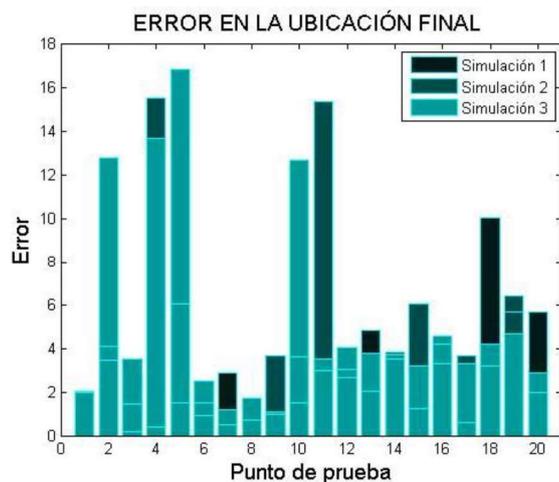


Figura 3.16. Error en la ubicación Prueba 4 (Simulación)

Los resultados obtenidos muestran que, conforme el error en las distancias aumenta, la precisión final del algoritmo disminuye, es decir el error al ubicar el dispositivo móvil aumenta.

Tabla 3.1. Error y porcentajes de ubicación dentro de rangos de precisión para cada prueba

	Prueba 1	Prueba 2	Prueba 3	Prueba 4
Error medio	6,19E-16	0,35	1,05	4,15
Error máximo	1,26E-15	0,83	2,90	16,81
Error mínimo	0,00E+00	0,03	0,06	0,18
Porcentaje de ubicación dentro del rango de error				
< 1m	100	100	50	13,33
1m - 2m	0	0	43,33	15,00
2m<	0	0	6,67	71,67

La Tabla 3.1 resume los valores de error y los porcentajes de ubicación dentro de rangos de precisión para cada prueba. Como se puede observar el error en la ubicación va desde un mínimo de 0 metros a un máximo de 16.81 metros, siendo los errores obtenidos en la Prueba 4 inaceptables para el funcionamiento idóneo del algoritmo. Sin embargo, este comportamiento ratifica la funcionalidad del algoritmo al desenvolverse dentro de la red *Bluetooth* LE, ya que las distancias estimadas en base a las mediciones de potencia obtenidas de la red inalámbrica presentan un error medio de 0.96 metros con una variación entre 0 y 2 metros. En consecuencia, las pruebas de simulación 2 y 3 son las que más se ajustan al comportamiento de la red inalámbrica, y los resultados obtenidos en estos dos escenarios son aceptables al alcanzar un 96% de ubicaciones con un error menor o igual a 2 metros.

3.3 RESULTADOS EN AMBIENTE DE PRUEBA

Las pruebas realizadas en este ambiente tienen la finalidad de comprobar si el algoritmo es capaz de responder de manera adecuada bajo condiciones en las que las señales no se ven tan afectadas como en espacios interiores. Por este motivo, los procesos de evaluación descritos a continuación tienen como objetivo determinar la precisión y limitaciones iniciales que se alcanzan con el algoritmo.

3.3.1 PRUEBAS 1, 2 Y 3: ESTIMACIÓN DE LA POSICIÓN CON TRES, CUATRO Y CINCO BEACONS.

Para realizar las Pruebas 1, 2 y 3 se dispone de tres, cuatro y cinco *Beacons* activos respectivamente, y en todos los casos se ubica al dispositivo central en cada punto de prueba P_i , como se muestra en la Figura 3.17. Una vez ubicado el dispositivo central, se procedió a estimar su posición. En el Anexo C se recogen los resultados obtenidos para cada Prueba en cada una de las posiciones P_i . Es importante mencionar que se realizaron tres mediciones por cada punto de prueba.

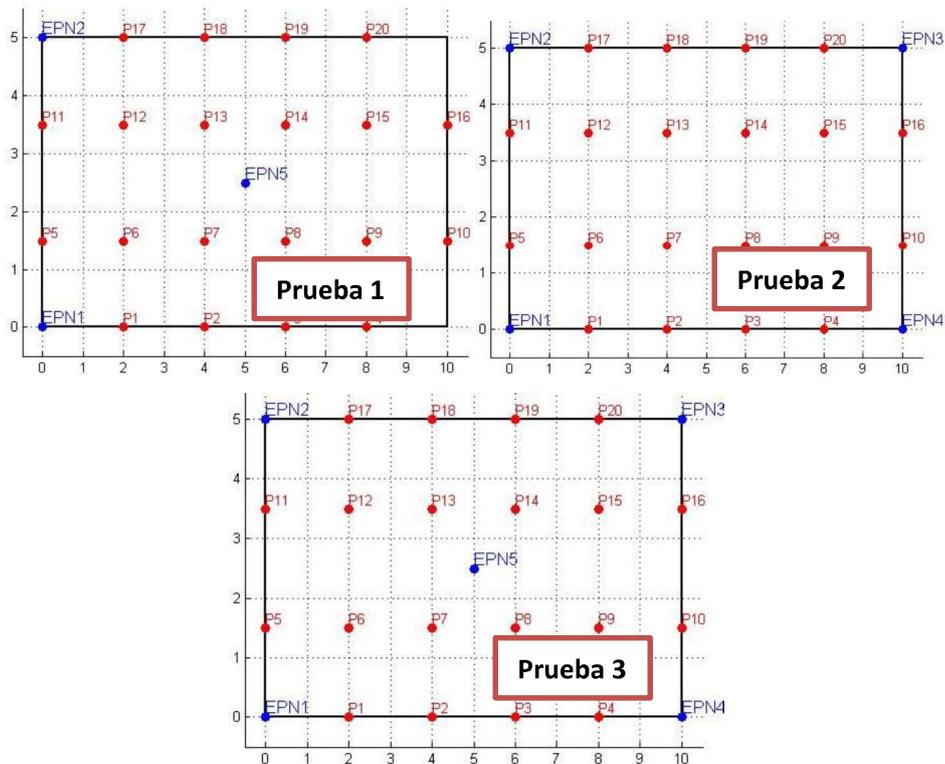


Figura 3.17. Escenarios de Prueba

A continuación, en las siguientes figuras se presentan las gráficas obtenidas a partir de la ejecución del algoritmo, donde se muestra la ubicación de cada punto de prueba y la ubicación estimada del mismo punto. Además, se presenta el error en la estimación final de la posición para cada punto de prueba.

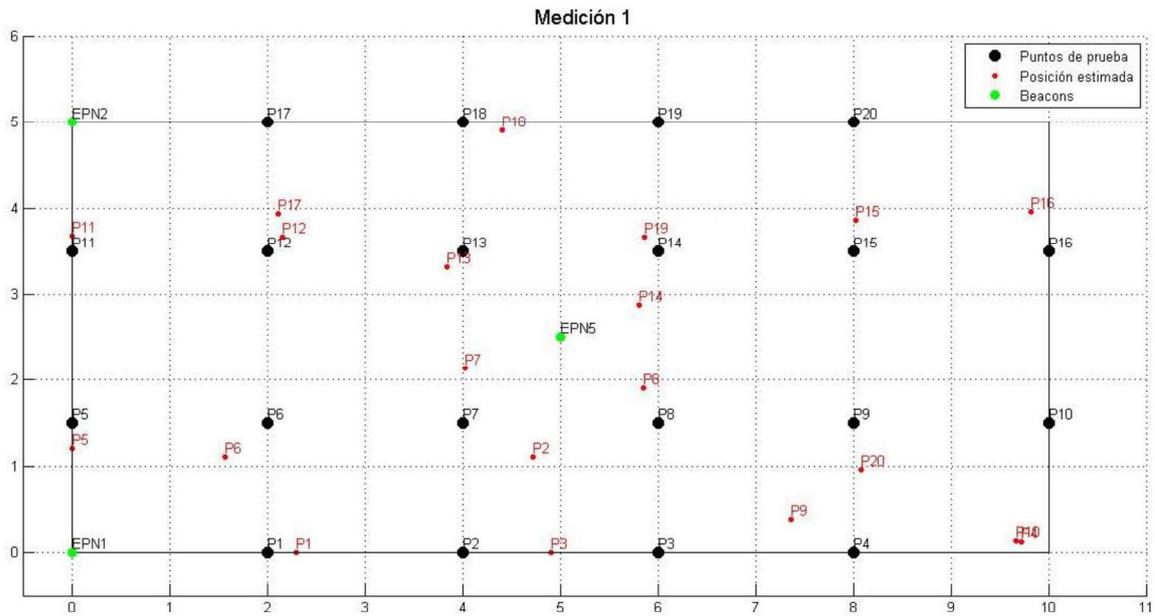


Figura 3.18. Ubicaciones estimadas Prueba 1 (Ambiente de prueba)

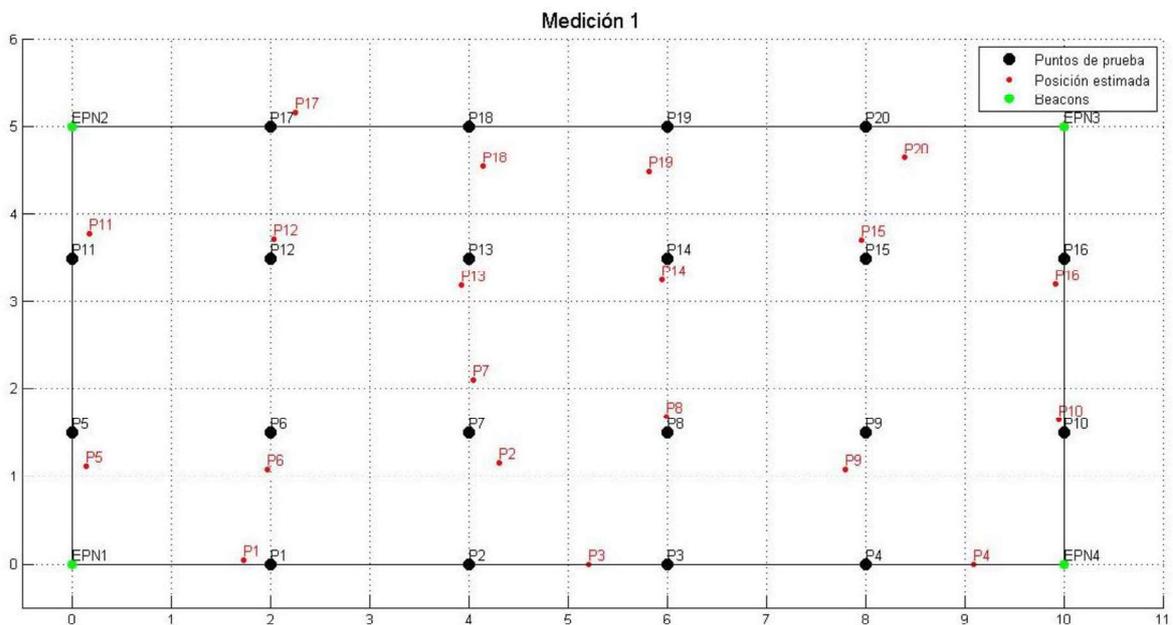


Figura 3.19. Ubicaciones estimadas Prueba 2 (Ambiente de prueba)

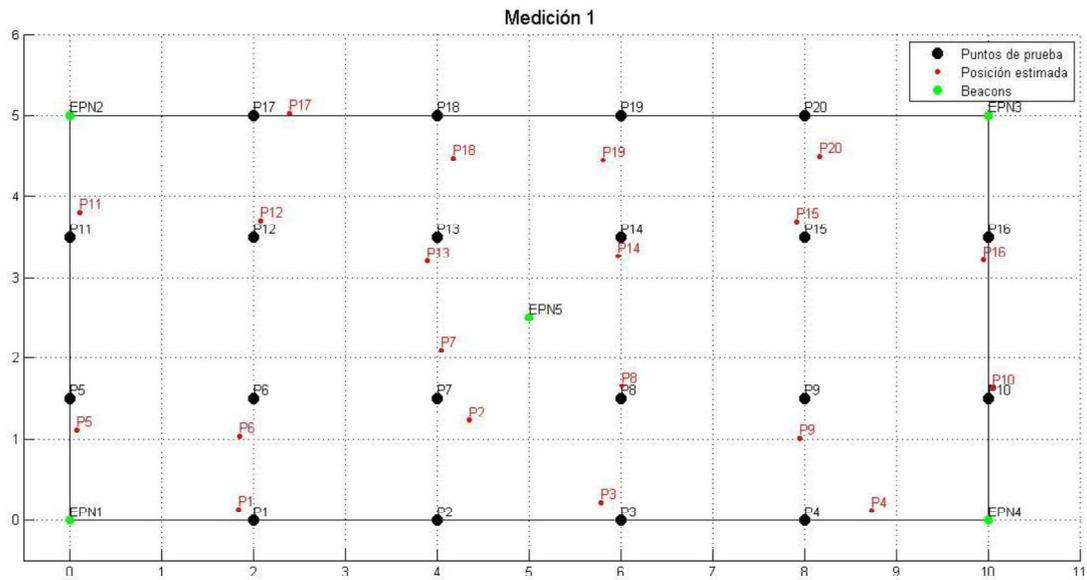


Figura 3.20. Ubicaciones estimadas Prueba 3 (Ambiente de prueba)

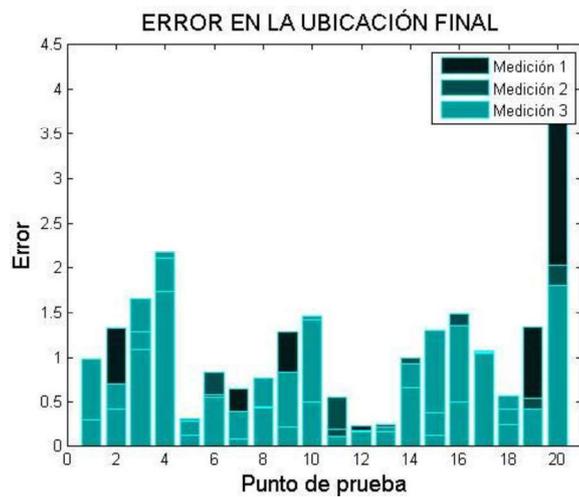


Figura 3.21. Error en la ubicación Prueba 1 (Ambiente de prueba)

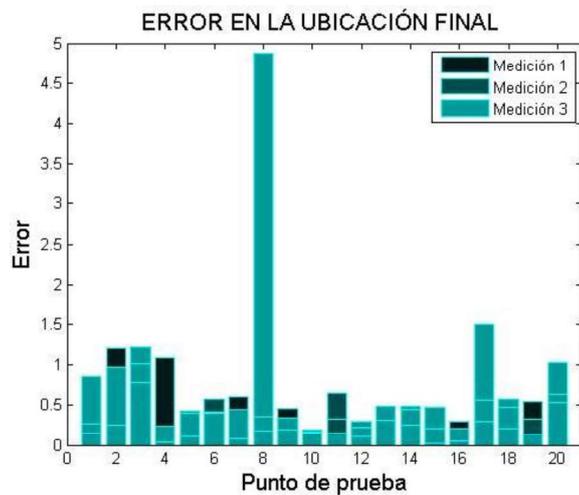


Figura 3.22. Error en la ubicación Prueba 1 (Ambiente de prueba)

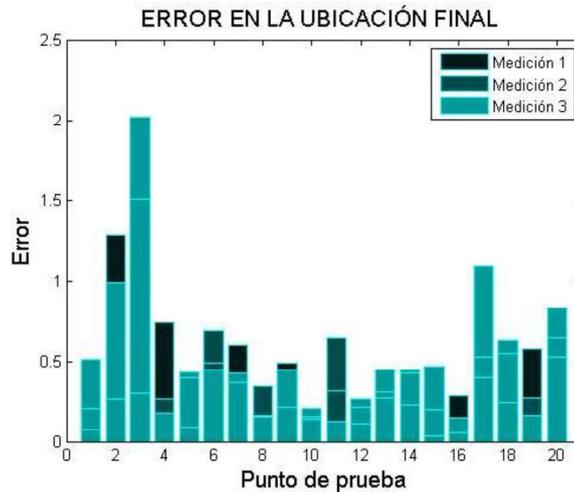


Figura 3.23. Error en la ubicación Prueba 1 (Ambiente de prueba)

Como se puede observar, los resultados obtenidos en las tres pruebas son similares. Esto debido a que las señales no sufren tanta degradación como en un ambiente interior real, de tal manera que se pueden obtener estimaciones de la ubicación bastante adecuadas. Sin embargo, existen aún posiciones que presentan un error mayor a dos metros en la ubicación estimada, esto se presenta por el uso compartido de los tres canales de *advertising* disponibles para todos los *Beacons*, además de los efectos generados al no ser un ambiente ideal que presente reflexión nula de las señales en el piso.

Tabla 3.2. Comparación de resultados alcanzados en las tres pruebas

	Prueba 1	Prueba 2	Prueba 3
Error medio	0,83	0,51	0,44
Error máximo	2,77	2,36	1,61
Error mínimo	0,12	0,11	0,10
Porcentaje de ubicación dentro del rango de error			
< 1m	66,67	88,33	93,33
1m - 2m	26,67	10,00	5,00
2m<	6,67	1,67	1,67

La Tabla 3.2 expone los rangos de error obtenidos para cada prueba dentro de este ambiente, y se observa que a mayor cantidad de *Beacons* activos la estimación de la ubicación es más precisa. Por lo contrario, a lo que se podría pensar, estos resultados únicamente reflejan el funcionamiento adecuado del algoritmo al igual que los resultados obtenidos en las pruebas de simulación, ya que el objetivo de la implementación de este sistema es que su funcionamiento sea adecuado en ambientes interiores reales.

3.4 RESULTADOS EN AMBIENTE INTERIOR

Las pruebas realizadas en el ambiente interior tienen la finalidad de comprobar la validez de los resultados obtenidos al estimar la posición del móvil dentro del espacio para el cual fue diseñado. Es por esta razón, que los procesos de evaluación descritos a continuación tienen mayor importancia para determinar la precisión y limitaciones finales que se alcanzan con el algoritmo.

3.4.1 PRUEBA 1: ESTIMACIÓN DE LA POSICIÓN CON TRES *BEACONS*.

Para realizar la Prueba 1 se dispone únicamente de tres *Beacons* activos (EPN1, EPN2 y EPN5) y se ubica al dispositivo central en cada punto de prueba P_i , como se muestra en la Figura 3.24. Una vez ubicado el dispositivo central, se procedió a estimar su posición. En el Anexo C se recogen los resultados obtenidos para cada posición P_i . Cabe mencionar que se realizaron tres mediciones por cada punto de prueba.

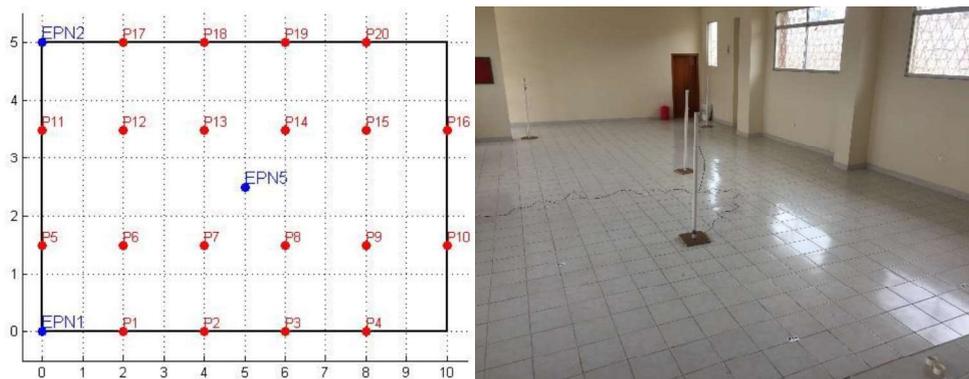


Figura 3.24. Escenario de Prueba con tres *Beacons* activos

A continuación, en la Figura 3.25 se presenta la gráfica obtenida a partir de la ejecución del algoritmo, donde se muestra la ubicación de cada punto de prueba y la ubicación estimada del mismo punto. Además, la Figura 3.26 presenta el error en la estimación final de la posición para cada punto de prueba.

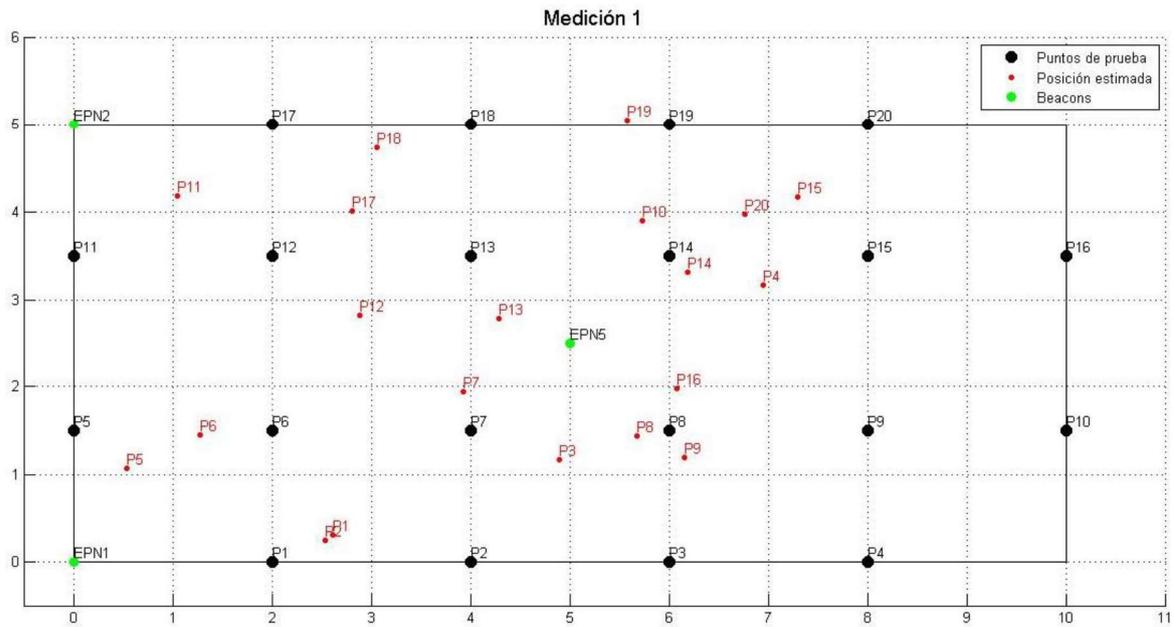


Figura 3.25. Ubicaciones estimadas Prueba 1 (Indoor)

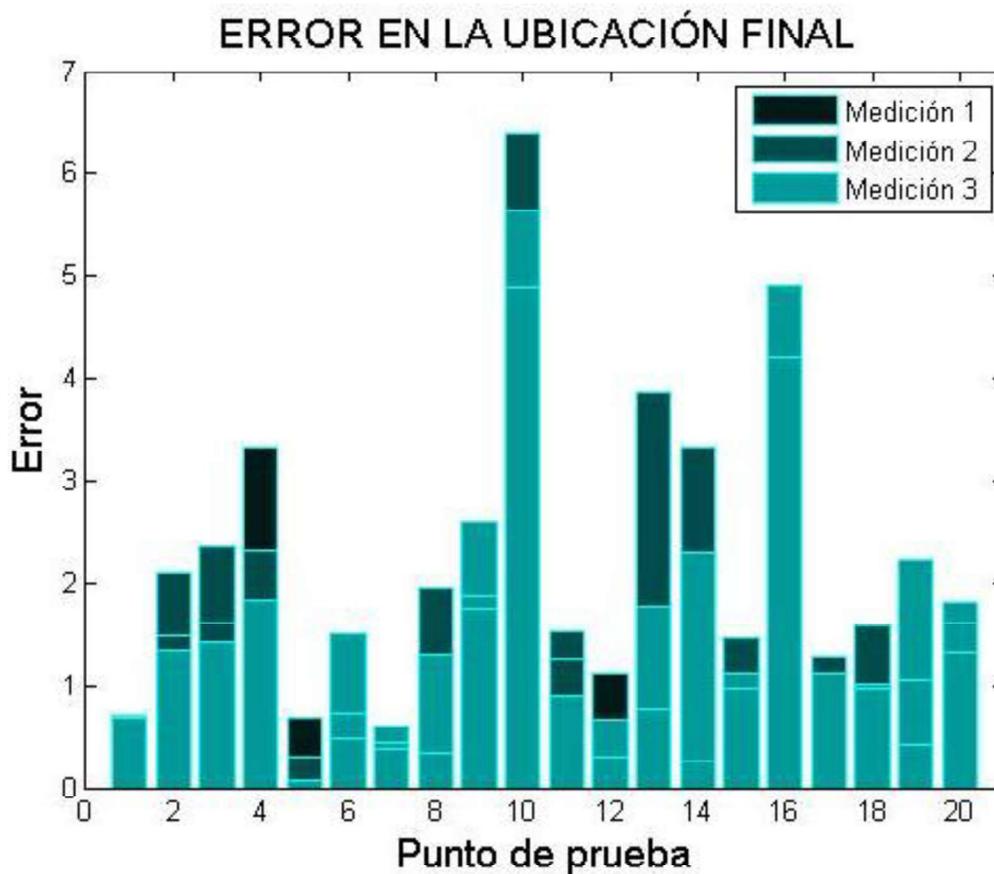


Figura 3.26. Error en la ubicación Prueba 1 (Indoor)

Los resultados obtenidos muestran que, conforme la separación entre los *Beacons* y el dispositivo central aumenta, el error al ubicar el dispositivo móvil aumenta también. La

Tabla 3.3 resume los errores cometidos al estimar la ubicación del módulo central BLE y los porcentajes de ubicaciones logradas para cada rango de error.

Tabla 3.3. Error y porcentajes de ubicación dentro de rangos de precisión para cada medición (Prueba 1 *indoor*)

	Medición 1	Medición 2	Medición 3
Error medio	1,45	1,93	1,74
Error máximo	4,89	6,41	5,64
Error mínimo	0,26	0,29	0,07
Porcentaje de ubicación dentro del rango de error			
< 1m	50	25	25
1m - 2m	35	40	50
2m<	15	35	25

Como se puede observar el error en la ubicación va desde un mínimo de 0.07 metros a un máximo de 6.41 metros, con un 75% de posiciones con un error menor o igual a dos metros, lo cual permite concluir que el algoritmo implementado responde de manera aceptable dentro del ambiente interior para el cual se lo diseñó. Sin embargo, los resultados ilustrados en la Figura 3.25 reflejan que la mayor parte de ubicaciones estimadas en puntos que se alejan de los tres *Beacons* activos presentan errores elevados, lo cual indica que la efectividad del algoritmo está ligada a la separación que presente el dispositivo central con respecto a los nodos activos. Entonces, en primera instancia se puede hablar de zonas de cobertura de la red inalámbrica, dentro de las cuales se obtienen mejores resultados al estimar la posición. Esta hipótesis se podrá validar con los resultados obtenidos en las Pruebas 2 y 3 que se presentan a continuación.

3.4.2 PRUEBA 2: ESTIMACIÓN DE LA POSICIÓN CON CUATRO *BEACONS*.

Esta Prueba se realiza con cuatro *Beacons* activos (EPN1, EPN2, EPN3 y EPN4) y se ubica al dispositivo central en cada punto de prueba P_i , como se muestra en la Figura 3.27. Una vez ubicado el dispositivo central, se procedió a estimar su posición. En el Anexo C se recogen los resultados obtenidos para cada posición P_i . Al igual que en la Prueba 1 se realizaron tres mediciones por cada punto de prueba.



Figura 3.27. Escenario de Prueba con cuatro *Beacons* activos

A continuación, en la Figura 3.28 se presenta la gráfica obtenida a partir de la ejecución del algoritmo, donde se muestra la ubicación de cada punto de prueba y la ubicación estimada del mismo punto. Además, la Figura 3.29 presenta el error en la estimación final de la posición para cada punto de prueba.

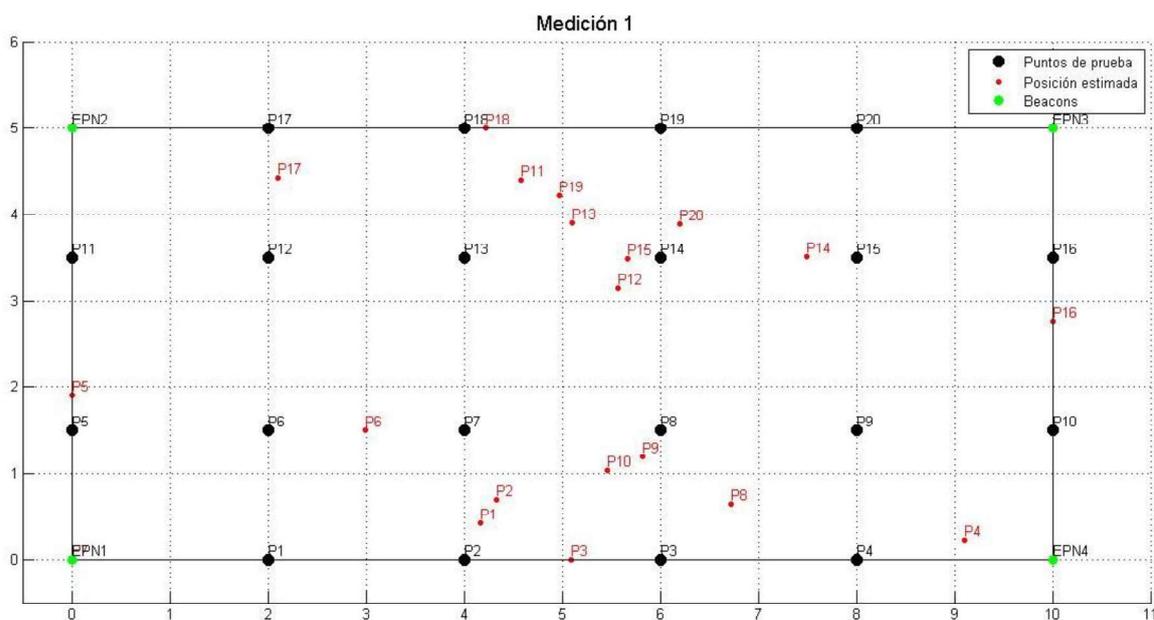


Figura 3.28. Ubicaciones estimadas Prueba 2 (*Indoor*)

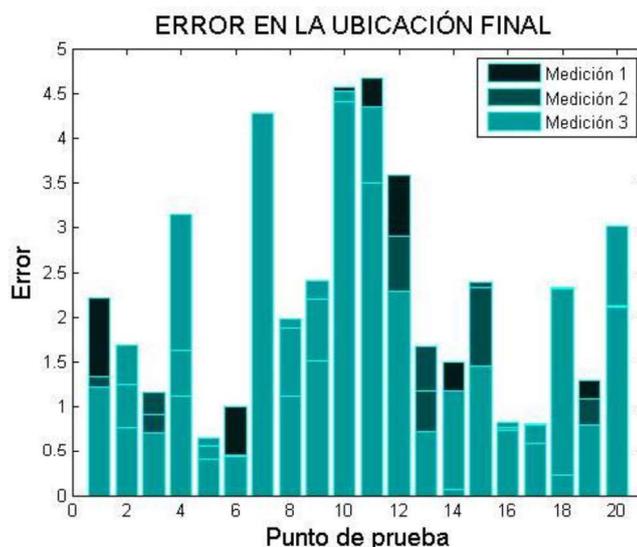


Figura 3.29. Error en la ubicación Prueba 2 (*Indoor*)

Los resultados obtenidos ratifican lo planteado en la Prueba 1 evidenciando que, conforme la separación entre los *Beacons* y el dispositivo central aumenta, el error al ubicar el dispositivo móvil aumenta también. Esto es justificable debido a que en esta prueba para cada punto, dos de los *Beacons* están a una distancia mayor a 5 metros del dispositivo central y de estos, al menos uno presenta distancias mayores entre siete y nueve metros. La Tabla 3.4 resume los errores cometidos al estimar la ubicación del módulo central BLE y los porcentajes de ubicaciones logradas para cada rango de error.

Tabla 3.4. Error y porcentajes de ubicación dentro de rangos de precisión para cada medición (Prueba 2 *indoor*)

	Medición 1	Medición 2	Medición 3
Error medio	1,84	1,80	1,94
Error máximo	4,68	4,40	4,52
Error mínimo	0,22	0,06	0,45
Porcentaje de ubicación dentro del rango de error			
< 1m	35	25	35
1m - 2m	25	40	25
2m<	40	35	40

Como se puede observar el error en la ubicación para esta prueba aumenta en comparación a la primera, ya que el número de posiciones estimadas con un error mayor a dos metros cambia del 25% a un 38.33%. Estos resultados reflejan el incremento de las separaciones, tomando en cuenta que el error en la estimación de las distancias aumenta para en rango entre 5 y 9 metros, como se planteó en la Sección 2.2. Además, según lo expuesto en el marco teórico, los dispositivos al no establecer conexiones utilizan

únicamente los tres canales de advertising, por lo tanto existe la posibilidad de que en cierto instante de tiempo dos *Beacons* estén transmitiendo en el mismo canal, generando de esta manera interferencias en las señales.

Por otro lado, como se mencionó en la prueba anterior, la zona de cobertura de la red aumenta, denotando así que, mientras más pequeña sea la el área, mejores serán los resultados se obtienen al estimar la posición.

3.4.3 PRUEBA 3: ESTIMACIÓN DE LA POSICIÓN CON CINCO BEACONS.

La última Prueba se efectúa con cinco *Beacons* activos (EPN1, EPN2, EPN3, EPN4 y EPN5) y se ubica al dispositivo central en cada punto de prueba P_i , como se muestra en la Figura 3.30. Una vez ubicado el dispositivo central, se procedió a estimar su posición. En el Anexo C se recogen los resultados obtenidos para cada posición P_i . Al igual que en las Pruebas anteriores se realizaron tres mediciones por cada punto de prueba.

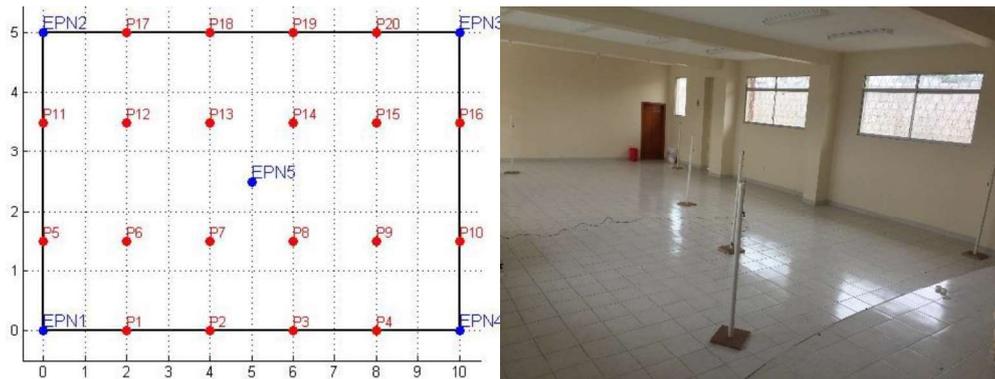


Figura 3.30. Escenario de Prueba con cinco *Beacons* activos

A continuación, en la Figura 3.31 se presenta la gráfica obtenida a partir de la ejecución del algoritmo, donde se muestra la ubicación de cada punto de prueba y la ubicación estimada del mismo punto. Además, la Figura 3.32 presenta el error en la estimación final de la posición para cada punto de prueba.

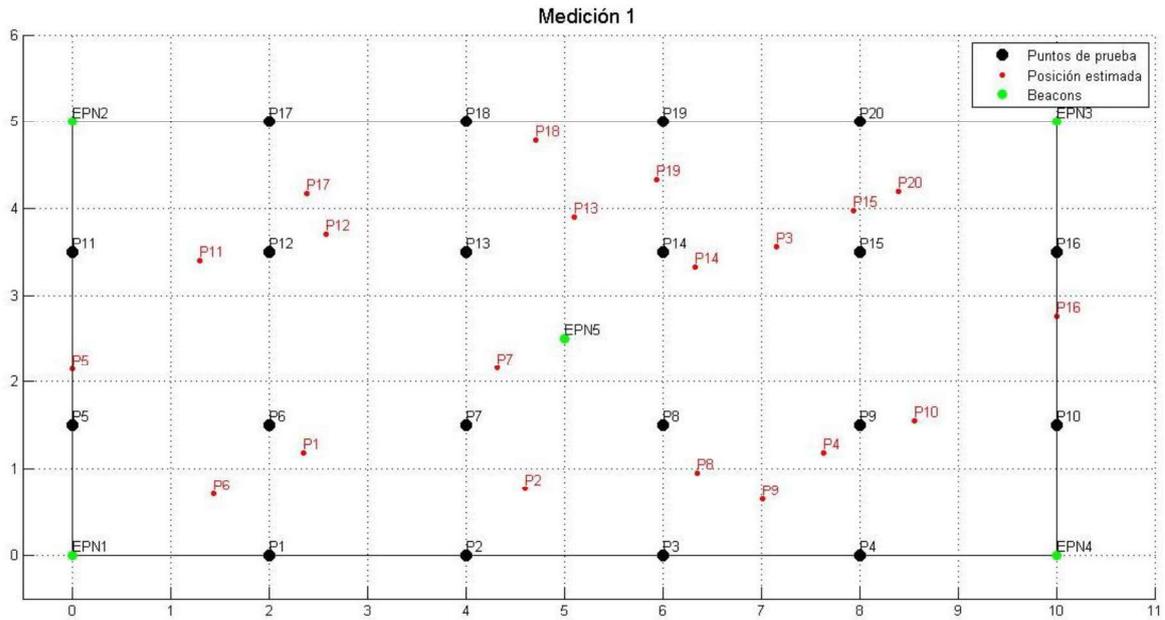


Figura 3.31. Ubicaciones estimadas Prueba 3 (*Indoor*)

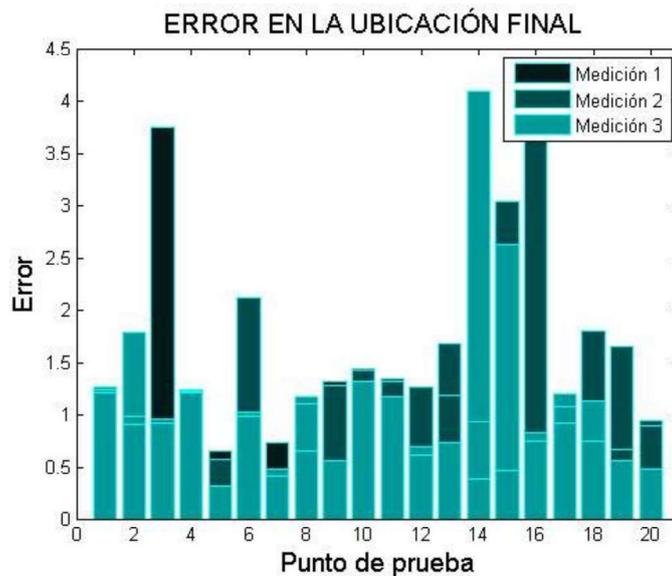


Figura 3.32. Error en la ubicación Prueba 3 (*Indoor*)

Con los resultados obtenidos en esta prueba, se verifica que la separación entre los *Beacons* y el dispositivo central es uno de los principales factores que inciden en la presencia de errores al estimar la ubicación. Sin embargo, se observa en la Tabla 3.5 que el error medio disminuye en comparación con las dos pruebas anteriores; además, el número de ubicaciones que se consiguen con un error menor a dos metros representan el 87% de las estimaciones realizadas. Los niveles de precisión alcanzados con esta prueba, se consiguen gracias a que las distancias entre los tres dispositivos más cercanos al

central, no supera los 7 metros, manteniéndose así dentro del rango en el cual el modelo de propagación calcula las distancias con una mejor precisión.

Tabla 3.5. Error y porcentajes de ubicación dentro de rangos de precisión para cada medición (Prueba 3 *indoor*)

	Medición 1	Medición 2	Medición 3
Error medio	1,04	1,45	1,18
Error máximo	3,74	4,25	4,08
Error mínimo	0,38	0,41	0,31
Porcentaje de ubicación dentro del rango de error			
< 1m	65	30	45
1m - 2m	30	55	45
2m<	5	15	10

Cabe destacar que aún existen mediciones que superan los dos metros de error, esto se debe que los cinco *Beacons* están transmitiendo sobre los tres canales disponibles, de tal manera que la posibilidad que las señales interfieran entre si aumenta. Por otro lado, al aumentar en número de *Beacons*, las áreas cubiertas por tres dispositivos se hacen más pequeñas y por lo tanto la precisión en su interior aumenta.

Finalmente, en la Tabla 3.6 se resumen los resultados obtenidos en las tres pruebas realizadas.

Tabla 3.6. Comparación de resultados alcanzados en las tres pruebas

	Prueba 1	Prueba 2	Prueba 3
Error medio	1,71	1,86	1,22
Error máximo	5,65	4,53	4,02
Error mínimo	0,21	0,24	0,37
Porcentaje de ubicación dentro del rango de error			
< 1m	33,33	31,67	46,67
1m - 2m	41,67	30,00	43,33
2m<	25,00	38,33	10,00

Con los datos expuestos en la tabla anterior, se puede establecer que la mejor configuración de la red inalámbrica se consigue con cinco dispositivos configurados como *Beacon*. Por otro lado, el error medio obtenido y el porcentaje de ubicaciones con error menos a dos metros, cumplen con los objetivos planteados inicialmente. A pesar de esto, no se puede concluir que el presente proyecto es exitoso, debido a la presencia de errores

máximos de entre 4 y 5 metros para el 25% de las posiciones estimadas. En general la respuesta del algoritmo como se puede observar en los resultados obtenidos en los escenarios de prueba de simulación y ambiente de prueba, es buena, mostrando que gran parte de los errores cometidos son producto de los efectos del ambiente interior sobre las señales transmitidas, además de las características de la red prototipo implementada. Esto abre las puertas a expandir la investigación alrededor de la tecnología *Bluetooth* 4.0 aplicada a sistemas de localización en interiores.

A continuación, se analizan algunos aspectos característicos de un sistema de comunicaciones, enfocados al algoritmo y la red implementados:

- **Escalabilidad:** Como se evidencia en los resultados obtenidos en los escenarios de prueba en los ambientes interior y de prueba, el algoritmo es capaz de responder y acoplarse de manera efectiva a la variación del número de *Beacons* disponibles, inclusive presentando errores menores al aumentar el número de transmisores. En consecuencia, al permitir el incremento de dispositivos (de tres a cinco) sin la necesidad de realizar modificaciones trascendentales al algoritmo y mejorando la precisión en la ubicación, se comprueba que el algoritmo implementado es escalable.
- **Costo:** Los costos por implementación y mantenimiento de un sistema de localización, están ligados directamente al número de equipos que se instalen, es decir a mayor número de equipos, más cara será la solución. Con estas consideraciones, al conocer que el algoritmo presenta mejor precisión con un mayor número de dispositivos LE instalados, se puede asegurar que para alcanzar mayor precisión se debe incurrir en más gastos. Sin embargo, este proyecto se desarrolló en base a una tecnología más económica con respecto a otras opciones existentes, garantizando de esta manera mantener una relación costo precisión adecuada.
- **Consumo de energía:** Una de las características más sobresalientes de la tecnología *Bluetooth Low Energy* es su bajo consumo de energía, de esta manera, el algoritmo implementado aprovecha al máximo las características de bajo consumo que se pueden encontrar en los módulos LE. Es así que el algoritmo es capaz de responder manteniendo los niveles de precisión encontrados, inclusive con configuraciones que minimicen al máximo el consumo de energía.

4. CONCLUSIONES

4.1 CONCLUSIONES

- Se ha logrado implementar un algoritmo de localización basado en trilateración y tecnología *Bluetooth Low Energy* para la ubicación de objetos o personas en ambientes interiores, haciendo uso de trilateración circular y esférica, y técnicas de medición de potencia de las señales. Además, la información obtenida de la especificación *Bluetooth 4.0*, el protocolo *iBeacon* y el datasheet del módulo *Bluetooth* HM-10 permitieron la configuración e implementación de la red inalámbrica BLE.
- El conjunto implementado red-algoritmo puede convertirse en una opción aplicable a sistemas de monitoreo de pacientes en hospitales o sistemas de evacuación en casos de emergencia, gracias a su facilidad y bajo costo de implementación, su independencia de la red eléctrica y de otras redes de telecomunicaciones, y la extendida disponibilidad de esta tecnología en teléfonos inteligentes. Además, es importante mencionar que el algoritmo se puede migrar hacia una aplicación móvil y de esta manera desarrollar soluciones comerciales y distribuibles.
- Considerando un uso eficiente de energía por parte de los dispositivos, el bajo consumo que presenta la tecnología *Bluetooth LE* hace que sea una solución óptima para desarrollar sistemas de ubicación en interiores, pues se ha comprobado experimentalmente que al emplear este tipo de módulos el tiempo de duración promedio de la carga de pilas comunes triple A supera los 30 días de tal manera que se reducen los costos por despliegue y mantenimiento de la red.
- Durante el proceso de calibración del sistema para determinar el valor adecuado de las constantes que definen el modelo de propagación para convertir las mediciones de potencia en distancia, son factores limitantes el tipo y la orientación de las antenas, ya que debido a la no uniformidad del lóbulo de radiación que presentan, se incrementa la inestabilidad al medir la potencia; en consecuencia, el cálculo de la distancia y la estimación de la posición disminuyen sus niveles de precisión. Por otra parte, se debe considerar que por efectos de construcción física los dispositivos siempre van a presentar diferencias, generando variaciones en los niveles de potencia que transmiten o reciben y por lo tanto limitando el desempeño de la calibración estática implementada.

- Las técnicas de trilateración por medición de potencia dependen directamente del grado de precisión con el cual se calcule la distancia, ya que mientras más cercana sea la distancia estimada a la real, menor será el error al hallar la ubicación del dispositivo móvil. En el caso de ambientes tan cambiantes como son los interiores, una manera de contrarrestar estas variaciones y mejorar los resultados es el uso de métodos de estimación adaptativa de la distancia que determinen continuamente el valor de referencia $RSSI_0$ y del exponente de atenuación n .
- La selección de la posición inicial P_0 es un factor determinante en la estimación final de la ubicación del dispositivo móvil, puesto que una mala elección del mismo puede generar matrices singulares y limitar el cálculo de la matriz inversa del Jacobiano de la función de error que se considera en el algoritmo de Gauss-Newton, dando resultados completamente erróneos en la posición final del objetivo. De la misma manera, se pueden generar este tipo de errores al ubicar los *Beacons* en puntos de referencia que tienden a estar alineados o muy cercanos entre sí.
- Al incrementarse la separación entre los *Beacons* y el dispositivo central, el cálculo de la distancia se vuelve más inestable y de esta manera también lo hace la estimación de la posición, incrementando los errores en la ubicación final obtenida tras la ejecución del algoritmo.
- La respuesta del algoritmo dentro de un ambiente interior real se puede considerar buena, sin embargo se presenta un porcentaje elevado de estimaciones con errores superiores a dos metros. Razón por la cual se concluye que el desarrollo del presente proyecto no fue 100% exitoso.

4.2 RECOMENDACIONES

- La solución propuesta hace uso de las antenas que están integradas con los módulos. Se recomienda hacer las modificaciones necesarias para adicionar antenas omnidireccionales y de esta manera disminuir los problemas generados por la variabilidad que presentan los lóbulos de radiación de las antenas.
- Las mediciones de potencia a través del indicador RSSI se ven afectadas en gran medida por las múltiples trayectorias que se encuentran en los ambientes interiores. Un método que se sobrepone al problema de las trayectorias múltiples es calcular la distancia basado en tiempos de vuelo. De esta manera, considerando que la actualidad se ha desarrollado dispositivos BLE que permiten la medición de estos

tiempos, se recomienda la implementación del algoritmo propuesto, alrededor de estos nuevos dispositivos *Bluetooth*.

- Con el objetivo de mejorar la precisión del algoritmo, se recomienda el desarrollo de un proceso dinámico para determinar las constantes del modelo de propagación, con el fin de disminuir los efectos generados por las variaciones intrínsecas de los ambientes internos.
- Una de las limitantes que afectan al algoritmo es la ubicación de los *Beacons* de referencia, por tanto, se recomienda realizar un estudio sobre las ubicaciones óptimas de los dispositivos para sistemas de localización en interiores.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] D. Barba, A. Moreton Fernandez, H. Del Campo Pando, S. Garcia Villanueva, J. March y D. Ferraris, «Uso de Bluetooth de baja energía en aplicaciones de localización indoor.,» de *Jornadas SARTECO*, Málaga, 2017.
- [2] X. Zhao, L. Ruan, L. Zhang, Y. Long y F. Cheng, «An Analysis of the Optimal Placement of Beacon in Bluetooth-INS Indoor Localization,» de *Adjunct Proceedings of the 14th International Conference on Location Based Services*, Zurich, ETH Zurich, 2018, pp. 50-55.
- [3] C. Loja Muzha, Desarrollo de una solución prototipo que permita determinar salidas de emergencia dentro de un ambiente cerrado, a través de dispositivos móviles Android, utilizando localización mediante WIFI y establecimiento de rutas óptimas, Quito: Escuela Politécnica Nacional, 2016.
- [4] N. Cadena Ramírez y A. Olivo Solis, Diseño e Implementación de un Sistema Para la Localización De Personas En Ambientes Cerrados En Base A Tecnología Bluetooth, Quito: Escuela Politécnica Nacional, 2010.
- [5] A. Carcavilla Sanz, Sistemas de posicionamiento basados en WiFi, Barcelona: Universidad Politécnica de Cataluña, 2006.
- [6] A. Agudo de Carlos, Sistema de Localización de Dispositivos Móviles Basada en Wireless LAN, Madrid: Universidad Carlos III de Madrid, 2009.
- [7] C. Dobarco Alvarez, Sistema De Posicionamiento Para Superficies Comerciales Mediante Dispositivos Móviles, Madrid, España: Universidad Pontificia ICAI, 2013.
- [8] F. D. Ruiz Pereda, Localización y Autocalibrado con Sistemas de Posicionamiento Local Ultrasónicos, Madrid, España: Universidad de Alcalá, 2011.
- [9] P. A. Romero Macas, Análisis de la posición y tiempo de detección para dispositivos móviles en un sistema de posicionamiento local en espacios interiores basados en el estándar IEEE 802.11 mediante la técnica Fingerprints y el filtro de Kalman, para aplicaciones de robótica, Sangolquí, Ecuador: ESPE, 2017.

- [10] INEC, «ecuadorencifras,» INEC, [En línea]. Available: <http://www.ecuadorencifras.gob.ec/documentos/web-inec/Infografias-INEC/2017/smartphones2.jpg>. [Último acceso: 01 03 2019].
- [11] M. Hazas, J. Scott y J. Krumm, «Location-aware computing comes of age,» *Computer*, vol. 37, nº 2, pp. 95-97, Febrero 2004.
- [12] L. Dudkowski, «IR Location Tracking System,» Computer Engineering, Columbia University, 2003. [En línea]. Available: http://www.cs.columbia.edu/~hgs/research/projects/stamptag/Final/clean_index.html.
- [13] Y. Jiang, R. Song y M. Yuan, «Improvement of Ultrasonic Distance Measuring,» de *ITM Web of Conferences*, 2018.
- [14] R. Sánchez Vítores, «ILS (Indoor Location Systems) Sistemas de Localización en Interiores,» *Bit*, nº 148, pp. 57-59, 2005.
- [15] P. Ludeña, K. Rohoden, C. Palacios y M. Rohoden, «Yanapay: Sistema de evacuación basado en tecnología RFID y dispositivos Android,» *Ingenius*, 2016.
- [16] V. Shinde, R. Panchal y J. Panchal, «Zigbee Based Indoor Location Tracking System,» *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 5, 2016.
- [17] F. Barba Barba, Estudio de algoritmos de localización en interiores, para tecnologías móviles de última generación, Madrid: Universidad Politécnica de Madrid, 2012.
- [18] S. Banerji y R. Singha Chowdhury, «On IEEE 802.11: Wireless Lan Technology,» *International Journal of Mobile Network Communications & Telematics*, vol. 3, 2013.
- [19] W. Yang, C. Xiu, J. Ye, Z. Lin, H. Wei, D. Yan y D. Yang, «LSS-RM: Using Multi-Mounted Devices to Construct a Lightweight Site-Survey Radio Map for WiFi Positioning,» *Micromachines*, 2018.
- [20] R. Campos, «POSITIONING IN BLUETOOTH AND UWB NETWORKS,» *Cadernos do IME - Série Informática*, vol. 38, pp. 35-54, 2017.
- [21] F. Gonzalez-Castano y J. Garcia Reinoso, «Survivable Bluetooth location networks,» *IEEE International Conference*, vol. 2, 2003.

- [22] GISGeography, «Trilateration vs Triangulation – How GPS Receivers Work,» GISGeography, [En línea]. Available: <https://gisgeography.com/trilateration-triangulation-gps/>. [Último acceso: 1 Abril 2019].
- [23] V. Gao, «Bluetooth SIG,» 21 Septiembre 2015. [En línea]. Available: <https://www.bluetooth.com/blog/proximity-and-rssi/>. [Último acceso: 29 Marzo 2019].
- [24] N. Vara, G. Poletto, M. Cacéres y A. Busso, «Cálculo de Distancia entre los Nodos de una Red Inalámbrica ZigBee en Función del Parámetro RSSI,» *Extensionismo Innovación y Transferencia Tecnológica*, vol. 2, pp. 1-6, 2015.
- [25] Alcatel-Lucent, Balizas BLE Alcatel-Lucent OmniAccess StellarBaliza Bluetooth 4.0 de baja energía con tecnología iBeacon, 2018.
- [26] ShenzhenRadiolandTechnology CO.,LTD., Bluetooth Smart Module BLE4.0 NRF51822-Beacon Datasheet, 2018.
- [27] NORDIC Semiconductor, nRF51822 Bluetooth® Smart Beacon Kit User Guide v1.1, 2018.
- [28] F. Dalkılıç, U. Çabuk, E. Arıkan y A. Gürkan, An Analysis of the Positioning Accuracy of iBeacon Technology in Indoor Environments, 2017.
- [29] D. Grzechca, P. Pelczar y L. Chruszczyk, «Analysis of Object Location Accuracy for iBeacon Technology based on the RSSI Path Loss Model and Fingerprint Map,» *INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS*, vol. 62, nº 4, pp. 371-378, 2016.
- [30] C. A. Balanis, Antenna Theory Analysis and Desing, Tercera ed., Wiley-Interscience, 2005, p. 94.
- [31] O. S. Oguejiofor, V. N. Okorogu, A. Adewale y B. O. Osuesu, «Outdoor Localization System Using RSSI Measurement of Wireless Sensor Network,» *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 2, nº 2, Enero 2013.
- [32] E. Lau, S. C. Lee, B. G. Lee y W. Y. Chung, «ENHANCED RSSI BASED HIGH ACCURACY REAL-TIME USER LOCATION TRACKING SYSTEM FOR INDOOR AND OUTDOOR ENVIRONMENTS,» *INTERNATIONAL JOURNAL ON SMART SENSING AND INTELLIGENT SYSTEMS*, vol. 1, nº 2, 2008.

- [33] SIG Bluetooth, «System Core System Package [Low Energy Controller volume],» de *Specification of the Bluetooth*, 2010, pp. 2165-2302.
- [34] «Bluetooth Low Energy Controller Low IF GFSK Modem,» 2013.
- [35] C. d. T. Arkaitz Arregui, Beacons BLE (Bluetooth Low Energy) en el sector turístico, control de afluencia y servicios de valor añadido, Barcelona: Universidad Oberta de Cataluña, 2016.
- [36] GSMA, A Guide to Bluetooth Beacons, 2014.
- [37] Apple, Getting Started with iBeacon, 2014.
- [38] D. Helms, AltBeacon Protocol Specification v1.0, 2015.
- [39] JN Huamao Technology Co., Ltd., HM Bluetooth module datasheet, 2018.
- [40] Texas Instruments, CC2540 and CC2541 Bluetooth® low energy Reference Guide, 2010.
- [41] JN Huamao Technology Co., Ltd., HM-10/11/15 firmware upgrade instructions, 2018.
- [42] Consejo Metropolitano de Quito, Normas de Arquitectura y Urbanismo, Ordenanza 3457, Quito, 2013.
- [43] F. Thomas y L. Ros, «Revisiting Trilateration for Robot Localization,» *IEEE TRANSACTIONS ON ROBOTICS*, vol. 21, nº 1, pp. 93-101, 2005.
- [44] A. Basiri, E. Lohan, P. Figueiredo, P. Peltola, C. Hill y T. Moore, «Overview of positioning technologies from fitness-to-purpose point of view,» de *International Conference on Localization and GNSS*, IEEE, 2014.
- [45] J. Caspi y J. Guaman, «Desarrollo de aplicaciones basadas en ubicación para espacios interiores utilizando tecnologías de localización,» 2017. [En línea]. Available: <http://repositorio.espe.edu.ec/handle/21000/13574>. [Último acceso: 14 Junio 2018].
- [46] A. Al-Odienat y A. Al-Mbaideen, «Optimal Length Determination of the Moving Average Filter for Power System Applications,» *International Journal of Innovative Computing, Information and control*, vol. 11, nº 2, pp. 691-705, 2015.

ANEXOS

ANEXO A. Código del algoritmo implementado

ANEXO B. Código de la función de trilateración de Gauss Newton

ANEXO C. Conjunto de datos obtenidos en las mediciones de cada prueba

ANEXO D. Manual de usuario

ANEXO E. Archivos y códigos de la interfaz grafica

ANEXO A

Código del algoritmo de localización basado en trilateración y tecnología *Bluetooth Low Energy* para la ubicación de objetos o personas en ambientes interiores.

```
%*****
**
%           ESCUELA POLITÉCNICA NACIONAL
%           FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA
%IMPLEMENTACIÓN DE UN ALGORITMO DE LOCALIZACIÓN EN INTERIORES PARA
RESOLVER
%EL PROBLEMA DE UBICACIÓN DE OBJETOS USANDO TECNOLOGÍA BLUETOOTH LOW
ENERGY
%TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
%ELECTRÓNICA Y TELECOMUNICACIONES
%AUTOR: WILLIAM PATRICIO MOYA VELASCO
%DIRECTOR: ING. RAMIRO EDUARDO MOREJON TOBAR
%*****
**
%*****
**
%Algoritmo basado en las técnicas de trilateración circular y esférica,
%implementado dentro de una red inalámbrica Bluetooth Low Energy; permite
%encontrar la ubicación en dos dimensiones de un dispositivo móvil
%(BLE central) dentro de un ambiente interior
%*****
**
%% Inicialización
clear           %Elimina variables del workspace usadas en procesos
anteriores
close all      %Cierra cualquier ventana emergente
%% Inicialización de la comunicación serial
disp('Inicialización de la comunicación serial...');
disp('');
delete(instrfind);           %Cierra cualquier puerto y
comunicacion previas
portFlag=0;connFlag=0;
while portFlag==0 || connFlag==0
    serialPorts = instrhwinfo('serial');           %Busca informacion de
puertos diponibles
    if isempty(serialPorts.SerialPorts)
        disp('Ningun puerto disponible')
        %errorhdlg('Ningun puerto disponible');
    else
        fprintf('Puertos: \n');
        disp(serialPorts.SerialPorts)
        serPort = input('Ingrese un puerto: ','s');
        for i=1:length(serialPorts.SerialPorts)
            if strcmp(serPort,serialPorts.SerialPorts(i))
                portFlag=1;
                break
            else
                portFlag=0;
            end
        end
    end
    if portFlag==1
```

```

serConn = serial(serPort, 'BaudRate', 115200, ...
'DataBits', 8, 'StopBits', 1, 'Terminator', 'CR/LF');
try
    fopen(serConn);
    disp('Conexión exitosa')
    connFlag=1;
catch e
    errorDlg(e.message);
    fclose(serConn);
    delete(instrfind);
    connFlag=0;
end
else
    disp('Puerto invalido')
    errorDlg('Select a valid COM port');
end
end
end
%% Búsqueda de beacons activos
disp('Búsqueda de beacons activos...');
disp('');
N_beacons=0;
while N_beacons<3
    N_beacons=0;
    beacons=[0 0 0 0 0];
    fprintf(serConn, '%s\n', 'AT+DISI?'); %write data
    tic;
    while toc<=4
        end
        i=0;
        readData={};
        while serConn.BytesAvailable~=0
            i=i+1;
            readData{i}=fscanf(serConn);
            serConn.BytesAvailable;
            if length(readData{i})>32
                beacon_id=readData{i}(1, 62:73);
                switch beacon_id
                    case '44EAD82A8ADE'
                        beacons(1)=1;
                    case '44EAD82AACC8'
                        beacons(2)=1;
                    case 'D436395E19E6'
                        beacons(3)=1;
                    case '44EAD82A69C5'
                        beacons(4)=1;
                    case '44EAD82A93C5'
                        beacons(5)=1;
                end
            end
            N_beacons=sum(beacons);
        end
        if N_beacons<=2
            disp('Escaneando...Beacons insuficientes');
            disp('');
        end
    end
end
%% Procesamiento de señales
disp('Procesamiento de señales...');
disp('');

```

```

% Adquisición de datos
N_beacons_inicial=N_beacons;
mediciones=zeros(10,5);
indicador=0;
for n_medicion=1:10
    while N_beacons<3 || indicador==0
        beacons=[0 0 0 0 0];
        fprintf(serConn,'%s\n','AT+DISI?'); %write data
        tic;
        while toc<=4
            end
            i=0;
            readData={};
            while serConn.BytesAvailable~=0
                i=i+1;
                readData{i}=fscanf(serConn);
                serConn.BytesAvailable;
                if length(readData{i})>32
                    beacon_id=readData{i}(1,62:73);
                    switch beacon_id
                        case '44EAD82A8ADE'
                            beacons(1)=1;
                        case '44EAD82AACC8'
                            beacons(2)=1;
                        case 'D436395E19E6'
                            beacons(3)=1;
                        case '44EAD82A69C5'
                            beacons(4)=1;
                        case '44EAD82A93C5'
                            beacons(5)=1;
                    end
                end
            end
            N_beacons=sum(beacons);
            if N_beacons<=2
                disp('Adquiriendo datos... Beacons insuficientes');
            elseif N_beacons==N_beacons_inicial
                indicador=1;
                disp('Adquiriendo datos...');
            end
            if N_beacons>=3
                disp('Adquiriendo datos...');
            end
        end
        for dato=1:i
            if length(readData{dato})>32
                beacon_id=readData{dato}(1,62:73);
                switch beacon_id
                    case '44EAD82A8ADE'
mediciones(n_medicion,1)=str2num(readData{dato}(1,75:78));
                    beacons(1)=1;
                    case '44EAD82AACC8'
mediciones(n_medicion,2)=str2num(readData{dato}(1,75:78));
                    beacons(2)=1;
                    case 'D436395E19E6'
mediciones(n_medicion,3)=str2num(readData{dato}(1,75:78));
                    beacons(3)=1;
                end
            end
        end
    end
end

```

```

        case '44EAD82A69C5'
mediciones(n_medicion,4)=str2num(readData{dato}(1,75:78));
        beacons(4)=1;
        case '44EAD82A93C5'
mediciones(n_medicion,5)=str2num(readData{dato}(1,75:78));
        beacons(5)=1;
        end
    end
end
indicador=0;
end
% Filtrado
%Descarte de datos fuera de rango
for i=1:5
    for j=2:10
        if mediciones(j,i)<=-74
            mediciones(j,i)=mediciones(j-1,i);
        end
    end
    if mediciones(1,i)<=-74
        mediciones(1,i)=mediciones(10,i);
    end
end
promedio=mean(mediciones);
for i=1:5
    if promedio(i)<=-75
        N_beacons=N_beacons-1;
    end
end
end
if N_beacons>=3
    for i=1:5
        for j = 2:9
            mediciones(j,i) = (mediciones(j-1,i) + mediciones(j,i) +
mediciones(j+1,i))/3;
        end
        mediciones(1,i)=mediciones(2,i);
        mediciones(10,i)=mediciones(9,i);
    end
    distancia=99*ones(1,5);
    rssi_final=mean(mediciones);
    rssi_0=-50.10;
    n=2.2;
    for i=1:5
        if (rssi_final(i)<=-40 && rssi_final(i)>-50.10)
            distancia(i)=1;
        elseif (rssi_final(i)<=-50.10 && rssi_final(i)>-76)
            distancia(i)=10^((rssi_0-rssi_final(i))/(10*n));
        end
    end
end
dist=distancia;
[distancia,beacon]=sort(distancia);
%Distancia y referencia ordedas de mayor a menor
referencias=[0 0 1.5;...
            0 5 1.5;...
            10 5 1.5;...
            10 0 1.5;...
            5 2.5 1.5];

```

```

    ref_graf=referencias;
    datos_trilateracion=zeros(N_beacons,4);
    for i=1:N_beacons
        datos_trilateracion(i,:)=[distancia(i) referencias(beacon(i),:)]];
    end
else
    disp('Beacon activos insuficientes...');
end

%% Trilateración básica (Circular)
disp('Trilateración básica...');
disp('');
radio_2=datos_trilateracion(:,1).^2; %Radios al cuadrado
syms x y %Variables x y para resolver las ec
ecuacion_1=(x-datos_trilateracion(1,2))^2+...
    (y-datos_trilateracion(1,3))^2==radio_2(1);
ecuacion_2=(x-datos_trilateracion(2,2))^2+...
    (y-datos_trilateracion(2,3))^2==radio_2(2);
cruce=solve(ecuacion_1,ecuacion_2);
cruce_x=double(cruce.x); %Valores en x donde intersecan las
circunferencias
cruce_y=double(cruce.y); %Valores en y donde intersecan las
circunferencias
if isreal(cruce_x) && isreal(cruce_y)
    %Ecuacion de la recta
    if (cruce_y(2,1)-cruce_y(1,1))==0
        recta=y==cruce_y(2,1);
    elseif (cruce_x(2,1)-cruce_x(1,1))==0
        recta=x==cruce_x(2,1);
    else
        recta=(x-cruce_x(1,1))/(cruce_x(2,1)-cruce_x(1,1))==...
            (y-cruce_y(1,1))/(cruce_y(2,1)-cruce_y(1,1));
    end
    %Ecuacion tercera circunferencia
    ecuacion_3=(x-datos_trilateracion(3,2))^2+...
        (y-datos_trilateracion(3,3))^2==radio_2(3);
    S=solve(recta,ecuacion_3);
    ble_x=double(S.x);
    ble_y=double(S.y);
    if isreal(ble_x) && isreal(ble_y)
        d_1=sqrt((ble_x(1,1)-cruce_x(1,1))^2+((ble_y(1,1)-
cruce_y(1,1))^2));
        d_2=sqrt((ble_x(2,1)-cruce_x(1,1))^2+((ble_y(2,1)-
cruce_y(1,1))^2));
        if d_1<d_2
            p_0=[ble_x(1,1) ble_y(1,1) -1];
        else
            p_0=[ble_x(2,1) ble_y(2,1) -1];
        end
    else
        if cruce_x(1,1)>=0 && cruce_y(1,1)>=0
            p_0=[cruce_x(1,1) cruce_y(1,1) -1];
        elseif cruce_x(2,1)>=0 && cruce_y(2,1)>=0
            p_0=[cruce_x(2,1) cruce_y(2,1) -1];
        end
    end
    end
    trilatFlag=1;
else
    trilatFlag=1;
    p_0=[real(cruce_x(1,1)) real(cruce_y(1,1)) -1];
end

```

```

    posicion_final=p_0;
    disp('');
end
%% Algoritmo de trilateración Gauss Newton
disp('Algoritmo de trilateración Gauss Newton...');
disp('');
if trilatFlag==1
    %BLE_F=gaussnewton(referencias,distancias,posicion inicial,N de
Beacons activos);
    referencias=datos_trilateracion(:,2:4);
    distancias=sqrt(radio_2+1);
    posicion_final=gaussnewton(referencias,distancias,p_0,N_beacons);
if posicion_final(1)>20 || posicion_final(2)>20 || posicion_final(2)<-20
|| posicion_final(2)<-20
    posicion_final=abs(p_0);
end
if isnan(posicion_final)
    posicion_final=abs(p_0);
end
for i=1:3
    if posicion_final(i)<0
        posicion_final(i)=0;
    end
end
trilatFlag=1;
end
%% Presentación de Resultados
if trilatFlag==1
disp('Presentación de Resultados...');
disp('');
figure(1)
hold on
grid on

%Area de prueba
rectangle('Position',[0 0 10 5],'LineWidth',1);

%Ubicación de los beacons referenciales
labels = {'EPN1','EPN2','EPN3','EPN4','EPN5'};
referencias=[0 0 1.5;0 5 1.5;10 5 1.5;10 0 1.5;5 2.5 1.5];
colores=['r' 'y' 'g' 'b' 'c'];
for i=1:5
    plot(referencias(i,1),referencias(i,2),...
        'Marker','o',...
        'MarkerFaceColor','b',...
        'MarkerEdgeColor','b');
    text(referencias(i,1),referencias(i,2),labels(i),...
        'FontSize',12,'Color','b',...
        'VerticalAlignment','bottom');
end
%Ubicación de los circulos con distancias mas cortas al dispositivo
central
for i=1:3
    viscircles([datos_trilateracion(i,2) datos_trilateracion(i,3)],...
        datos_trilateracion(i,1),...
        'LineStyle','- ',...
        'EdgeColor',colores(i),...
        'LineWidth',0.25);
end
end
%Puntos de cruce entre las circunferencias mas cercanas

```

```

plot(real(cruce_x),real(cruce_y),...
      'black','Marker','*',...
      'MarkerFaceColor','black',...
      'MarkerEdgeColor','black');
%Ubicacion generada por trilateracion basica
plot(p_0(1),p_0(2),...
      'w','Marker','o',...
      'MarkerFaceColor','g',...
      'MarkerEdgeColor','g','MarkerSize',8)

%Ubicacion generada por trilateracion Gauss Newton
plot(posicion_final(1),posicion_final(2),...
      'w','Marker','o',...
      'MarkerFaceColor','r',...
      'MarkerEdgeColor','r','MarkerSize',10)
legend('Posición inicial (Po)','Posición Estimada');

end

```

ANEXO B

Código del algoritmo de la función de trilateración Gauss Newton.

```
function [BLE_GN]=gaussnewton(referencias,distancias,BLE_i,N)

for interacion=1:100                                % # de
interaciones max                                    % # de
    for i=1:N
        di(i)=sqrt(((referencias(i,1)-BLE_i(1))^2)+... %Distancia
estimada                                             % # de
                ((referencias(i,2)-BLE_i(2))^2)+...
                ((referencias(i,3)-BLE_i(3))^2));
        A(i,1)=(BLE_i(1)-referencias(i,1))/di(i);    %Matriz de
derivadas parciales
        A(i,2)=(BLE_i(2)-referencias(i,2))/di(i);
        A(i,3)=(BLE_i(3)-referencias(i,3))/di(i);
        B(i,1)=di(i)-distancias(i);                 %Matriz de error
    end

    AT=A';                                           % A' = A^T
    ATA=AT*A;                                         % A' * A
    ATA_1=ATA^-1;                                     % (A' * A)^-1
    ATB=AT*B;                                         % A'*B
    %Calculo de la variacion de posicion en x , y , z
    DX=ATA_1*ATB;                                     % (A' * A)^-1 * A'*B = Delta(x,y,z)
    %Nueva posicion estimada
    BLE_i=BLE_i-DX';                                  % (X Y Z)n+1=(Xn Yn Zn)+Delta(X,Y,Z)

    if abs(DX)<0.2*1.0e-15 % Tolerancia de variacion en la posicion
        break
    end
end
BLE_GN=BLE_i;                                         %Ubicacion estimada del movil
```

ANEXO C

Anexo digital.

ANEXO D

Anexo digital.

ANEXO E

Anexo digital.

ORDEN DE EMPASTADO