



# Universidad Politécnica de Cartagena

**Trabajo Fin de Master**

## **MASTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN**

Diseño y desarrollo de un prototipo de localización de interiores basado en beacons Bluetooth Low Energy



AUTOR: Pedro Javier García Paterna

DIRECTOR: Alejandro Martínez Sala

Diciembre/2020

# *Agradecimientos*

*Agradecer, en primer lugar, a la Universidad Politécnica de Cartagena y a la Escuela Técnica Superior de Ingeniería de Telecomunicación por haberme dotado de los conocimientos que esta titulación conlleva y haberme acogido como alumno por más de 6 años.*

*A mi director de proyecto y amigo, D. Alejandro Martínez Sala, por el traspaso de conocimientos y la confianza depositada en mi desde el inicio del proyecto.*

*A mis compañeros de universidad que me han acompañado durante toda esta etapa, y en especial a María García por hacerme más divertido el camino.*

*A la Teleco LAN Party, por haberme dotado de aspectos que no se aprenden en las aulas y que me han servido de mucho en el sector empresarial.*

*A mi familia, especialmente a mis padres, por confiar en mi día tras días y por ser un ejemplo de lucha y superación.*

*A mi pareja Naomi Saura, por enseñarme a disfrutar de la vida y celebrar mis logros como si fuesen suyos.*

*Gracias a todos.*





<b>Autor</b>	Pedro Javier García Paterna
<b>E-mail del autor</b>	pedrolo.gpaterna@gmail.com
<b>Director</b>	Alejandro Martínez Sala
<b>E-mail del director</b>	alejandros.martinez@upct.es
<b>Título del TFM</b>	Prototipo de localización de interiores basado en beacons Bluetooth Low Energy
<b>Resumen</b>	Los objetivos del presente proyecto son diseñar e implementar un sistema de localización basado en el despliegue de beacons Bluetooth Low Energy (BLE) usando la técnica de fingerprint y algoritmos de machine learning basado en aprendizaje supervisado para la estimación de la zona (a nivel de habitación) donde se encuentra una persona. En la técnica de fingerprint se generan vectores característicos con los patrones de señal de los beacons BLE muestreados por zona. El problema de localización se formula como un problema de clasificación; cada zona se considera una clase y dado un fingerprint instantáneo (vector característico de señal muestreado en un intervalo de tiempo) se busca estimar la clase (habitación) a la que pertenece dicho fingerprint. Como algoritmo de clasificación se emplea el kNN (k-Nearest Neighbours).
<b>Titulación</b>	Master Universitario en Ingeniería de Telecomunicación.
<b>Departamento</b>	Tecnología de la Información y las Comunicaciones.
<b>Fecha de presentación</b>	Diciembre 2020

## Resumen

Los sistemas de localización en IoT son de interés para el desarrollo de servicios de localización de objetos y personas en entornos de interior (hospital, supermercado, aeropuerto, universidad, etc.). En los entornos de interior no funciona la tecnología GPS por lo que se necesitan otras tecnologías. En particular la tecnología Bluetooth Low Energy (BLE) 4.0 es una evolución del estándar Bluetooth Classic diseñado para el IoT, con un enfoque en el bajo consumo, reducido coste y facilidad de instalación y despliegue. La tecnología BLE tiene un enorme potencial para el desarrollo de servicios de localización en IoT. Los beacons BLE son dispositivos de bajo coste que emiten a intervalos regulares mensajes de advertising con su ID. Un escáner BLE es un dispositivo IoT que monitoriza ciertos canales a 2,4GHz, escucha los ID de los beacons en cobertura y su nivel de señal. Una arquitectura de localización de interés consiste en desplegar beacons BLE en un entorno y un dispositivo IoT monitoriza los beacons en cobertura y, aplicando técnicas de machine learning, se infiere la ubicación del dispositivo. Existen servicios de interés donde el dato de la ubicación de una persona en un área acotada a nivel de habitación (*zone level*), es suficiente y útil y no se requiere una resolución de a posición en coordenadas (x,y).

Los objetivos del presente proyecto son diseñar e implementar un sistema de localización basado en el despliegue de beacons Bluetooth Low Energy (BLE) usando la técnica de fingerprint y algoritmos de machine learning basado en aprendizaje supervisado para la estimación de la zona (a nivel de habitación) donde se encuentra una persona. En la técnica de fingerprint se generan vectores característicos con los patrones de señal de los beacons BLE muestreados por zona. El problema de localización se formula como un problema de clasificación; cada zona se considera una clase y dado un fingerprint instantáneo (vector característico de señal muestreado en un intervalo de tiempo) se busca estimar la clase (habitación) a la que pertenece dicho fingerprint. Como algoritmo de clasificación se emplea el kNN (*k-Nearest Neighbours*).

También se ha estudiado la capa física RF de BLE y los factores que afectan a la localización zonal y se han definido parámetros y métricas para medir y que sirven para desplegar los beacons en puntos adecuados y garantizar que se puede localizar en todas las zonas. Finalmente se ha hecho una campaña de medidas sistemática en el entorno real de una casa y evaluado el desempeño y el error de la estimación zonal.

## Índice

Capítulo 1. Introducción .....	10
1.1. Introducción .....	10
1.2. Objetivos.....	12
1.3. Estado del arte sistemas de localización de interiores .....	12
1.3.1. Métodos y técnicas de localización .....	13
1.3.2. Técnica del fingerprint.....	16
1.4. Estructura de la memoria .....	19
Capítulo 2. Tecnología Bluetooth Low Energy .....	20
2.1. Capa física y canales BLE .....	20
2.2. Modo beacon y envío de mensajes de advertising .....	22
2.2.1. Formato lbeacon.....	25
2.3. Beacons iBKS de Accent Systems .....	26
2.4. Modo scanner para detección de beacons .....	29
2.5. Comandos HCI en Linux para implementar el modo scanner .....	31
2.5.1. Implementación del escáner BLE con Bluez .....	31
2.5.2. Implementación de escáner BLE con Python .....	33
Capítulo 3. Planificación RF de los beacons mediante la métrica de visibilidad .....	35
3.1. Variables que afectan al canal inalámbrico .....	35
3.2. Análisis de la capa física mediante el <i>Packet Reception Rate</i> y la RSSI.....	40
3.3. Definición de la métrica de visibilidad y matriz de visibilidad.....	44
3.3.1. Planificación RF beacons en entorno real .....	44
Capítulo 4. Arquitectura del servicio de localización zonal .....	47
4.1. Diagrama de bloques arquitectura de localización zonal .....	47
4.1.1. Beacons BLE .....	49
4.1.2. Dispositivo IoT .....	50
4.1.3. Receptor BLE.....	50
4.1.4. Gestor de datos.....	52
4.1.5. Calibración.....	53
4.2. Motor de localización basado en kNN.....	56
4.2.1. Modo de operación online del motor KNN .....	58

4.2.2. Modo de operación offline y evaluación del desempeño del sistema.....	59
Capítulo 5. Resultados y conclusiones.....	62
5.1. Evaluación del desempeño de la estimación de la zona real .....	64
5.1.1. Confusion matrix de las sesiones de test por cada dispositivo. ....	66
5.1.2. Resumen del desempeño en las sesiones de test.....	70
5.2. Análisis de los resultados de desempeño de las sesiones de test .....	71
5.3. Conclusiones y trabajos futuros .....	72
5.3.1. Trabajos futuros .....	73
Bibliografía y enlaces .....	74

## Índice de figuras

Figura 1. Mapa de la vivienda sobre la que se implementa el sistema de localización.....	11
Figura 2. Intersección de señales basadas en medidas de rangos o distancias.....	14
Figura 3. Cálculo de ángulos en la intersección de señales. ....	15
Figura 4. Intersección de señales basados en medida de potencia (RSS) .....	16
Figura 5. Obtención de la posición del nodo cliente.....	18
Figura 6. . Canales utilizados en bluetooth low energy. ....	21
Figura 7. Estructura de un mensaje BLE.....	22
Figura 8. Estructura de un mensaje advertising .....	22
Figura 9. Advertising interval .....	23
Figura 10. Estructura iBeacon .....	25
Figura 11. Chasis beacon iBKS.....	26
Figura 12. PCB beacon iBKS.....	26
Figura 13. Configuración de iBKS (1).....	27
Figura 14. Configuración de iBKS (2).....	28
Figura 15. Scanning interval.....	30
Figura 16. Cronograma envío-recepción advertising.....	30
Figura 17. Comando HCI para escáner BLE. ....	31
Figura 18. Parar escaneo BLE .....	32
Figura 19. Iniciar escaneo BLE.....	32
Figura 20. LE Set Scan Enable Command .....	33
Figura 21. Configuración de parámetros para escáner BLE .....	33
Figura 22. Inicio de escaneo BLE .....	34
Figura 23. Parada de escaneo BLE .....	34
Figura 24. Potencia de recepción frente a distancia. IBKS 105.....	35
Figura 25. Diagramas de una antena direccional (Yagi).....	36
Figura 26. Diagramas de una antena omnidireccional (Dipolo) .....	37
Figura 27. Duración de la batería (meses) en función de la ganancia de transmisión .....	37
Figura 28. Distintos factores que afectan al nivel de señal .....	39
Figura 29. Relación entre SNR y Noise Floor.....	40
Figura 30. Puntos escogidos y orientación del entorno de pruebas.....	42
Figura 31. Alumno realizando pruebas con beacon BLE en verde.....	45
Figura 32. Distribución de beacons sobre escenario .....	45
Figura 33. Arquitectura global del sistema de localización .....	47
Figura 34. Herramientas de desarrollo utilizadas .....	48
Figura 35. Escenario de implantación del sistema.....	49
Figura 36. Diagrama de bloques receptor BLE.....	51
Figura 37. Diagrama de bloques del módulo “Gestor de datos” .....	52
Figura 38. Diagrama de bloques de la calibración .....	54
Figura 39. Clasificador KNN con dos clases.....	56



Figura 40. Funcionamiento del algoritmo KNN.....	57
Figura 41. Diagrama de flujo de motor KNN.....	58
Figura 42. Diagrama de bloques de motor KNN online .....	59
Figura 43. Diagrama de flujo de motor offline.....	60
Figura 44. Dispositivos de medición utilizados. ....	63
Figura 45. Compromiso de precisión y exactitud.....	64

## Índice de tablas

Tabla 1. Intersección de señales basados en indicador de potencia RSSI .....	18
Tabla 2. Comparación de protocolos Bluetooth .....	20
Tabla 3. Tipos de advertising .....	24
Tabla 4. Distancia (m) desde punto de medición a beacon 1 .....	41
Tabla 5. PRR, media de RSSI y desviación típica para los puntos tomados. ....	43
Tabla 6. Matriz de visibilidad obtenida con medición con portátil Dell. ....	46
Tabla 7. Fingerprint de calibración generado .....	55
Tabla 8. Fingerprint de visibilidad .....	56
Tabla 9. Fichero de desempeño generado.....	61
Tabla 10. Días de testeo y calibración.....	65
Tabla 11 . Confusion matrix día 1, dispositivo 1 .....	66
Tabla 12. Confusion matrix día 2, dispositivo 1 .....	66
Tabla 13. Confusion matrix día 3, dispositivo 1 .....	67
Tabla 14. Confusion matrix día 4, dispositivo 1 .....	67
Tabla 15. Confusion matrix día 1, dispositivo 2 .....	68
Tabla 16. Confusion matrix día 2, dispositivo 2 .....	68
Tabla 17. Confusion matrix día 3, dispositivo 2 .....	69
Tabla 18. Confusion matrix día 4, dispositivo 2 .....	69
Tabla 19. Accuracy global y zonales.....	70
Tabla 20. Precisión global y zonal .....	70
Tabla 21. Recall global y zonal .....	71

# Capítulo 1. Introducción

## 1.1. Introducción

El presente TFM propone el diseño y desarrollo de una solución completa cuya finalidad es la localización de una persona mediante un dispositivo IoT en espacios de interior mediante la tecnología Bluetooth Low Energy (BLE).

Como caso de uso se ha implementado y probado un prototipo en un escenario real consistente en una vivienda familiar (la del autor del proyecto en este caso). Se definen una serie de requisitos técnicos a cumplir:

- Usar tecnologías con estándares maduros y que exista disponibilidad de productos de varios fabricantes.
- Tener una solución escalable y adaptable a cualquier vivienda y que ofrezca una identificación zonal (nivel de habitación).
- Desarrollar un sistema con el mínimo cableado posible y de fácil instalación.
- Los datos de localización se tienen que refrescar en ventanas temporales de 4-5 segundos como máximo.
- Que el coste para implantar el sistema en una vivienda sea el mínimo posible.

Como hipótesis de trabajo se ha considerado que la tecnología RF que es idónea para este proyecto es Bluetooth Low Energy, ya que tiene un bajo consumo energético, opera en una banda de frecuencia libre de 2,4 GHz a una baja potencia y existen numerosos dispositivos y productos comerciales de bajo coste. En particular, los beacons BLE son dispositivos de bajo coste (5-10€), reducido tamaño y alimentados a baterías que emiten mensajes periódicos (denominados *advertisements*); un beacon BLE es muy fácil y rápido de instalar, la duración de la batería puedes ser de muchos meses hasta el año.

La arquitectura del servicio de localización zonal escogida consiste en la distribución de varios beacons BLE en distintos puntos estratégicos de la vivienda y la asunción que una persona lleva un dispositivo IoT con un módulo BLE que es capaz de recibir los mensajes de los beacons BLE; a partir del procesamiento y triangulación de las señales periódicas de los beacons, se puede inferir la zona donde se encuentra una persona.

El escenario real de pruebas se muestra en la siguiente figura donde se representa el mapa 2D de la planta baja de una vivienda con un área de 160 metros cuadrados, y dónde se disciernen 10 zonas que serán la salida del sistema de localización.

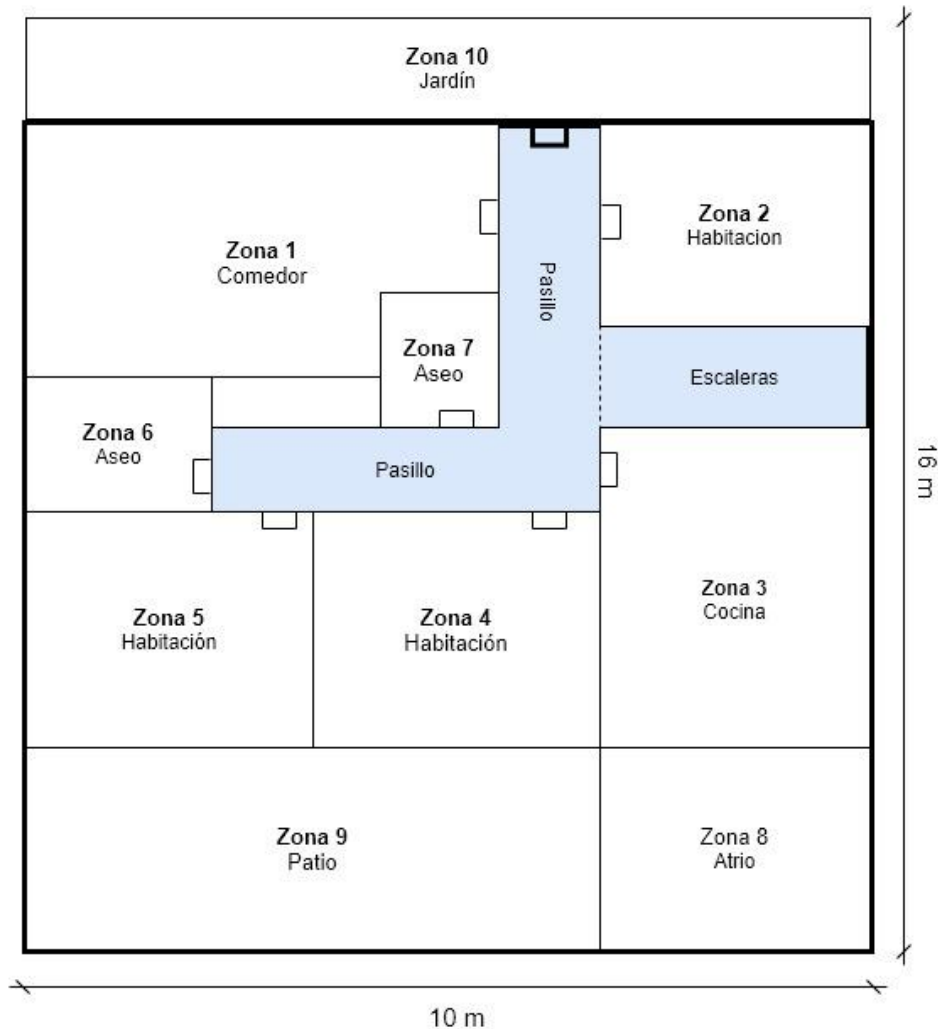


Figura 1. Mapa de la vivienda sobre la que se implementa el sistema de localización.

En este TFM se ha trabajado en diseñar un sistema completo e implementar un demostrador tecnológico para la identificación y localización de un individuo en zonas de la vivienda mediante la adaptación de tecnologías de localización inalámbricas basadas en Bluetooth Low Energy. Como visión de conjunto, el principio de funcionamiento del sistema implementado es el siguiente:

1. Se instalan varios beacons BLE en puntos estratégicos de la vivienda.
2. El dispositivo IoT de la persona escanea y muestrea los patrones de señal de los beacons en cobertura y envía dichos vectores de datos (fingerprints) a un servidor para su procesamiento. Para el envío periódico de datos del dispositivo IoT al Servidor se usa la red Wifi del hogar.
3. En cada zona se calibra los patrones de señal de los beacons muestreados y se genera una base de datos con datos etiquetados, denominado fingerprint de calibración.
4. El Servidor recibe los fingerprint de un instante muestreados por el dispositivo IoT y aplicando un algoritmo de clasificación kNN, infiere la zona donde se encuentra la persona.

## 1.2. Objetivos

Un primer objetivo ha sido la formación y estudio de sistemas de localización zonal y el uso de las tecnologías IoT de comunicaciones aplicadas al problema de la localización. EL siguiente objetivo ha sido el diseño e implementación de una arquitectura de localización de interiores para IoT usando la tecnología Bluetooth Low Energy consiste en tener un conjunto de beacons BLE desplegados en un entorno y un dispositivo IoT escáner, asociado a una persona, que monitorizan el ID de los beacons en cobertura y su nivel de señal.

Mediante técnicas de machine learning se procesan los patrones de señales recibidos de los beacons en el dispositivo IoT e inferir su ubicación zonal. La tecnología Bluetooth Low Energy opera en la banda ISM de 2,4GHz y el nivel de señal es muy complejo de predecir porque está afectado por numerosos factores como el multipath y el fading, fluctuaciones del SNR debido a interferencias (existencia de redes Wifi que operan a 2,4GHz) y ruido ambiente, movilidad del dispositivo, etc.

Se definen una serie de objetivos técnicos a cumplir durante el desarrollo del proyecto:

1. Un primer objetivo es definir un modelo cualitativo de la capa física y enlace BLE que permita entender empíricamente el desempeño RF de los beacons BLE en un entorno. Para ello se trabajará a bajo nivel con módulos nRF52 de Nordic para desarrollar un sniffer a medida y realizar medidas empíricas de la capa física y enlace.
2. Aplicar la técnica del fingerprint en el contexto de BLE. La técnica del fingerprint es ampliamente usada en sistemas de localización y consiste en generar un fingerprint (o huella de la señal característica) de un entorno formado a partir de los patrones de señal de los beacons BLE en cobertura. Mediante técnicas de machine learning, procesando los fingerprint muestreados se puede inferir y estimar la zona donde se encuentra una persona.
3. Finalmente, el último objetivo es pasar de las pruebas de laboratorio a un entorno real demostrativo. Se trata de abordar una metodología rigurosa de pruebas y evaluación que permita extraer conclusiones sobre el desempeño en un entorno real.

## 1.3. Estado del arte sistemas de localización de interiores

La localización en interiores, a día de hoy, se encuentra en plena expansión a causa de las deficiencias de la geolocalización en interiores. En entornos indoor no funciona el GPS debido a la pérdida de potencia o atenuación de señal de los satélites GPS. Antes de entrar en detalles del sistema a desarrollar, se va a hacer una pequeña introducción en lo que se ha llamado sistemas de localización basados en radiofrecuencia de corto alcance para entornos indoor.

La posición de un objeto (o usuario) móvil es sencilla de estimar si este se encuentra en el exterior. Si existe visión directa con los satélites, un sistema de localización, como los sistemas de navegación basados en GPS, tiene una estimación muy aproximada de la posición del objeto o usuario con un error de unos pocos metros.

Otro sistema de localización en exteriores, que utiliza la red de telefonía móvil para estimar una posición, no necesita estar en un espacio totalmente despejado, ya que hace uso de varias estaciones o BTS para triangular la posición de un objeto móvil. Estas BTS pueden tener un radio de unos 35 Km y suelen

encontrarse en núcleos urbanos. Mediante diferencias de tiempos en la comunicación con las distintas BTS se obtiene una precisión de unos 20 o 30 metros. Si a esto le añadimos la posibilidad de sincronizarse con una conexión GPS, reducimos aún más el posible error de cálculo a unos cuantos metros, como si estuviéramos a cielo abierto.

Estos sistemas nombrados anteriormente no son válidos para estimar la posición de un móvil que se encuentra en el interior de un edificio. Por un lado, en los sistemas satelitales basados en GPS, las señales de los satélites no pueden usarse en entornos interiores debido a que no tienen la potencia suficiente para atravesar la mayoría de obstáculos. Por otro lado, los sistemas de telefonía móvil tienen un problema similar con la recepción de la señal y además se le suma el problema de la poca precisión que ofrecen, por lo que no se considera un sistema eficiente para la localización.

La alternativa es la utilización de infraestructuras inalámbricas como sustitutos de GPS y la telefonía móvil para ofrecer una cobertura y precisión adecuada a los servicios de localización indoor. Por ejemplo, existen numerosas propuestas de sistemas basados en Wifi debido a su uso generalizado. Desde hace mucho tiempo es objeto de estudio e investigación por parte de diferentes empresas y universidades cuyo fin es encontrar una solución sencilla y fiable. Sin embargo, no existe una solución y tecnología única para cualquier caso de uso de localización.

Un factor importante son los costes de la infraestructura ya que la mayoría de los métodos necesitan tener una estructura fija (nodos, sensores, estaciones base, etc.) para dar soporte a todo el sistema. Con el paso del tiempo, los costes han disminuido y han ido apareciendo otras técnicas bastante más asequibles.

Independientemente de la infraestructura inalámbrica, los sistemas de localización se pueden clasificar en función de la métrica de RF que se mide entre el emisor y el receptor de la señal inalámbrica. Las métricas de la señal RF a medir principales son: el tiempo de llegada (ToA, *Time of Arrival*), la diferencia entre tiempos de llegada (TDoA, *Time Difference of Arrival*), el ángulo de llegada (AoA, *Angle of Arrival*), o el nivel de potencia recibida de la señal (RSS, *Received Signal Strength*). A partir de la métrica RF medida se puede triangular la señal es estimar la posición. También hay que decir que existen técnicas que combinan varias de estas métricas para aumentar y mejorar la precisión del sistema de localización.

### 1.3.1. Métodos y técnicas de localización

Es realmente complejo crear un modelo que se ajuste a la propagación en interiores como consecuencia de una serie de problemas que interfieren en la recepción de la señal, como el multipath o los parámetros propios de la construcción del entorno, destacando la complejidad en la distribución, los objetos en movimiento y los materiales. A tenor de estos inconvenientes y circunstancias, han surgido alternativas que tratan de atenuar estos errores de precisión como son los algoritmos de posicionamiento mediante análisis de escena o de proximidad. Otros, como la triangulación tradicional, han sufrido una transformación para su adaptación en entornos de interior.

En general, se entiende que un sistema de localización está formado de un lado, por una serie de dispositivos que emiten una señal al medio ya sean ultrasonidos, radio e incluso luz, y de otro, por dispositivos electrónicos que reciben esa señal tras su propagación a través de dicho medio. Una vez en posesión de esa información, se tiene que procesar para poder estimar la posición del usuario.

Existen diferentes técnicas para medir la posición basándonos en ciertas características de la señal transmitida: ToA, TDoA, AoA y RSS

#### 1.3.1.1. Tiempo de Llegada (ToA)

Esta técnica ToA (*Time of Arrival*) se basa en la relación existente entre el tiempo de propagación de una señal y la distancia entre emisor y receptor. Con el fin de permitir la localización, las mediciones deben ser realizadas con respecto a las señales de al menos tres puntos de referencia. Para los sistemas basados en ToA, se mide el tiempo de propagación de ida y la distancia entre la unidad de medición y el transmisor de la señal.

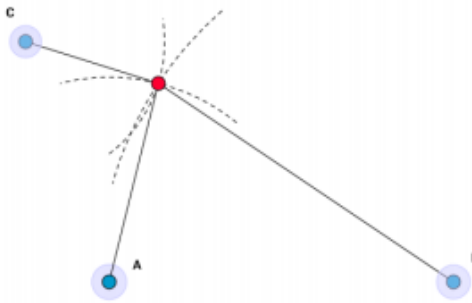


Figura 2. Intersección de señales basadas en medidas de rangos o distancias

La medición ToA tiene inherente una serie de dificultades, por un lado, los transmisores y receptores en el sistema tienen que estar sincronizados previamente, y por otro debe llevar etiquetada con una marca de tiempo la señal de transmisión para que el receptor pueda calcular la distancia recorrida de la señal. Además, este método debe emplear un sistema de sincronización de reloj muy robusto y preciso para minimizar los errores.

#### 1.3.1.2. Diferencia en Tiempo entre Llegadas (TDoA)

Basado en el método anterior ToA, en la técnica TDoA (*Time Difference of Arrival*) no se usan los tiempos absolutos, lo que facilita eliminar la costosa sincronización entre los relojes de transmisores y receptores. La técnica consiste en determinar la posición relativa del móvil calculando los diferentes tiempos de llegada de la señal en diferentes medidas. Se producen dos señales, una acústica y otra electromagnética. Como el tiempo de propagación es diferente para cada medio, la acústica viaja a la velocidad del sonido y la electromagnética lo hace a la velocidad de la luz, el receptor puede calcular la distancia mediante la diferencia de tiempo entre ambas señales.

Aunque se ha eliminado la sincronización con los receptores, se tiene que mantener la sincronización entre los emisores, necesaria para poder hacer los cálculos sin error en los tiempos de emisión de las señales, por lo que arrastra la misma desventaja del método anterior.

### 1.3.1.3. Ángulo de Llegada (AoA)

En la técnica AoA (*Angle of Arrival*) se determina la ubicación de un objeto a partir del ángulo de llegada de la señal. Su ventaja descansa en el bajo número de nodos satélites que se necesitan para el posicionamiento, dos para la localización en el plano y tres para una localización en tres dimensiones. Pero las desventajas gravan el método y lo hacen inoperativo para el posicionamiento. Para poder calcular el ángulo de llegada de la señal correctamente, emisor y receptor tienen que estar en línea de visión directa para evitar una angulación incorrecta de la señal de llegada (multipath), difícil de conseguir en posicionamiento en interiores.

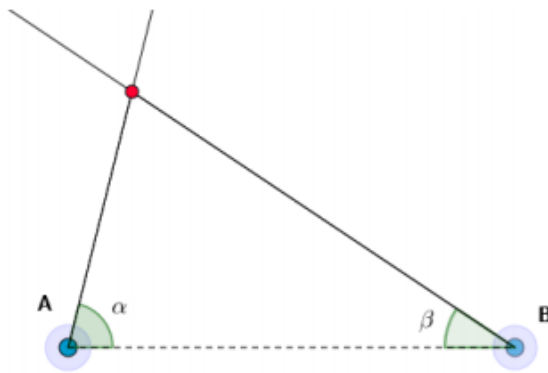


Figura 3. Cálculo de ángulos en la intersección de señales.

### 1.3.1.4. Potencia de la señal recibida (RSS)

La métrica RSS (*Received Signal Strength*) se basa en medir el nivel de potencia de la señal recibida de un transmisor en un receptor. De forma simplificada depende de la potencia de transmisión, la distancia entre el transmisor y el receptor y la atenuación de la señal que haya según el modelo de propagación del entorno. El fundamento de este método está basado en la atenuación que sufre la señal a medida que el receptor se aleja del emisor.

El principal problema es que la atenuación no es constante y puede variar dependiendo, además de la distancia, de factores externos como pueden ser las interferencias con otras señales o las reflexiones, difracciones o dispersiones en objetos o personas. La ventaja de esta técnica es que no es necesario la sincronización entre emisor y receptor para su cálculo, es de bajo coste y de fácil implementación.



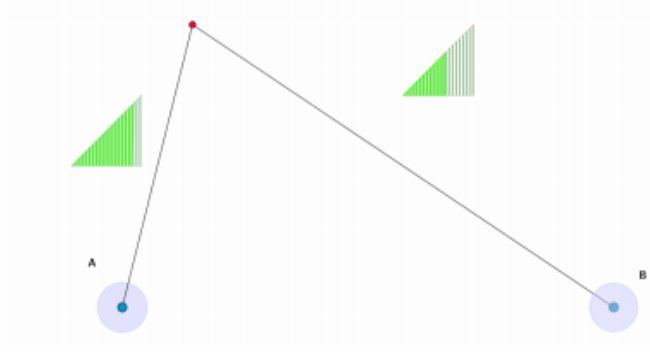


Figura 4. Intersección de señales basados en medida de potencia (RSS)

Aunque estos inconvenientes pueden insinuar que esta técnica no sea fiable para determinar una posición, será el método elegido para utilizar en el proyecto por ser la más sencilla de implementar y no requiere hardware adicional. Además, no requiere ningún cálculo ya que es el propio hardware el que nos va a proporcionar el valor (RSS) de potencia de la señal recibida mediante la lectura de la propia señal.

Se pueden diferenciar dos métodos de trabajo para la localización por RSS, según quien sea el transmisor y el receptor. Si se considera a los nodos satélites como emisores o transmisores de la señal y como receptor el nodo cliente se denomina sistema unilateral y es el nodo cliente el que hace la estimación de su ubicación. Si se considera transmisor al nodo cliente y receptores a los nodos satélites entonces tenemos lo que denomina Binsky sistema multilateral. De estos dos sistemas, en lo concerniente a este proyecto, se considerará que trabajaremos sobre un sistema multilateral.

### 1.3.2. Técnica del fingerprint

Técnica basada en el reconocimiento de patrones, en la comparación de estos dentro de una base de datos entre la del sistema, datos guardados en la fase de captura, y la del nodo cliente (el vector de potencias recibidas de los nodos satélites). Se usa en ambos modos, multilateral o unilateral. Hay dos modelos de Fingerprinting:

- Búsqueda del nodo satélite más cercano o
- Búsqueda por métodos estadísticos.

En cualquiera de los dos modelos hay que pasar por la fase previa (su principal inconveniente) llamada fase de entrenamiento.

$RSS_p = (AP_1, AP_2, AP_3, \dots, AP_k)$ , donde k es el número total de nodos satélites.

Para el método de búsqueda del nodo satélite más cercano, se compara el  $RSS_p$  de todos los puntos con el RSS cliente y escoge como posición del usuario la correspondiente a la distancia euclidiana mínima.

En un primer momento se ha de decidir el área a cubrir y elegir unos puntos determinados para tomar las medidas. Escoger los nodos satélites mínimos de acceso para que el error que se cometa entre la posición calculada y la real sea lo más aceptable posible. En un sistema unilateral, se mapea el área de búsqueda con P puntos, de ubicaciones conocidas, y se mide la potencia de cada nodo satélite para cada uno de esos puntos. A esto se le llama vector de potencias (RSSp). Por tanto, habrá tantos vectores como P puntos de mapeo. Estos vectores se guardan en una base de datos.

$RSSp = (AP_1, AP_2, AP_3, \dots, AP_k)$ , donde k es el número total de nodos satélites.

Para el método de búsqueda del nodo satélite más cercano, se compara el RSSp de todos los puntos con el RSS cliente y escoge como posición del usuario la correspondiente a la distancia euclidiana mínima.

En el método estadístico se utiliza, para el cálculo de la distancia, el teorema de Bayes, calculando la probabilidad de que el cliente se encuentre en el estado  $e_p$  (cada punto de captura es un estado), cuando él ha medido su observable  $O'$ .

$$P(e_p | O') = \frac{P(O'|e_p) P(e_p)}{P(O')}$$

Cuanto mayor sea la probabilidad más factible es que se encuentre el cliente en el estado  $e_p$ .

Tanto en uno como en el otro método, la calidad del posicionamiento depende del número de nodos satélites, de los puntos de referencia en el mapeo y de los efectos generados por la propagación. La interferencia Bayesiana suele dar mejores resultados que el método Euclidiano, en cualquier caso son necesarias muchas medidas para que el sistema sea lo más eficaz posible.

Existen otros sistemas para el cálculo probabilístico de posiciones, desde máquinas de soporte vectorial hasta algoritmos basados en redes neuronales, pasando por funciones de densidad como el método de Kernel. Cada uno, a su manera, tiene un gran número de opciones y posibilidades. Solamente la complejidad de implementación y ejecución, así como la necesidad o no de tiempo de procesado y de almacenamiento determina la elección de uno u otro método.

En cuanto al método determinista más utilizado es el KNN, K-Nearest Neighbors. Este consiste en utilizar una matriz semejante a la de la tabla 1, para obtener los puntos de la matriz más próximos a la posición del nodo cliente. Una vez que se obtienen, se promedian y determina el punto más próximo y por tanto a su posición.

Posición X	Posición Y	AP-1	AP-2	AP-3	...	AP-n
X <sub>1</sub>	Y <sub>1</sub>	RSSI <sub>1,1</sub>	RSSI <sub>1,2</sub>	RSSI <sub>1,3</sub>	...	RSSI <sub>1,n</sub>
X <sub>2</sub>	Y <sub>2</sub>	RSSI <sub>2,1</sub>	RSSI <sub>2,2</sub>	RSSI <sub>2,3</sub>	...	RSSI <sub>2,n</sub>
...	...	...	...	...	...	...
X <sub>n</sub>	Y <sub>n</sub>	RSSI <sub>n,1</sub>	RSSI <sub>n,2</sub>	RSSI <sub>n,3</sub>	...	RSSI <sub>n,n</sub>

Tabla 1. Intersección de señales basados en indicador de potencia RSSI

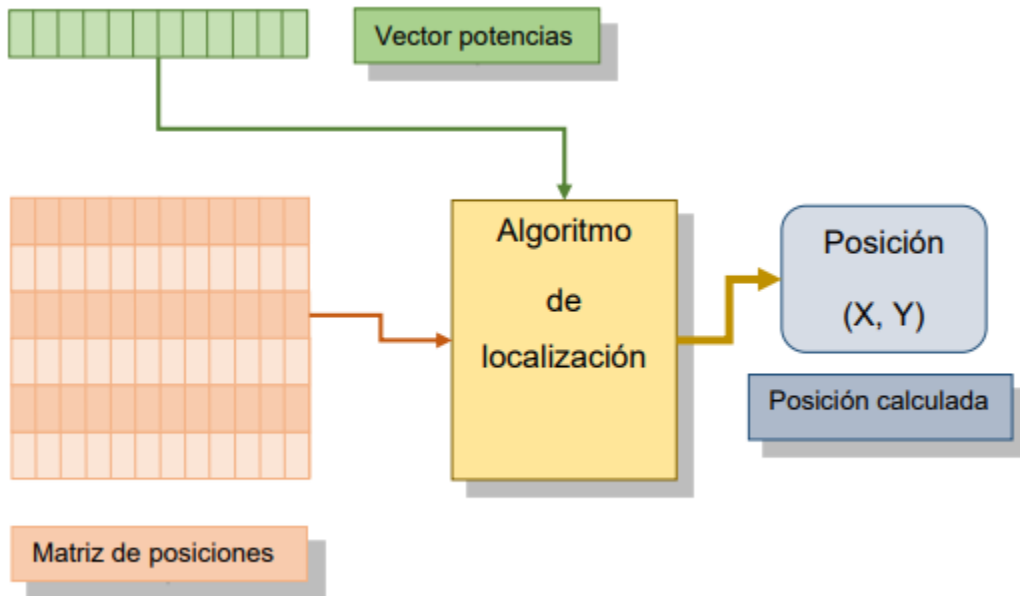


Figura 5. Obtención de la posición del nodo cliente

Para estimar la posición, es necesaria la matriz de posiciones obtenida anteriormente y el vector de potencias de la posición a calcular. Por supuesto, hay que tener en cuenta que todas estas mediciones son válidas siempre que no varíen de posición los nodos satélites o se incorporen nuevos elementos que atenúen la señal.

Teniendo un conjunto de medidas  $X = x_1, x_2, x_3, \dots, x_n$  obtenidas en un punto a determinar su posición, definimos la matriz con sus vectores de potencia tal que:

$$N = \begin{pmatrix} S_{1,1} & S_{1,2} & S_{1,3} & \dots & S_{1,n} \\ S_{2,1} & S_{2,2} & S_{2,3} & \dots & S_{2,n} \\ S_{3,1} & S_{3,2} & S_{3,3} & \dots & S_{3,n} \\ S_{4,1} & S_{4,2} & S_{4,3} & \dots & S_{4,n} \end{pmatrix}$$

Donde  $S_{1,1}$  es el índice de potencia (RSSI) del nodo satélite 1 en la primera posición de la matriz, donde se define el vector de potencia en la posición  $p$  como:

$$F(p) = (s_{p,1}, s_{p,2}, s_{p,3}, \dots, s_{p,n})$$

Para de esta forma poder calcular la distancia entre cada vector de potencia de la matriz  $F(p)$  y el vector de potencias medio,  $X$ .

$$dist_p = \frac{1}{n} \sqrt{(x_1 - s_{p,1})^2 + (x_2 - s_{p,2})^2 + \dots + (x_n - s_{p,n})^2}$$

Una vez se han obtenido todas las distancias para todos los vectores de la matriz de posiciones, se seleccionan los vectores cuya distancia euclídea al vector de potencia sea menor y por lo tanto más cercanas al nodo cliente. Se puede promediar usando métodos como la media aritmética o el cálculo de centro de masas (tomando como peso la inversa de la distancia calculada) de tal modo que los puntos más cercanos tengan mayor peso y nos devuelva el más cercano a la posición real del nodo cliente.

Es un método que no es muy complejo y los cálculos son fáciles de implementar en el sistema, pero sí hay que tener en cuenta que el tamaño de la matriz inicial influye en el tiempo de procesado para el cálculo de las distancias.

## 1.4. Estructura de la memoria

La memoria de este Trabajo Final de Máster se encontrará dividida en 5 capítulos. A continuación, se cita y explica brevemente cada parte de este trabajo:

- El primer capítulo es en el que nos encontramos. Su utilidad es presentar el proyecto con los objetivos que se pretendían lograr al comienzo del mismo.
- El objetivo del segundo capítulo es exponer la base teórica de la tecnología utilizada.
- En el tercer capítulo se tratará el diseño del sistema en la capa de RF, y los factores a tener en cuenta para un correcto funcionamiento.
- En el cuarto capítulo de la memoria se explicará la arquitectura llevada a cabo, tanto hardware como software, en este proyecto.
- Finalmente, el quinto capítulo contiene las conclusiones obtenidas a partir de los resultados. Además, se propondrán unas futuras líneas de desarrollo para mejorar la aplicación.

## Capítulo 2. Tecnología Bluetooth Low Energy

Bluetooth Low Energy (BLE) es la nueva especificación de la tecnología Bluetooth desarrollado por Bluetooth Special Interest Group (SIG). Se ha diseñado como una tecnología complementaria a Bluetooth clásico para garantizar un consumo de energía bajo.

BLE se trata de un nuevo estándar con fines y aplicaciones diferentes y está diseñado para la transmisión de pequeñas cantidades de datos, lo que permite que los dispositivos tengan periodos de autonomía muy altos.

### 2.1. Capa física y canales BLE

Bluetooth Low Energy comparte algunas similitudes con el Bluetooth clásico. Los dos usan la banda de 2.4 GHz. Bluetooth clásico y BLE usan la modulación GFSK a 1Mbps, pero con índices de modulación diferentes. Enhanced Data Rate (EDR) usa una modulación completamente diferente de la GFSK. El estándar Bluetooth clásico tiene 79 canales mientras BLE tiene 40 tal como indica la tabla que viene a continuación. La separación entre canales también es diferente. Debido a estas dos diferencias entre BLE y Bluetooth clásico, estos son incompatibles entre sí, por lo tanto, no se pueden comunicar. Sin embargo, existen dispositivos Dual Mode que soportan las dos tecnologías conmutando los parámetros de modulación y los canales donde se está radiando.

	<b>BLE</b>	<b>BR</b>	<b>EDR</b>
<b>Modulación</b>	GFSK 0.45 to 0.55	GFSK 0.28 to 0.35	DQPSK / 8DPSK
<b>Tasa Mbit/s</b>	1Mbit/s	1 Mbit/s	2 and 3 Mbit/s
<b>Nº Canales</b>	40	79	79
<b>Separación</b>	2MHz	1MHz	1MHz

*Tabla 2. Comparación de protocolos Bluetooth*

La capa física es la que se encarga de enviar las señales al aire, transmitiendo y recibiendo bits usando ondas radio en la banda de frecuencia ISM 2.4 GHz que se extiende desde 2402 MHz hasta 2480 MHz. La separación entre los 40 canales utilizados es de 2 MHz (numerados de 0 a 39 y de 1 MHz de anchura cada uno). Existen 3 canales dedicados para el Advertising y 37 para la transmisión de datos. Los canales 37, 38, y 39 son usados sólo para el envío de paquetes de Advertising. El resto son usados para el intercambio de datos durante la conexión.

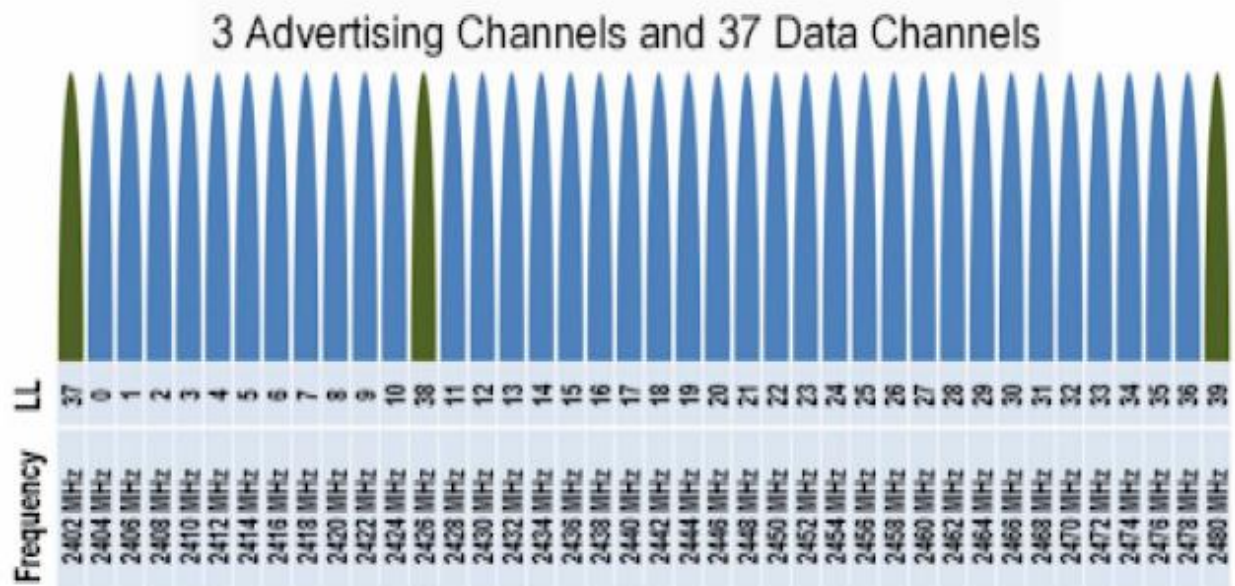


Figura 6. . Canales utilizados en bluetooth low energy.

Estos tres canales de Advertising están situados estratégicamente para evitar interferencias causadas por otras tecnologías que coexisten en el mismo espectro (IEEE 802, y ZigBee). BLE utiliza una modulación gaussiana con desplazamiento de frecuencia. Esta utiliza dos frecuencias para identificar el bit “1” o “0”. El filtro gaussiano se usa para suavizar las transiciones entre frecuencias y reducir el ensanchado de espectro causado por la ISI. La especificación de BLE limita la potencia transmitida máxima a +10 dBm y la mínima en -20 dBm. La sensibilidad recibida mínima requerida para BLE es de -70 dBm, aunque la mayoría de dispositivos BLE tienen una sensibilidad menor a -85 dBm.

El estándar BLE define dos formatos de mensajes, uno para advertising (con un payload de 0-37 bytes) y otro para datos (con un payload de hasta 25 bytes). Los mensajes de advertising se envían exclusivamente en los canales de advertising y sirven para un mecanismo sencillo de descubrimiento de dispositivos o para establecer conexiones entre dispositivos. Una vez que dos dispositivos Bluetooth Low Energy se conectan (parean) se usan el resto de canales de datos.

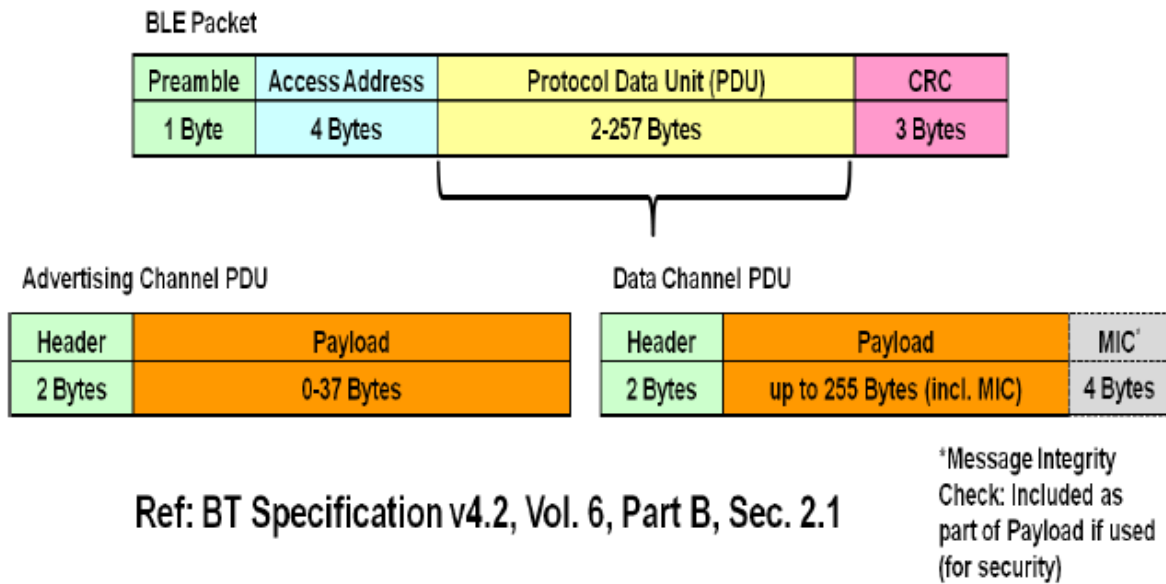


Figura 7. Estructura de un mensaje BLE

## 2.2. Modo beacon y envío de mensajes de advertising

Un Beacon o “baliza”, es un pequeño dispositivo basado en tecnología Bluetooth Low Energy, que emite un mensaje con un periodo de transmisión muy pequeño y una baja tasa de transferencia. Esto se traduce en enviar una pequeña cantidad de datos cada muy poco tiempo. Cada beacon tiene un número único que lo identifica, la MAC en este caso.

Los Beacons bluetooth son posibles de implementar gracias al modo advertising definido en el estándar BLE 4.0. La estructura del paquete Advertising es la misma que la de datos, reduciendo considerablemente el tamaño de su payload. Los bits se transmiten de derecha a izquierda. Por ejemplo, 0x80 se transmite como 00000001 enviando el bit menos significativo primero.

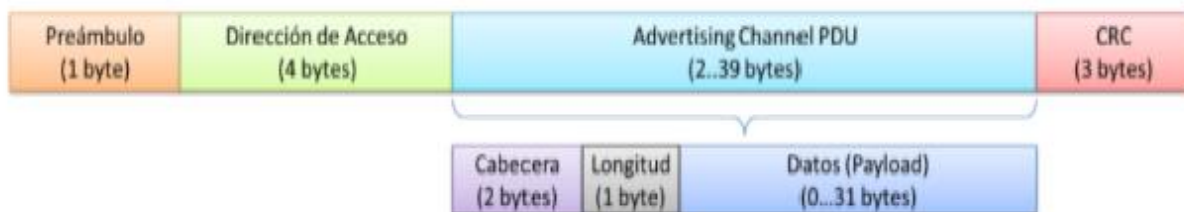


Figura 8. Estructura de un mensaje advertising

- Preámbulo (1 byte): patrón donde los 8 primeros bits son 10101010.
- Dirección de Acceso (4bytes): el estándar fija una dirección de acceso de 4 bytes que siempre tiene el valor en hexadecimal del 0x8E89BED6.
- Cabecera (2 bytes): el contenido de este campo depende de si es un paquete de Advertising de datos. Para un paquete de Advertising el campo de cabecera indica el tipo de paquete de Advertising.
- Longitud (1 byte): este campo indica la longitud de los datos.
- Datos (Payload) del advertising (0-31 bytes): el payload de un mensaje de advertising puede ser de tamaño variable, pero está limitado a un máximo de 31 bytes.
- CRC (3 bytes): son 3 bytes de Cyclic Redundancy Check.

Para el envío y recepción de mensajes advertising, se utilizan 3 de los 40 canales existentes en el protocolo BLE, en concreto el 37, 38 y 39. En cambio, cuando dos dispositivos se panean y se conectan entre sí, se utilizan los 37 canales restantes, así como un formato de mensaje distinto.

Se denomina Advertiser, al dispositivo BLE que utiliza los canales de Advertising para emitir datos, anunciar que es conectable y detectable o “descubrible”. A un dispositivo en estado de escaneo se le denomina Scanner. El dispositivo Scanner busca en los canales de Advertising paquetes de Advertising de otros dispositivos.

Durante el estado de Advertising, se envían paquetes de Advertising periódicamente en cada uno de los tres canales de Advertising. El intervalo de tiempo que separa el envío de estos paquetes es la suma de un intervalo jo y un retardo aleatorio. El intervalo jo es el Advertising Interval y se puede configurar entre 20 ms y 10.24 s, en pasos de 0.625 ms, excepto para los tipos Non-connectable y Scannable que como mínimo tiene que ser de unos 100 ms. El retardo es un valor aleatorio entre 0 ms y 10 ms que se añade automáticamente. Este último valor ayuda a reducir colisiones entre Advertising de dispositivos diferentes. De esta manera, BLE mejora la robustez del protocolo haciendo más fácil la búsqueda de los Advertising por parte del Scanner.

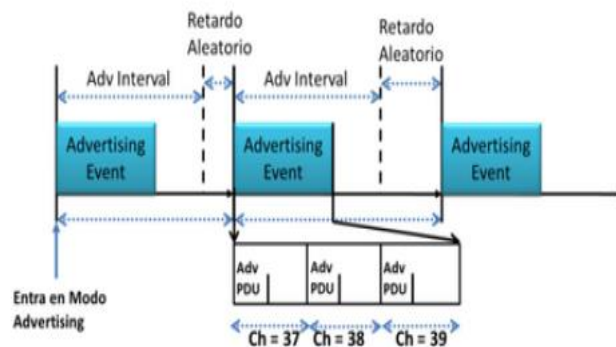


Figura 9. Advertising interval



Se distinguen cuatro tipos distintos de advertising: General, Directed, Nonconnectable y Discoverable:

- **Connectable Undirected Advertising.** Es el más común. El dispositivo escáner puede recibir los anuncios (advertising) y una vez intercambiados los datos necesarios, establecer una conexión con el dispositivo.
- **Connectable Directed Advertising.** Este tipo de advertising se usa cuando se necesita una conexión rápida. Este advertising debe repetirse cada 3.75 milisegundos. Se permite estar en este estado de anuncio un tiempo máximo de 1.28s.
- **Nonconnectable Advertising.** Este es el tipo de mensaje que se utiliza para los beacons BLE, ya que su funcionamiento es el de emitir mensajes advertising sin cesar y siendo no detectable (no puede recibir Scan Request) ni conectable.
- **Discoverable Advertising.** No se puede usar para iniciar una conexión, pero sí para ser escaneado por otros dispositivos a través de un Scan Request. El dispositivo escáner puede obtener datos del Advertiser ya que éste responde a cada Scan Request detectado con un Scan Response. También se puede utilizar para emitir datos.

<b>Advertising Packet Type</b>	<b>Connectable</b>	<b>Scannable</b>	<b>Directed</b>	<b>GAP Name</b>
<b>ADV_IND</b>	Yes	Yes	No	<i>Connectable Undirected Advertising</i>
<b>ADV_DIRECT_IND</b>	Yes	No	Yes	<i>Connectable Directed Advertising</i>
<b>ADV_NONCONN_IND</b>	No	No	No	<i>Non-connectable Undirected Advertising</i>
<b>ADV_SCAN_IND</b>	No	Yes	No	<i>Scannable Undirected Advertising</i>

*Tabla 3. Tipos de advertising*

En el caso de este TFM interesa utilizar el modo *Advertising Nonconnectable* desde un principio, ya que interesa tener una serie de “balizas” distribuidas por el escenario propuesto que no cesen de anunciarse y que, además, tengan el menor consumo posible. Esto nos permite recibir el nivel de potencia de cada uno de los Beacons pudiendo así estimar una posición del dispositivo escáner.

### 2.2.1. Formato iBeacon

A la hora de implementar un beacon se definen dos grandes estándares que especifican una estructura concreta para el payload de la trama advertising: Google eddystone y iBeacon.

En este proyecto se utilizará el estándar iBeacon, propuesto por Apple, ya que es el más utilizado comercialmente y por lo tanto se encuentran gran variedad de dispositivos que implementan este formato de advertising. Éste tiene 4 campos configurables.

- UUID (16 bytes)
- Major (2 bytes)
- Minor (2 bytes)
- Potencia transmitida (1 byte)

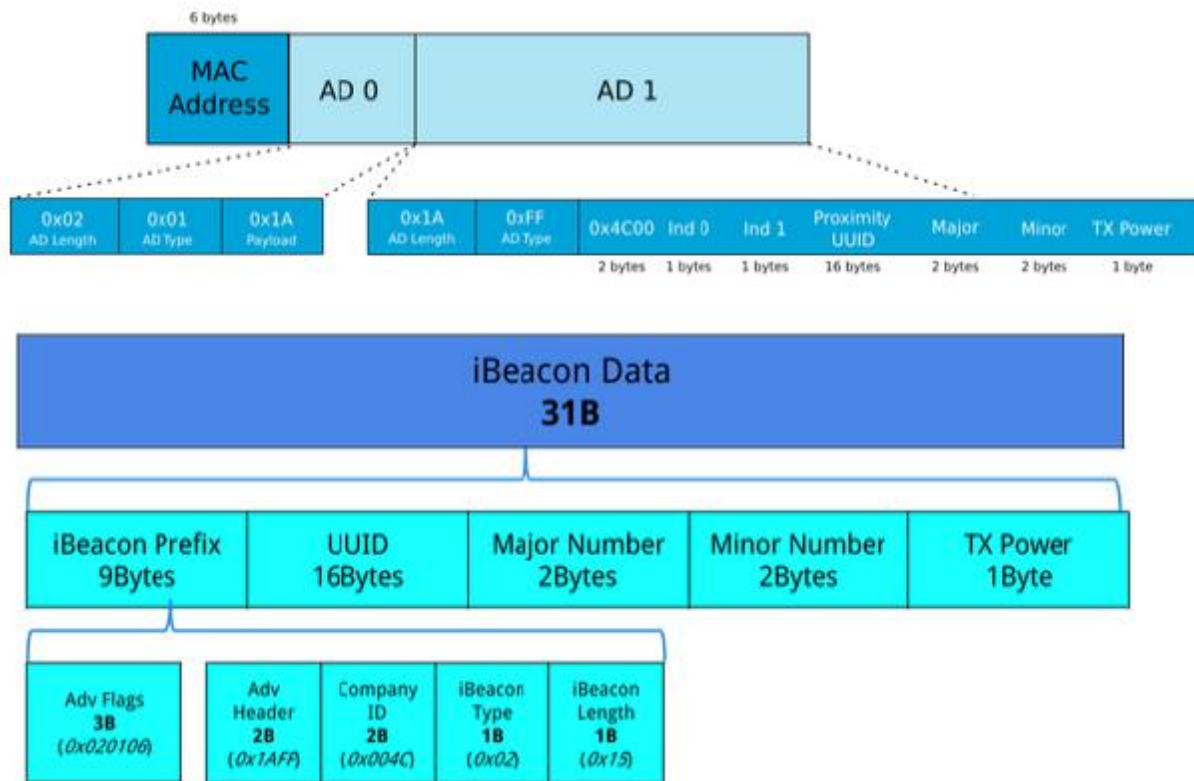


Figura 10. Estructura iBeacon

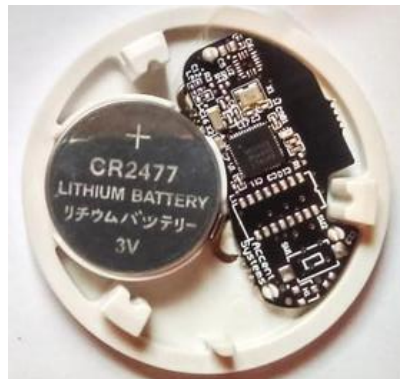
### 2.3. Beacons iBKS de Accent Systems

Para el desarrollo de este proyecto se han elegido los beacons iBKS 105 del fabricante “Accent Systems”. iBKS 105 es un beacon Bluetooth Low Energy (BLE) basado en un chipset Nordic Semiconductor nrf51822 que utiliza una pila de botón CR2477.



*Figura 11. Chasis beacon iBKS*

El PCB de estos dispositivos está preparado para implementar diferentes sensores y su chasis de plástico se puede abrir con un simple movimiento sin necesidad de herramientas, lo que permite que la instalación de estos pequeños beacons sea de lo más sencillo. Dispone de cinta de doble cara en su parte posterior pudiendo así anclarse a cualquier superficie.



*Figura 12. PCB beacon iBKS*

Este tipo de beacon BLE es compatible con el formato iBeacon y con Eddystone (UID, URL, TLM, EID) al mismo tiempo. Además, dispone de dos slots, lo que permite tener funcionando instantáneamente dos beacons virtuales configurados con distinto nivel de potencia, formato o periodo de transmisión.

Se ha escogido este beacon por su sencillez a la hora de configurarlo, ya que se realiza todo a través de una APP móvil de manera muy intuitiva. Esta APP se llama “IBKS config tool” y está disponible tanto en la App Store como en PlayStore.

Para configurar en beacon se ha de abrir la aplicación en cuestión, la cual realizará un escaneo inicial en búsqueda de los beacons iBKS existentes en el entorno. Lo próximo será localizar el beacon que se desea configurar y pinchar sobre el accediendo a la siguiente pantalla.

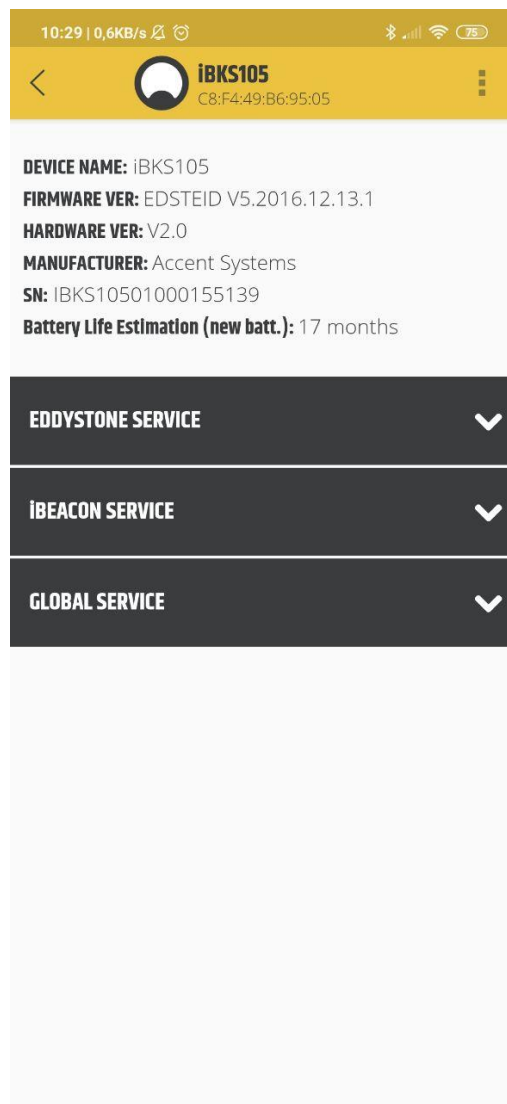


Figura 13. Configuración de iBKS (1)

A continuación, seleccionaremos el tipo de servicio que queremos configurar en el dispositivo, iBeacon o Eddystone. En el caso de este proyecto, como ya se ha comentado anteriormente, se utilizará el formato iBeacon, pudiendo configurar los siguientes parámetros a través de la aplicación.

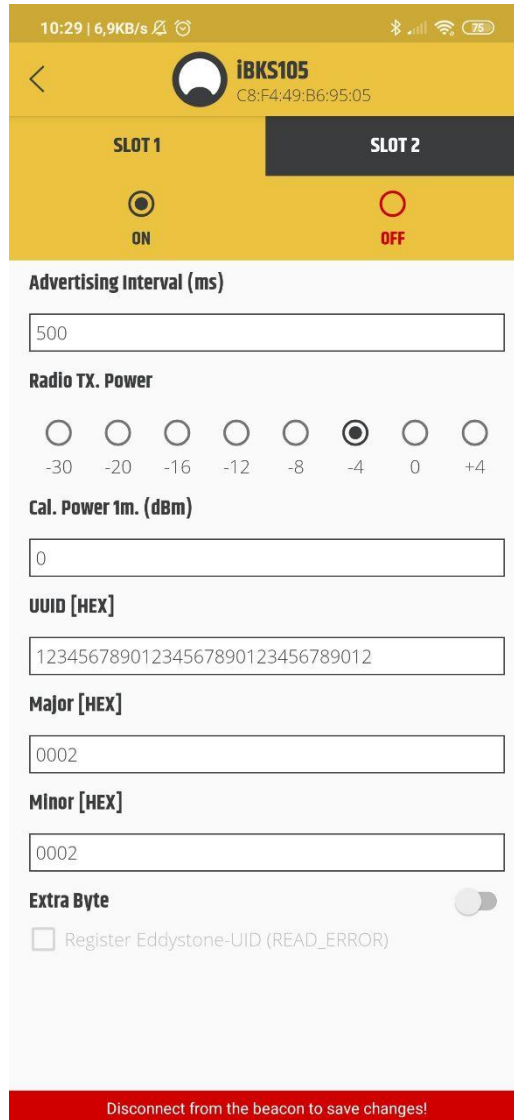


Figura 14. Configuración de iBKS (2)

Siendo:

- **SLOT:** slot BLE a utilizar, alternándose cada uno de ellos en el tiempo y pudiendo funcionar ambos de manera simultánea.
- **Advertising interval.** Intervalo de envío de mensajes advertising en milisegundos.
- **Radio TX Power.** Potencia de transmisión de la radio bluetooth, siendo configurable desde -30 dBm hasta +4 dBm.

- **Cal. Power 1m.** Potencia de calibración a 1 metro (valor teórico) en dBm.
- **UUID.** Payload de la trama BLE, utilizada en nuestro caso como identificador para un posterior filtrado en el sistema.
- **Major.** Valor de 2 bytes configurable, utilizado como identificador de beacon en nuestro sistema.
- **Minor.** Valor de 2 bytes configurable.

De esta manera tan simple quedaría configurado el beacon iBKS, funcionando siempre y cuando tenga pila disponible. Una pila de las que incluye este dispositivo tiene una duración estimada de 17-46 meses en función de la potencia de transmisión y los slots que se utilicen.

## 2.4. Modo scanner para detección de beacons

Además del funcionamiento beacon, se encuentra el modo scanner, el cual va escaneando en cada uno de los 3 canales de advertising durante un cierto tiempo configurable, antes de que se cambie al siguiente canal.

Los parámetros que se pueden configurar en un dispositivo haciendo de scanner BLE son los siguientes:

- LE Scan Type (1 octeto):
  - 0x00: Escaneo pasivo, no se envía paquetes SCAN REQ.
  - 0x01: Escaneo activo, se pueden enviar paquetes SCAN REQ.
- LE Scan Interval (2 octetos):
  - Rango: 0x0004 a 0x4000.
  - $T = N \times 0.625$  ms siendo N el valor introducido.
- LE Scan Window (2 octetos):
  - Rango: 0x0004 a 0x4000.
  - $T = N \times 0.625$  ms siendo N el valor introducido.

Este valor tiene que ser menor o igual que LE Scan Interval.

- Own Address Type (1 octeto):
  - 0x00: Dirección del dispositivo pública.
  - 0x01: Dirección del dispositivo aleatoria.
- Scanning Filter Policy (1 octeto):
  - 0x00: Sin filtro, se aceptan todos los paquetes, menos los paquetes dirigidos a otros dispositivos específicos.
  - 0x01: Se ignoran todos los paquetes de los dispositivos que no están en la lista blanca.

En nuestro caso, se fijará el Scan Interval y el Scan Window a un valor igual y fijo de 100 ms, ya que se quiere escanear durante todo el tiempo. El ahorro de energía en la fase de escaneo no es vital. Scan Type se fijará para escaneo pasivo, ya que no se hará uso del SCAN\_REQ.

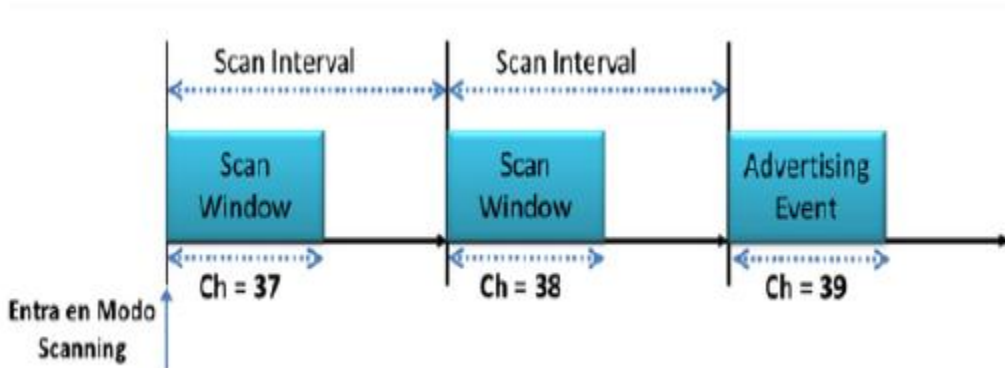


Figura 15. Scanning interval

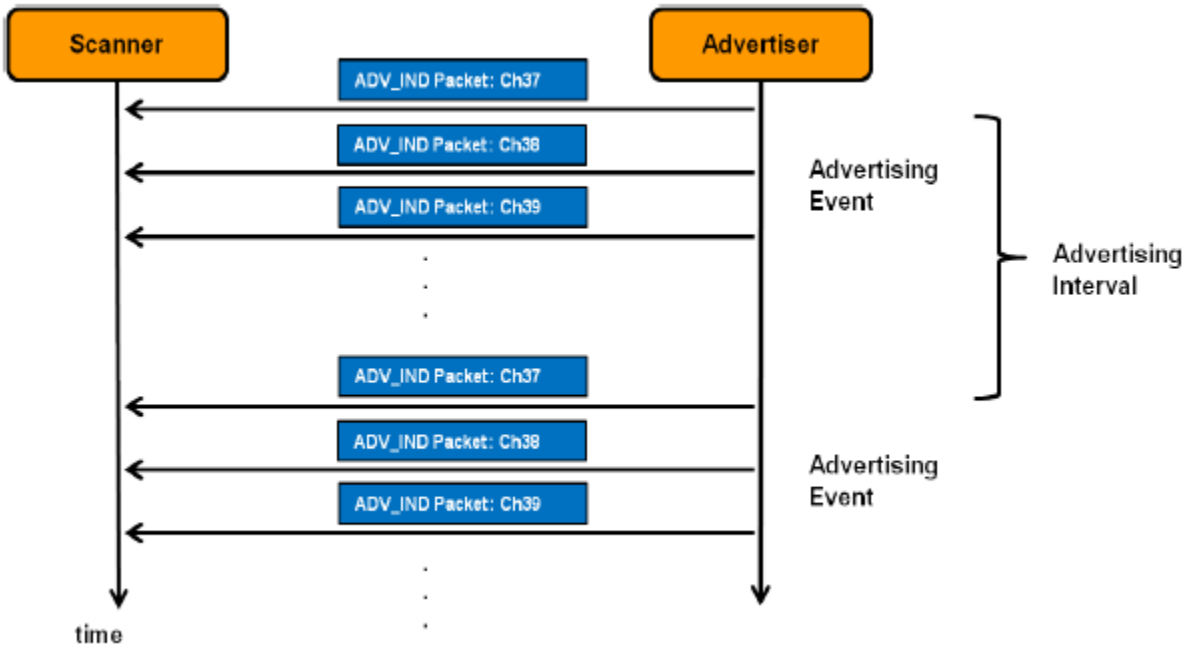


Figura 16. Cronograma envío-recepción advertising

## 2.5. Comandos HCI en Linux para implementar el modo scanner

Este proyecto se ha realizado bajo la plataforma Linux Ubuntu, ya que posee la pila de bluetooth llamada BlueZ pudiendo así enviar órdenes de manera sencilla al hardware bluetooth de la máquina en cuestión. Windows no soporta la librería BlueZ por lo que el dispositivo IoT se basa en Linux.

BlueZ nos permite interactuar con la interfaz HCI (Host controller interface) encargada de transportar los comandos y eventos entre el host y los elementos de control de la pila de protocolos Bluetooth. La capa HCI se implementa a través de un protocolo de transporte como SPI o UART.

De este modo BlueZ ofrece soporte para el núcleo Bluetooth y sus protocolos, siendo flexible, eficiente y sencillo de implementar. Entre sus características encontramos:

- Diseño modular completo
- Soporte para múltiples dispositivos bluetooth.
- Interfaz socket para todas las capas.

Sus principales módulos son:

- Subsistema bluetooth del kernel.
- HCI UART, USB, PCMCIA y drivers para dispositivos virtuales.
- Librerías generales para bluetooth.
- Herramientas de análisis y decodificación del protocolo.

### 2.5.1. Implementación del escáner BLE con BlueZ

Para ejecutar un escaneo BLE pasivo utilizando el dispositivo BLE integrado en nuestro portátil vamos a usar comandos HCI, que como ya hemos dicho, son aquellos que reciben y envían información al hardware BLE. El comando hcitool que enciende el chip bluetooth como escáner BLE es el siguiente.

Command	OCF	Command parameters	Return Parameters
HCI_LE_Set_Scan_Parameters	0x000B	LE_Scan_Type, LE_Scan_Interval, LE_Scan_Window, Own_Address_Type, Scanning_Filter_Policy	Status

```
hcitool -i hci0 cmd 0x08 0x000B 00 10 00 10 00 00 00
```

Figura 17. Comando HCI para escáner BLE.



Siendo "hci0" la interfaz HCI donde se encuentra el adaptador bluetooth a configurar. A partir del OCF 0x000B se configuran los siguientes campos:

- LE\_Scan\_Type (1 octeto):
  - 0x00: Escaneo pasivo, no se envía paquetes SCAN\_REQ.
  - 0x01: Escaneo activo, se pueden enviar paquetes SCAN\_REQ.
- LE\_Scan\_Interval (2 octetos):
  - Rango: 0x0004 a 0x4000.
  - $T = N \cdot 0.625$  siendo N el valor introducido.
- LE\_Scan\_Window (2 octetos):
  - Rango: 0x0004 a 0x4000.
  - $T = N \cdot 0.625$  siendo N el valor introducido. Este valor tiene que ser menor o igual que LE\_Scan\_Interval.
- Own\_Address\_Type: (1 octeto):
  - 0x00: Dirección del dispositivo pública.
  - 0x01: Dirección del dispositivo aleatoria.
- Scanning\_Filter\_Policy: (1 octeto):
  - 0x00: Sin filtro, se aceptan todos los paquetes, menos los paquetes dirigidos a otros dispositivos específicos.
  - 0x01: Se ignoran todos los paquetes de los dispositivos que no están en la lista blanca.

Para iniciar o parar el escaneo usaremos los siguientes comandos:

```
hcitool -i hci0 cmd 0x08 0x000C 00 01
```

*Figura 18. Parar escaneo BLE*

```
hcitool -i hci0 cmd 0x08 0x000C 01 01
```

*Figura 19. Iniciar escaneo BLE*

Command	OCF	Command parameters	Return Parameters
HCI_LE_Set_Scan_Enable	0x000C	LE_Scan_Enable Filter_Duplicates	Status

Figura 20. LE Set Scan Enable Command

A partir del OCF 0x000C que es el LE Set Scan Enable Command se configuran los siguientes campos:

- LE\_Scan\_Enable (1 octeto):
  - 0x00: Parar el escaneo.
  - 0x01: Iniciar el escaneo.
- Filter\_Duplicates:
  - 0x00: Filtro de duplicados desactivado.
  - 0x01: Filtro de duplicados activado.

### 2.5.2. Implementación de escáner BLE con Python

En el caso de este proyecto se ha apostado por python para su desarrollo debido a su gran flexibilidad y sencillez a la hora de programar.

En cuanto a la configuración del chip BLE se refiere, se ha hecho uso de la librería PyBluez, la cual permite ejecutar comandos Bluez de manera sencilla a través de la interfaz de python. A la hora de desarrollar un escáner BLE a través de esta librería se configurarán los distintos parámetros de escaneo del siguiente modo.

```
SCAN_RANDOM = 0x01
OWN_TYPE = SCAN_RANDOM
SCAN_TYPE = 0x01
interval = interval/0.625
window = window_s/0.625

cmd_pkt = struct.pack("<BHBB",SCAN_TYPE,int(interval),int(window),0x00,0x00)
bluez.hci_send_cmd(sock, OGF_LE_CTL,OCF_LE_SET_SCAN_PARAMETERS, cmd_pkt)
```

Figura 21. Configuración de parámetros para escáner BLE

Una vez configurados los parámetros de escaneo y enviados al dispositivo a través del comando HCI correspondiente, se inicia el escaneo BLE del siguiente modo.

```
hci_toggle_le_scan(sock, 0x01)
```

*Figura 22. Inicio de escaneo BLE*

```
hci_toggle_le_scan(sock, 0x00)
```

*Figura 23. Parada de escaneo BLE*

## Capítulo 3. Planificación RF de los beacons mediante la métrica de visibilidad

La arquitectura de localización adoptada, que se explica en profundidad en el capítulo 4, se basa en la técnica del fingerprint donde el dispositivo IoT tiene que recibir las señales de los beacons BLE en cobertura y generar fingerprints, que son vectores de datos con los patrones de la señal. Intuitivamente para que el motor de localización zonal pueda triangular la zona, en una ventana de muestreo se necesita recibir datos de varios beacons. Si en un instante no se reciben datos o sólo se muestrea un beacon, entonces no se podrá estimar la posición o la estimación tendrá una elevada incertidumbre.

La planificación RF de los beacons BLE consiste en saber qué número de beacons son necesarios instalar y en qué ubicación para garantizar que se generan fingerprints con patrones de señales válidos para estimar la zona.

En este capítulo se pretende analizar los factores que afectan a que se reciban datos de un beacon y definir métricas para hacer una primera aproximación al problema de la planificación RF de beacons Bluetooth. No se ha llegado a construir un planificador de beacons y este objetivo queda planteado como una línea de trabajo futuro.

### 3.1. Variables que afectan al canal inalámbrico

En el caso que se nos presenta, también existen una serie de factores que hay que tener en cuenta a la hora de posicionar los distintos beacons en el entorno. Estos son la distancia, los obstáculos a atravesar y la orientación de la persona que porte el dispositivo IoT.

1. **Distancia.** Este factor es común en cualquier comunicación, siendo inversamente proporcional al nivel de señal, es decir, a mayor distancia menor nivel de señal se tendrá en el receptor.

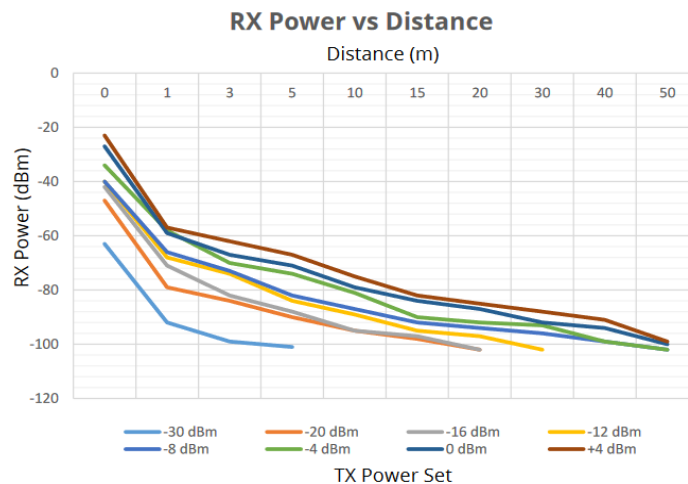


Figura 24. Potencia de recepción frente a distancia. IBKS 105

2. **Obstáculos.** En el entorno de pruebas escogido, se entiende por obstáculos a los distintos muros y puertas que ha de atravesar la señal para llegar a su destino. Además, al ser un entorno dinámico donde conviven distintas personas, la movilidad de estas personas también es considerado un obstáculo, ya que puede entorpecer a la señal en su camino hacia el receptor.
3. **Orientación de la persona.** La orientación de la persona es algo a tener muy en cuenta en este proyecto, ya que será la que llevará el dispositivo IoT encargado de recibir las tramas de los beacons BLE. Según la orientación, la persona portadora del dispositivo IoT puede considerarse un obstáculo, teniendo la señal que atravesar al individuo en cuestión, o puede que no suponga un obstáculo ofreciendo al dispositivo IoT visión directa con el beacon BLE.
4. **Directividad y ganancia.** Una antena es direccional cuando tiene una dirección de máxima radiación en su diagrama. En el caso de las antenas que se utilizan en este proyecto, todas son del tipo omnidireccional, por lo que radian con el mismo nivel de potencia en todas direcciones.

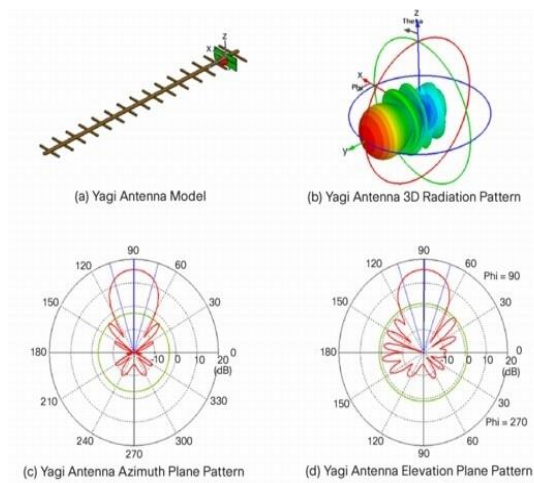


Figura 25. Diagramas de una antena direccional (Yagi)

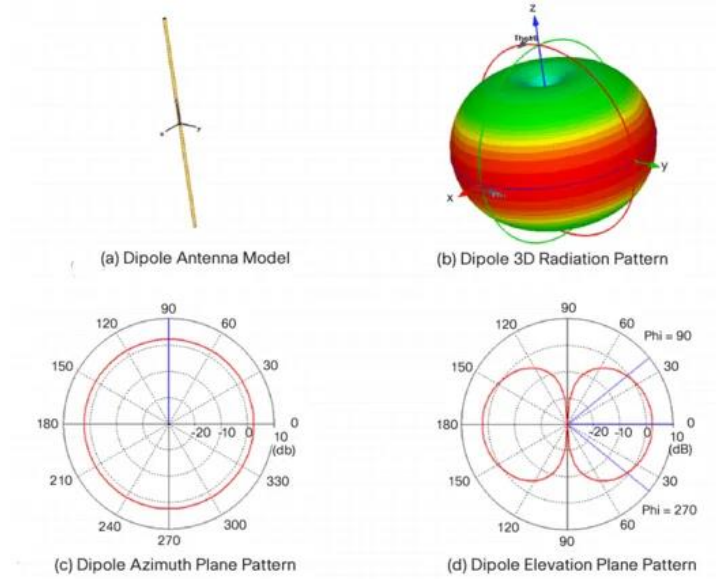


Figura 26. Diagramas de una antena omnidireccional (Dipolo)

La ganancia de una antena se define como el nivel de potencia con la que es capaz de transmitir una señal. A mayor nivel de potencia de transmisión, más sencillo será recibir mensajes en nuestro dispositivo IoT receptor. En el caso de los IBKS, tienen una potencia de transmisión configurable que va desde -30 dBm hasta +4 dBm. Es muy importante escoger un valor correcto de potencia, ya que un valor elevado no siempre es bueno, ya que en BLE prima la duración de la batería de los dispositivos beacon desplegados, pudiendo llegar hasta 46 meses si se configura el menor nivel de potencia.





Slots Enabled	Tx Power (dBm)							
	-30	-20	-16	-12	-8	-4	0	+4
 iBeacon	46	46	45	44	43	41	38	31
 UID Eddystone	45	45	44	43	42	40	37	31
 URL Eddystone	45	45	44	43	42	40	37	31
 EID Eddystone	48	48	47	46	45	44	41	34

Figura 27. Duración de la batería (meses) en función de la ganancia de transmisión

5. **Polarización.** La polarización de una antena es la polarización de la onda radiada por dicha antena en una dirección dada. En un entorno ideal, para que la comunicación se lleve a cabo entre dos antenas, estas deben de tener la misma polarización. En el caso real que se nos presenta, esto no es un factor crítico a tener en cuenta, ya que la mayoría de las veces no se tendrá visión directa y se recibirán mensajes debido al efecto multicamino. Este fenómeno consiste en la recepción de mensajes debido a los constantes rebotes que sufre la señal en el entorno, sufriendo atenuación y cambios de polarización, por lo que la comunicación funcionará incluso con polarizaciones distintas.
6. **Sensibilidad.** La sensibilidad es un concepto de la antena encargada de la recepción, y se define como el mínimo valor de potencia que es capaz de recibir. Esto quiere decir que, a menor nivel de sensibilidad, menor nivel de potencia es capaz de recibir y por lo tanto más amplio será el margen de recepción. Normalmente, los dispositivos BLE tienen una sensibilidad en torno a -100 dBm y no es un valor que pueda configurarse.
7. **Fenómenos de propagación.** Existen otros fenómenos debidos a la propagación de ondas electromagnéticas en el aire, como son la dispersión (fenómeno de scattering), la difracción y el efecto multicamino (multipath). El principal fenómeno que vamos a encontrar en este entorno indoor va a ser el efecto multicamino, el cual se basa en la recepción de ondas electromagnéticas debido a los continuos rebotes de las mismas en el entorno, llegando de esta manera con un nivel de atenuación elevado. El multipath nos permite recibir señal aun cuando no se tiene visión directa entre transmisor y receptor.

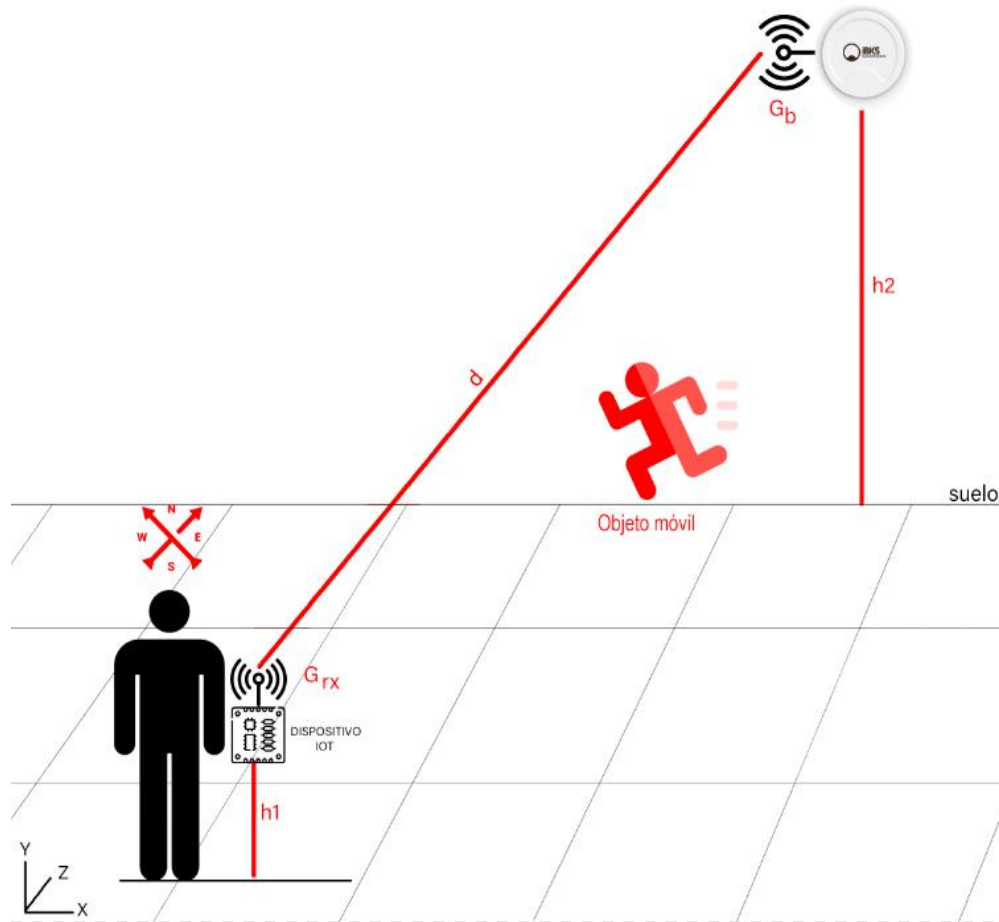


Figura 28. Distintos factores que afectan al nivel de señal

Por último, se han de tener en cuenta otros factores que afectan de manera negativa a la señal BLE, como son las posibles interferencias causadas por las tecnologías existentes en el entorno que trabajan en el mismo rango frecuencial, como es el Wifi. Estas interferencias no afectan a la señal del mismo modo que los factores descritos anteriormente, ya que no atenúan la señal, sino que añaden un nivel de ruido a la misma.

Para entender mejor este aspecto, se ha de conocer el concepto de señal a ruido o SNR. El SNR es un valor que expresa la relación entre la potencia recibida y la potencia de ruido existente, calculándose del siguiente modo:

$$SNR (dB) = 10 \log \frac{P_{señal}(w)}{P_{ruido}(w)}$$

A mayor nivel de SNR, mayor será la calidad de la señal, mientras que a menor SNR, mayor será el ruido de fondo y por tanto peor será la calidad de la comunicación.



Como ya se sabe, la tecnología Wifi trabaja en el mismo rango de frecuencia que BLE, 2.4 GHz, lo que supone un factor de ruido directo al sistema que se presenta en este trabajo. Este ruido no atenuará la señal, pero sí reducirá el margen de recepción, ya que las posibles tramas WiFi en el entorno crearán un determinado nivel de ruido (Noise Floor). Este Noise Floor afecta a la comunicación de modo que, si se recibe un paquete BLE con un nivel de potencia por debajo del nivel de ruido, este no será percibido por el receptor.

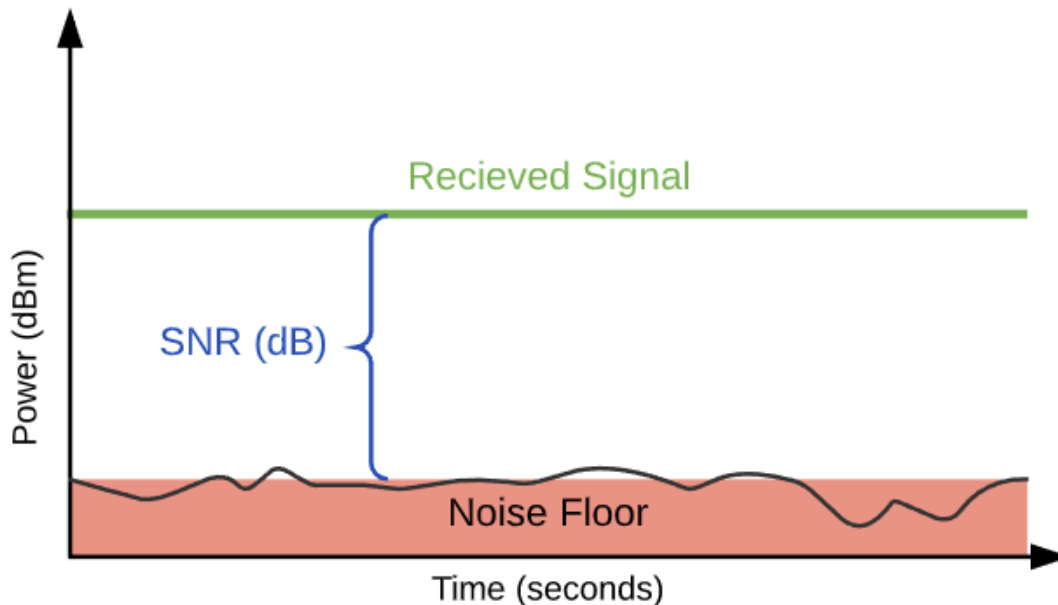


Figura 29. Relación entre SNR y Noise Floor.

### 3.2. Análisis de la capa física mediante el *Packet Reception Rate* y la RSSI

Con el fin de caracterizar los distintos factores cualitativos que afectan a la señal en el trabajo que nos presenta, se propone una serie de pruebas en las que se escoge un único beacon, cuya potencia de transmisión es constante, y se realiza una serie de mediciones variando la distancia, la orientación de la persona, y los obstáculos que atraviesa la señal. Estos tres factores se consideran como los más relevantes en este trabajo, ya que son los que más afectan al nivel de potencia en el receptor.

Para estudiar los resultados obtenidos en esta prueba, se monitorizará dos parámetros que nos mostrarán la calidad de la comunicación.

- **RSSI Medio.** Nivel de potencia medio obtenido en el receptor, medido en decibelios.
- **Packet Reception Rate (PRR).** Este parámetro consiste en la tasa de recepción de paquetes, es decir, el porcentaje de paquetes que se han recibido respecto al teórico.

$$PRR = \frac{\# \text{ Paquetes recibidos}}{N^{\circ} \text{ Paquetes teóricos}} \times 100$$

Para la prueba se toma un beacon, situado en lo alto de la zona 1, y se marcan 7 puntos en línea recta distanciados entre sí con el fin de estudiar la atenuación producida por la distancia entre transmisor y

receptor. En cada uno de estos puntos se realiza una medida de 80 segundos, variando la orientación cada 20 segundos, esto es:

- 20 segundos Norte
- 20 segundos Este
- 20 segundos Sur
- 20 segundos Oeste

Punto	Distancia a beacon
1	1m
2	3m
3	5m
4	7m
5	9m
6	11m
7	13m

*Tabla 4. Distancia (m) desde punto de medición a beacon 1.*

El número de paquetes teórico que se han de recibir para cada una de las orientaciones se calcula de manera sencilla, sabiendo que el periodo de transmisión de un beacon es de 200 ms.

$$\# \text{ Paquetes teóricos (20s)} = \frac{t_{\text{escaneo}}(s)}{T_{tx}(s)} = \frac{20}{0.2} = 100 \text{ mensajes}$$

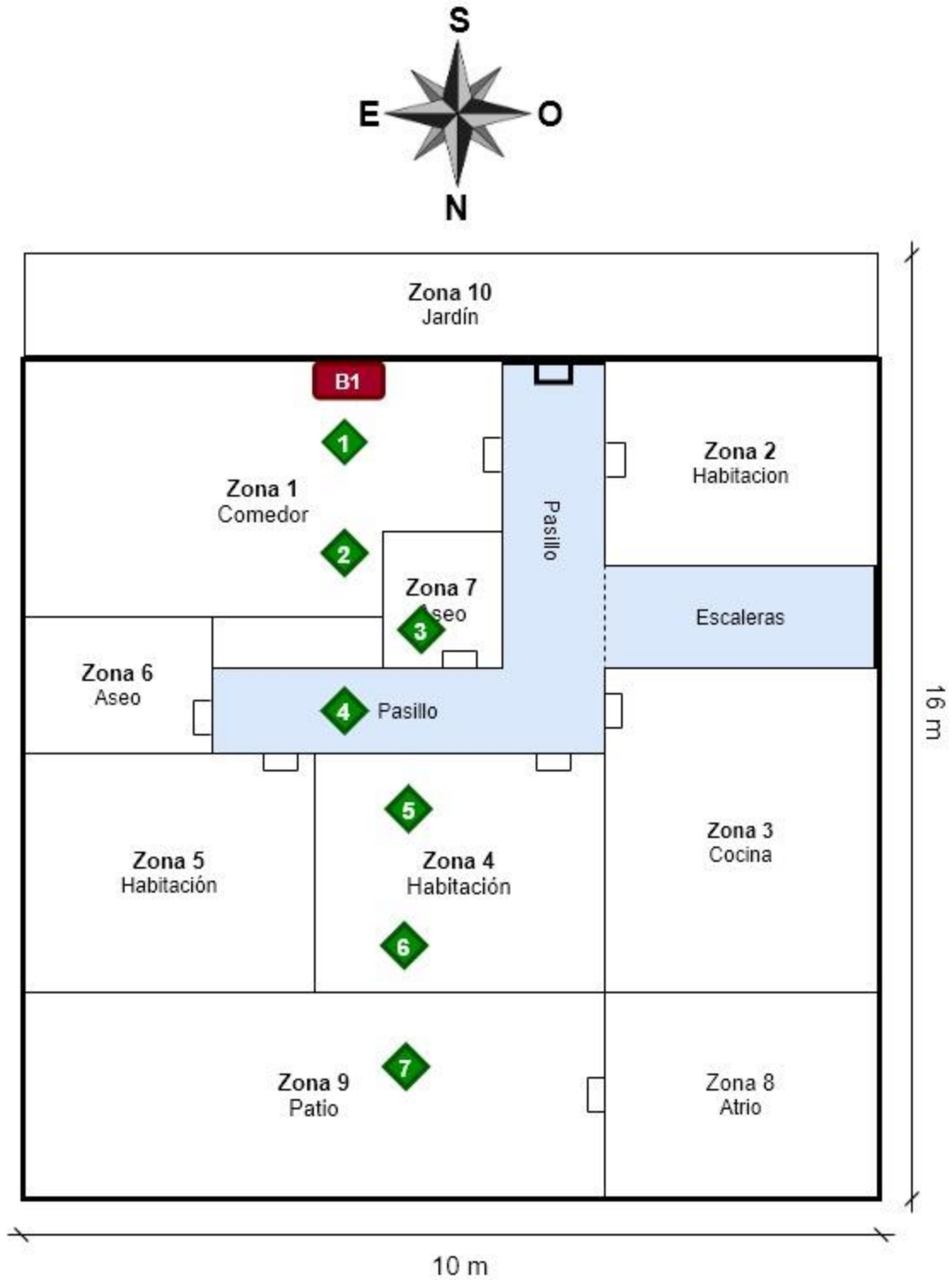


Figura 30. Puntos escogidos y orientación del entorno de pruebas

A continuación, se muestran los resultados obtenidos en la prueba donde se ha obtenido el RSSI medio y la desviación típica para cada uno de los puntos y orientaciones en las que se han capturado los datos del beacon BLE. Cabe destacar que estas pruebas se han realizado con el dispositivo número 1, portátil Dell.

ID	Norte		Este		Sur		Oeste	
	PRR (%)	X/ $\sigma$ (dB)	PRR (%)	X/ $\sigma$ (dB)	PRR (%)	X/ $\sigma$ (dB)	PRR (%)	X/ $\sigma$ (dB)
1	87	-72.1/ 2.6	90	-71.5/ 1.3	93	-62.2/ 0.8	97	-70.9/ 3.9
2	89	-74.3/ 0.8	90	-63.8/ 1.8	91	-68.2/ 1.6	91	-69.4/ 3.0
3	93	-79.6/ 1.1	89	-70.3/ 3.6	86	-74.1 / 1.8	90	-75.1/ 1.7
4	84	-86.3/ 1.9	88	-86.6/ 1.4	86	-82.3/ 1.9	88	-82.6/ 1.5
5	77	-92.9/ 2.0	68	-89.9/ 0.0	77	-89.1/ 1.0	76	-92.4/ 2.4
6	50	-92.7/ 1.2	86	-89.5/ 1.8	85	-91.4/ 1.9	76	-90/ 1.7
7	22	-94.5/ 1.9	0	-100/ 1.6	1	-99.7/ 1.8	32	-95.6/ 1.5

*Tabla 5. PRR, media de RSSI y desviación típica para los puntos tomados.*

Se puede observar de la tabla anterior como influye la orientación de la persona para una medición en el mismo punto. Además, se puede estimar que la atenuación producida por un muro se encuentra en torno a 5 dB, valor significativo suponiendo que en algunos puntos del entorno la señal RF tiene que atravesar hasta 4 muros. Por último, se puede confirmar que el PRR tiene una relación inversamente proporcional a la distancia, es decir, a mayor distancia, menor tasa de recepción se tendrá del dispositivo, tal y como era de esperar.

### 3.3. Definición de la métrica de visibilidad y matriz de visibilidad

Se define la métrica de visibilidad como el porcentaje de mensajes advertising recibidos en relación con el número de mensajes teórico a recibir para cada una de las zonas definidas en el sistema de localización. También se puede definir como la tasa de recepción de paquetes por zona.

Este parámetro nos permite conocer la visibilidad de un beacon en una zona concreta, sabiendo así cuales son los beacons que mayor impacto tienen en el sistema y conocer así con exactitud el número de beacons mínimo que son necesarios para un correcto funcionamiento del sistema.

#### 3.3.1. Planificación RF beacons en entorno real

A la hora de elegir la ubicación de los beacons en el entorno del sistema de localización es necesario tener en cuenta una serie de reglas.

- Los beacons han de colocarse lo más elevado posible. Esto permite que haya una mejor comunicación entre beacon y dispositivo IoT ya que aumenta la posibilidad de visión directa entre ambos dispositivos y elimina el número de obstáculos entre transmisor y receptor. Existen dos tipos de configuración, techo y pared. Para este proyecto se ha escogido la configuración de pared con el fin de facilitar la instalación del sistema.
- Repartir los beacons de manera equidistante por el entorno. Para un correcto funcionamiento del sistema, se ha de tener cobertura BLE en toda el área en el que se despliega el sistema de localización, por esto los beacons han de estar repartidos de manera distante y cubriendo la mayor área posible.
- Evitar obstáculos fijos. Se ha de tener en cuenta a la hora de instalar el beacon que no haya ningún objeto o pared que pueda obstruir la señal de manera considerable.

Para el sistema de localización que se ha desarrollado en este proyecto, se han instalado un total de 6 beacons BLE repartidos de manera equidistante por la casa, utilizando las cajas de las persianas, 2 metros de altura, cumpliendo así con una elevación óptima además de facilitar la instalación de los beacons los cuales vienen con adhesivo.



Figura 31. Alumno realizando pruebas con beacon BLE en verde

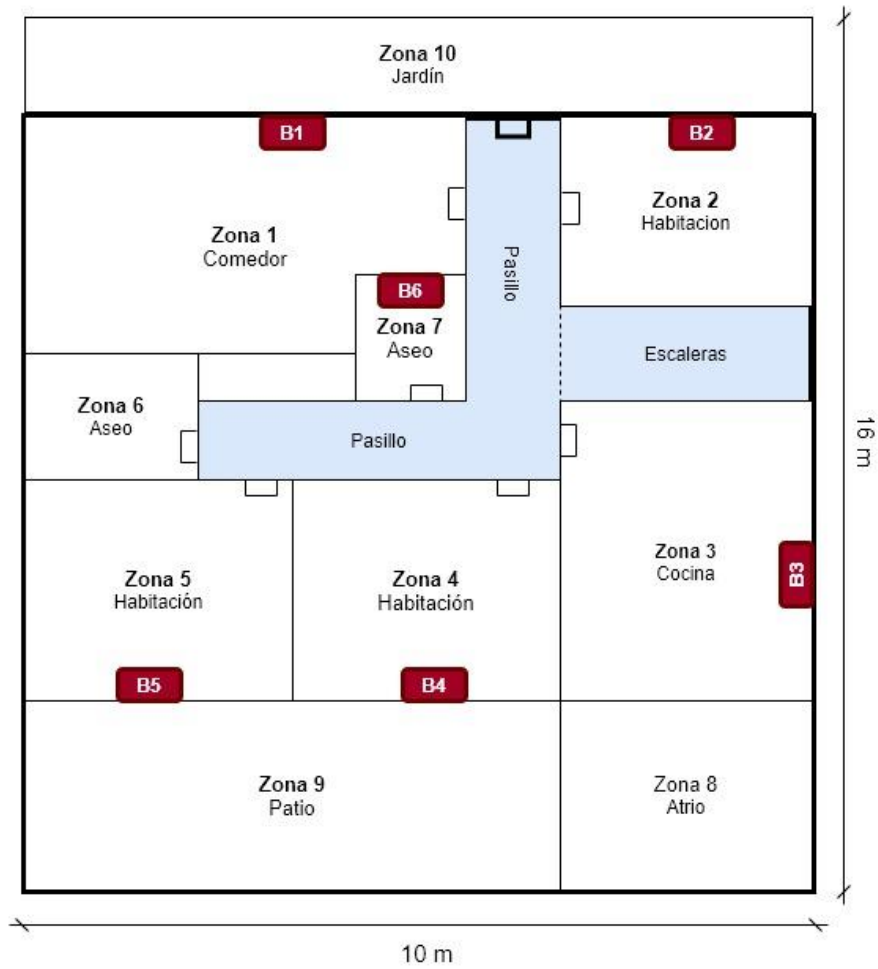


Figura 32. Distribución de beacons sobre escenario

ID BLE \ ID Zona	Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10	Med.
B1	71%	62%	45%	46%	56%	68%	70%	22%	23%	72%	54%
B2	54%	79%	40%	43%	1%	12%	59%	59%	18%	64%	43%
B3	23%	21%	70%	60%	45%	32%	55%	55%	55%	0%	42%
B4	50%	53%	72%	73%	70%	59%	67%	67%	72%	24%	61%
B5	55%	19%	59%	68%	74%	66%	47%	47%	66%	17%	52%
B6	72%	66%	64%	58%	56%	61%	72%	72%	57%	57%	64%
Med.	54%	50%	58%	58%	50%	50%	62%	54%	49%	39%	

*Tabla 6. Matriz de visibilidad obtenida con medición con portátil Dell.*

Observando la matriz de visibilidad de la tabla 3.3, se puede llegar a una serie de conclusiones:

- 1) Atendiendo a la última columna de la matriz de visibilidad, se puede concluir que los beacons 4 y 6 son los que mayor visibilidad presentan en general. Esto es debido a su ubicación centralizada en el entorno de pruebas, ofreciendo así una mejor cobertura para todas las zonas. En cambio, los beacons 2 y 3 son los que peor tasa de recepción presentan debido a que están ubicados en los extremos de la casa, ofreciendo cobertura a un número menor de zonas.
- 2) Se puede afirmar que con un único beacon centralizado en el entorno del sistema de localización, se tiene una cobertura idónea en todas las zonas de la casa. Únicamente sería necesario reforzar las zonas de menor visibilidad con otros dispositivos BLE, pero no se puede conocer con exactitud el número de beacons mínimo necesario en el sistema.
- 3) Además, se pueden concluir los beacons que son visibles en cada una de las zonas, como por ejemplo en la zona 10 donde se puede confirmar que los beacons 3, 4 y 5 tienen una visibilidad reducida, coincidiendo con que son los más alejados a la zona en cuestión.

## Capítulo 4. Arquitectura del servicio de localización zonal

En este capítulo se presentará el sistema propuesto para captar la información necesaria para la localización, la parte que ejecuta dicha localización a través de los datos obtenidos, así como la parte encargada de transmitir la información de localización al usuario final.

Comenzaremos con una visión global del sistema e iremos indagando los distintos subsistemas que lo componen y que hacen que se cumpla el objetivo de localizar un dispositivo a través de BLE.

### 4.1. Diagrama de bloques arquitectura de localización zonal

En la solución propuesta para este trabajo, se ha optado por una arquitectura en la que los beacons BLE permanecen estáticos y donde el dispositivo a localizar se trata de un dispositivo IoT que recibe todas las tramas advertising de los distintos beacons BLE estimando así su ubicación aproximada.

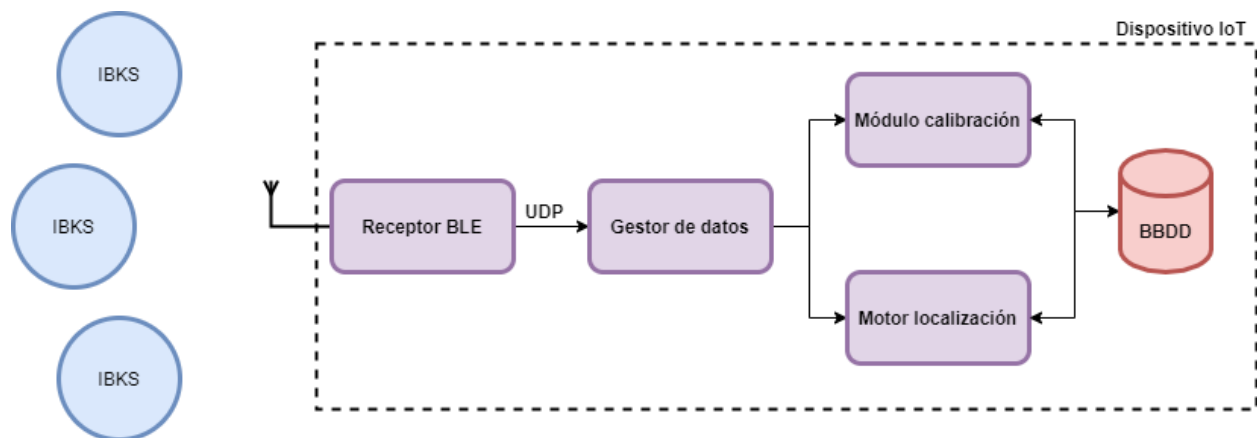


Figura 33. Arquitectura global del sistema de localización

A continuación, se hace una descripción resumida de los principales elementos que componen el sistema:

- **Beacons BLE.** Dispositivos encargados de transmitir periódicamente mensajes de advertising y que servirán de referencia para estimar una localización al estar distribuidos de manera estática por el escenario.
- **Dispositivo IoT.** Dispositivo hardware sobre el que se ejecutan los distintos módulos software para la recepción y tratamiento de datos con el fin de estimar una localización del mismo.



- **Receptor BLE.** Módulo encargado de recibir todos los mensajes BLE existentes en el entorno y realizar un primer filtrado de datos atendiendo únicamente a los beacons BLE pertenecientes al sistema.
- **Gestor de datos.** Este módulo recopilará la gran cantidad de datos generada por el receptor BLE y los resumirá en pequeñas cantidades de significativa información que será enviada con una periodicidad más elevada a través de MQTT hacia el motor de localización.
- **Módulo de calibración.** Como su nombre indica, será el encargado de generar los ficheros de calibración que se tomarán como referencia para obtener una estimación de la localización del dispositivo IoT en cuestión.
- **BBDD.** Almacenamiento del dispositivo IoT donde se almacenarán los ficheros de calibración.
- **Motor KNN.** Por último, tenemos el motor de localización basado en KNN (K-nearest neighbours) que será el encargado de estimar una posición a través de los ficheros de calibración y la información en tiempo real recibida por el gestor de datos.

En cuanto a las tecnologías de desarrollo utilizadas, se han utilizado tanto Python como Matlab. Todos los módulos comentados anteriormente, excepto el motor KNN, se han desarrollado en Python 3.7, aunque es compatible con cualquier versión. En cuanto al motor de KNN, se ha utilizado Matlab 2013 debido a su facilidad a la hora de tratar los datos, así como evaluar el desempeño del sistema en tiempo real.



*Figura 34. Herramientas de desarrollo utilizadas*

Todo esto se ha realizado sobre el sistema operativo Linux Ubuntu 20.04 y siendo compatible con cualquier otro sistema operativo Linux, pero no con Windows debido a las limitaciones que ofrece este sistema en cuanto a comandos HCL.

#### 4.1.1. Beacons BLE

El número y ubicación de Beacons BLE es variable y dependerá de un estudio de planificación RF a medida para el entorno en el que se instale el sistema. Cuanto mayor sea el número de Beacons BLE mejores serán las prestaciones del sistema, pero mayor será su costo de instalación. Es por esto que se ha de elegir un número óptimo de dispositivos BLE atendiendo a una planificación RF previa con el fin de abaratar costes, pero sin poner en compromiso las prestaciones del sistema a la hora de estimar una localización.

En la figura 4.2 podemos observar el despliegue realizado de los distintos Beacons BLE en el entorno, así como la identificación zonal para cada una de las ubicaciones. Los pasillos del escenario no se han considerado como zonas en cuestión ya que son zonas de tránsito donde un usuario no debe permanecer de manera estática durante un tiempo elevado.

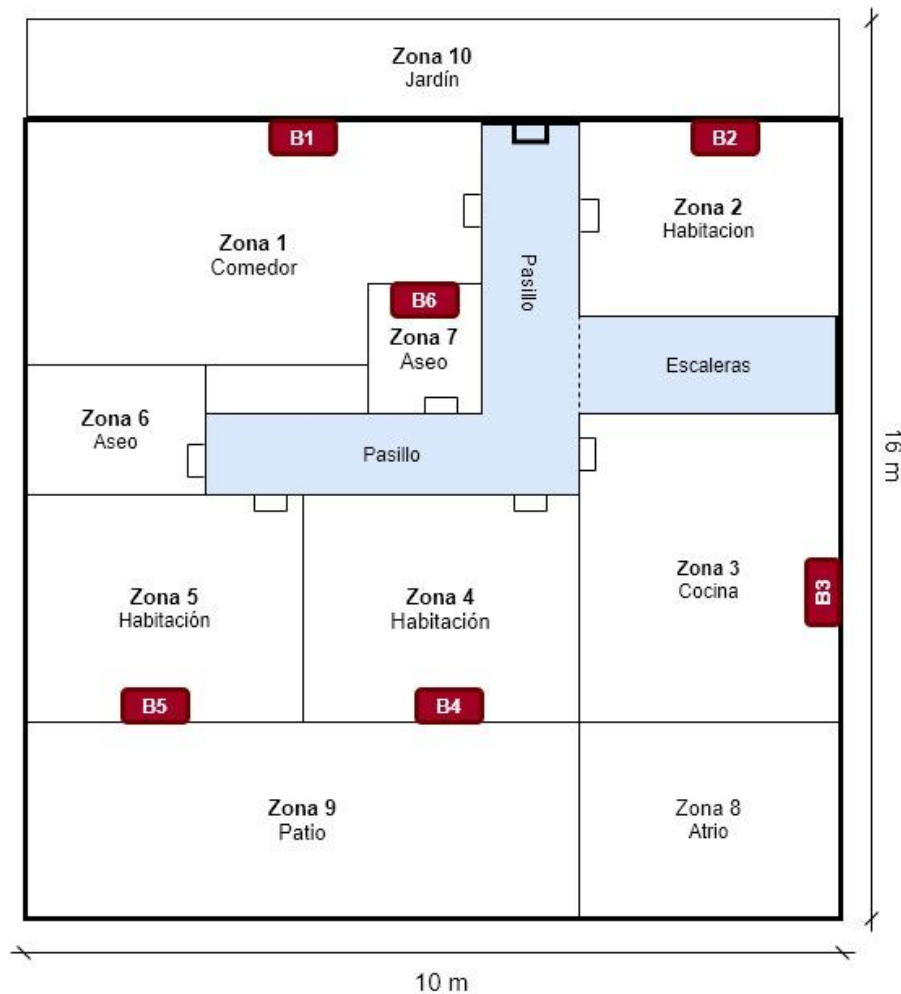


Figura 35. Escenario de implantación del sistema

Se ha escogido el formato iBeacon para la transmisión de los mensajes advertising debido a su extendido uso comercial y fácil configuración. Este formato contiene un campo UUID de 16 Bytes que se usará como filtro del sistema, pudiendo así distinguir los dispositivos pertenecientes al sistema de otros externos que puedan causar interferencias.

Además, se usará el campo Mayor del formato iBeacon, el cual tiene 2 Bytes, para codificar el ID del beacon y poder discernir así el origen de la señal BLE. El otro campo configurable de 2 Bytes, Minor, se reservará para futuras funcionalidades como seguridad, redundancia o funciones de mantenimiento.

Por último, se ha establecido un nivel de potencia igual para todos los beacons del sistema, siendo este un nivel medio ya que entra en juego el compromiso entre cobertura y vida útil de la pila de los beacons.

#### 4.1.2. Dispositivo IoT

Este dispositivo IoT será el objeto a localizar por el sistema desarrollado, siendo cualquier dispositivo hardware con un chip BLE capaz de ejecutar funciones lógicas: raspberry, portátil, arduino, etc.

En el caso que se nos presenta se utilizará un portátil común por su sencillez a la hora de ejecutar distintas tareas en paralelo, y aprovechando su chip BLE integrado para recibir los mensajes advertising del entorno.

#### 4.1.3. Receptor BLE

El módulo de recepción BLE será el encargado de recibir las tramas advertising existentes en el espectro radioeléctrico a través de su interfaz bluetooth, y realizar un primer filtrado atendiendo al UUID de los beacons.

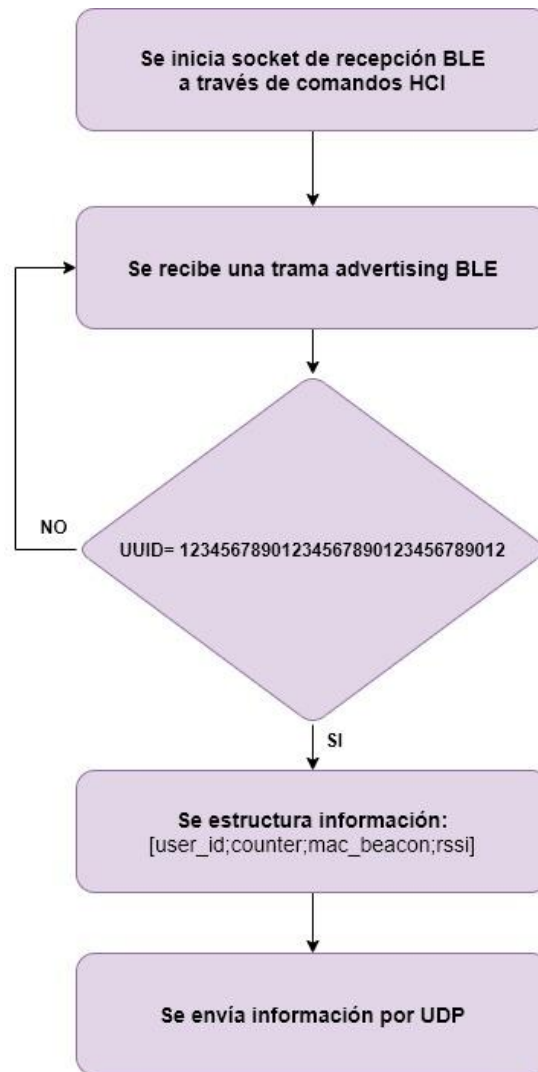


Figura 36. Diagrama de bloques receptor BLE

Por cada trama BLE recibida, se enviará un mensaje UDP con una estructura de datos definida donde se incluye:

- ID del dispositivo IoT que se va a localizar
- Contador de mensajes UDP a modo de seguridad en la comunicación
- Mac del beacon en cuestión
- Identificador del beacon
- Potencia recibida del beacon

Se ha escogido una comunicación UDP debido a su sencillez y capacidad de transmisión, ya que, al no tener ningún control de seguridad en la comunicación, permite enviar pequeñas cantidades de datos con periodos muy bajos.

#### 4.1.4. Gestor de datos

Este módulo es el encargado de recibir y gestionar toda la información proveniente del módulo anterior a través de la comunicación UDP. Recibe la información de la imagen 4.4 como una cadena de caracteres y la trata con el fin de proporcionar información útil en un intervalo temporal más amplio.

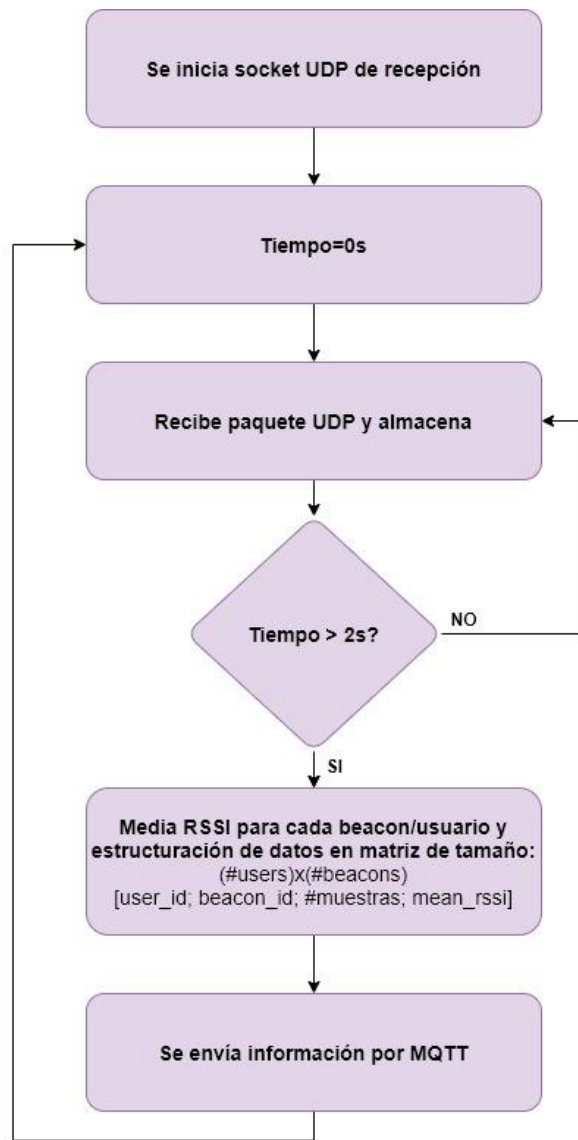


Figura 37. Diagrama de bloques del módulo "Gestor de datos"

Este módulo crea un array de tamaño  $N \times B$ , siendo  $N$  el número de dispositivos IoT existentes en el sistema y  $B$  el número de beacons pertenecientes al mismo, donde en cada fila se almacena la información de interés que será la entrada al motor de localización.

**[IDdispositivo; IDbeacon; ADVcounter; RSSImean]**

Siendo:

- IDdispositivo. ID del dispositivo IoT que realiza la medida en cuestión.
- IDbeacon. ID del beacon el cual se ha recibido la trama advertising correspondiente.
- ADVcounter. Contador de tramas advertising recibidas para el beacon anterior.
- RSSImean. Media del valor de RSSI realizada con cada una de las potencias recibidas.

Esta matriz se va rellenando con cada uno de los mensajes UDP recibidos y durante un intervalo temporal de 2 segundos, pasado este tiempo se envía a través de MQTT con el topic "BLELOC" hacia el broker y se resetean sus valores para volver a comenzar. Esto permite tener un valor significativo de potencia para cada uno de los beacons realizando la media de estos valores obtenidos en intervalos de 2 segundos. Además, el contador de tramas advertising, nos indica la validez del valor de RSSI medio, ya que, si este es muy bajo, la media realizada no tendrá demasiada fiabilidad.

Un aspecto importante de este módulo es que, si no se recibe ninguna señal de un beacon, el valor de potencia que se incluye en la matriz a enviar es de -100 dBm, ya que, si se deja el valor de 0 dBm, el motor de localización entendería que está justo al lado de ese beacon, siendo todo lo contrario. Se ha escogido el valor de -100 dBm debido a que la sensibilidad de los dispositivos BLE se mueven en torno a este valor, indicando así al sistema de localización que ese beacon se encuentra en el límite de cobertura.

#### 4.1.5. Calibración

El módulo de calibración es una parte fundamental del sistema, ya que nos permite generar los fingerprints de calibración, a partir de los cuales el sistema de localización tomará una decisión de donde se encuentra el dispositivo a localizar. Una buena calibración se traducirá en una buena resolución a la hora de estimar una posición.

La calibración que se presenta en este trabajo, se ha realizado con el objetivo de ser lo más sencilla posible y con el fin de que cualquier usuario pueda realizarla de manera rápida y sin dificultad alguna.

Este módulo recibirá continuamente los mensajes MQTT enviados por el gestor de datos para su posterior tratamiento y generación del fingerprint de datos. Para ello se ha de indicar como parámetro de entrada, el dispositivo IoT que va a realizar la calibración, filtrando así los valores de potencia recibidos por ese dispositivo y empaquetando estos valores de RSSI en el fingerprint.

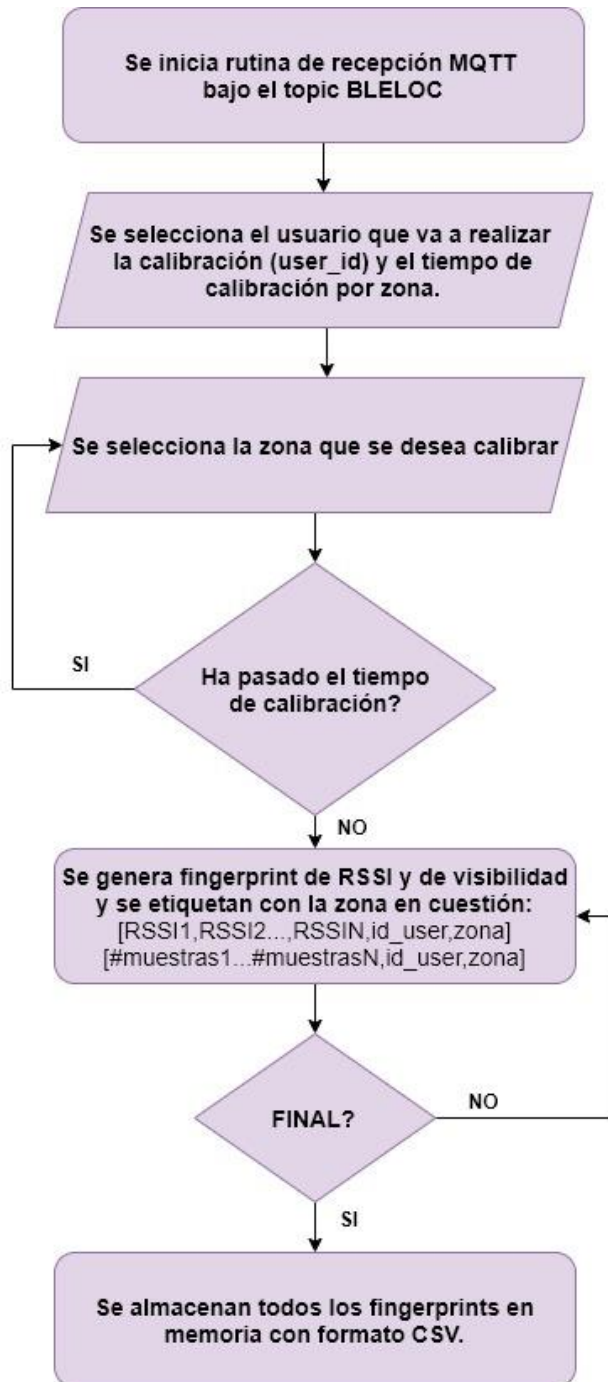


Figura 38. Diagrama de bloques de la calibración

Además, para la calibración el usuario ha de ir desplazándose naturalmente por cada zona durante un tiempo de un minuto, indicando en cada intervalo de tiempo la zona en la que se encuentra para poder etiquetar así el fingerprint de calibración.

[ RSSI\_B1, RSSI\_B2, RSSI\_B3 ... RSSI\_Bn, IDdispositivo, IDzona]

Siendo:

- RSSI\_Bx. Valor medio de RSSI del beacon X.
- IDdispositivo. ID del dispositivo IoT que está realizando la calibración
- IDzona. ID de la zona donde se encuentra el usuario.

Una vez recorridas todas las zonas y dada por terminada la calibración, se genera un fichero CSV con todos los fingerprints de calibración obtenidos y se almacena en memoria.

RSSI_B1	RSSI_B2	RSSI_B3	RSSI_B4	RSSI_B5	RSSI_B6	IDdispositivo	Idzona
-92	-84	-76	-62	-79	-78	1	4
-88	-88	-78	-63	-79	-80	1	4
-87	-90	-77	-63	-77	-77	1	4
-90	-97	-76	-56	-72	-84	1	4
-90	-91	-78	-57	-70	-85	1	4
-88	-91	-79	-57	-79	-78	1	4
-85	-89	-81	-62	-76	-78	1	4
-92	-88	-77	-65	-68	-74	1	4
-89	-88	-72	-69	-67	-77	1	4
-89	-89	-76	-67	-71	-81	1	4
-91	-89	-79	-69	-67	-80	1	4
-85	-92	-83	-67	-72	-78	1	4
-88	-94	-87	-58	-70	-80	1	4
-89	-95	-77	-58	-70	-83	1	4
-88	-93	-83	-57	-68	-83	1	4
-92	-89	-81	-56	-72	-82	1	4
-93	-92	-80	-60	-73	-80	1	4
-88	-91	-81	-63	-66	-83	1	4

Tabla 7. Fingerprint de calibración generado

Además del fichero de calibración, se genera el fingerprint de visibilidad, que consiste en la misma idea que el de calibración, pero con el número de muestras BLE recibidas por cada uno de los beacons. Este fichero nos sirve de ayuda a la hora de conocer el correcto funcionamiento del sistema y se podría utilizar como fuente adicional de datos a la hora de estimar una posición, aunque su uso actual es para depurar el sistema.



#beacon1	#beacon2	#beacon3	#beacon4	#beacon5	#beacon6	lddisposi	ldzona	
5	4	7	6	8	5	1	4	
2	0	5	6	8	4	1	4	
4	2	7	6	7	6	1	4	
2	5	8	8	8	6	1	4	
6	5	6	4	6	6	1	4	
6	2	4	9	7	4	1	4	
3	6	5	9	8	3	1	4	
6	5	5	8	6	7	1	4	
8	8	7	9	5	7	1	4	
6	9	6	6	7	6	1	4	
3	6	5	6	8	3	1	4	
6	8	4	9	4	5	1	4	
3	6	9	9	6	6	1	4	
2	4	7	9	5	7	1	4	
5	7	5	8	7	6	1	4	

Tabla 8. Fingerprint de visibilidad

## 4.2. Motor de localización basado en kNN

Este módulo es el principal objeto de este trabajo, el cual estima la posición de un dispositivo IoT basándose en un clasificador KNN, K-nearest neighbours. KNN es un algoritmo de aprendizaje supervisado, lo que quiere decir que a partir de un dataset inicial de datos su función será clasificar correctamente todas las nuevas instancias de entrada.

La idea de este algoritmo es realmente sencilla, estimará una posición para cada entrada de datos nueva según tenga k vecinos más cerca de una zona o de otra. Es decir, calculará la distancia del elemento nuevo a cada uno de los existentes, y ordenará dichas distancias de menor a mayor para ir seleccionando la zona a la que pertenece.

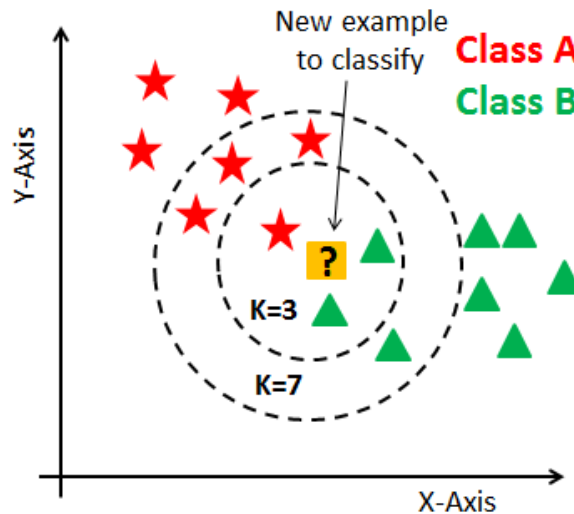


Figura 39. Clasificador KNN con dos clases

En contraste con otros algoritmos de aprendizaje supervisado, KNN no genera un modelo fruto del aprendizaje con datos de entrenamiento, sino que el aprendizaje sucede en el mismo momento en el que se prueban los datos de test. A este tipo de algoritmos se les llama *lazy learning methods*.

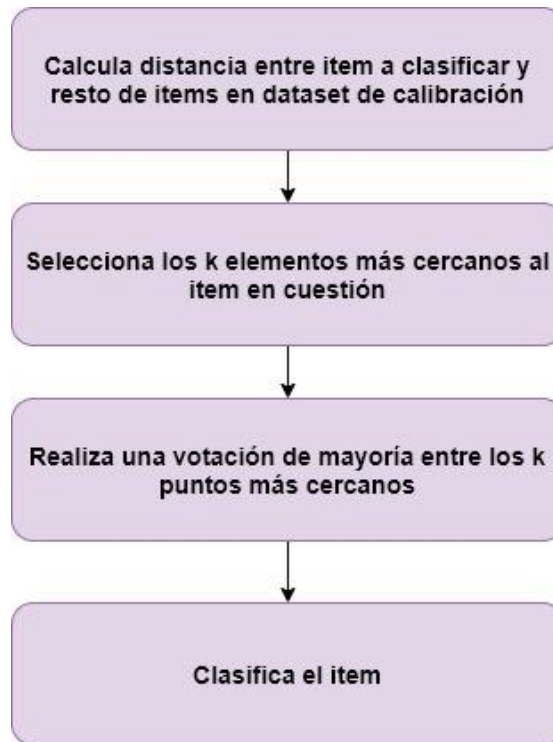


Figura 40. Funcionamiento del algoritmo KNN

En el caso que se nos presenta, se ha desarrollado un algoritmo KNN propio, con el fin de evaluar las distintas configuraciones del motor KNN de manera ilimitada y pudiendo así establecer una configuración óptima para el sistema de localización que se plantea en este trabajo.

El clasificador KNN que se ha realizado para este trabajo calculará la distancia entre dos fingerprints de potencias a través de la distancia euclídea y decidirá una zona atendiendo a los cinco ( $K=5$ ) vecinos más cercanos. Esos 5 vecinos tendrán la misma importancia de voto, por lo que la zona que más se repita (moda) será la estimada por el clasificador.

Para que el valor de  $K=5$  sea óptimo, se ha de tener un buen número de fingerprints de calibración, ya que si este es reducido la resolución del clasificador se vería gravemente afectada. Es por esto que se recomienda una calibración mínima de 2 minutos por zona, produciendo así alrededor de 60 fingerprints de calibración para cada una de las zonas implicadas en el sistema.

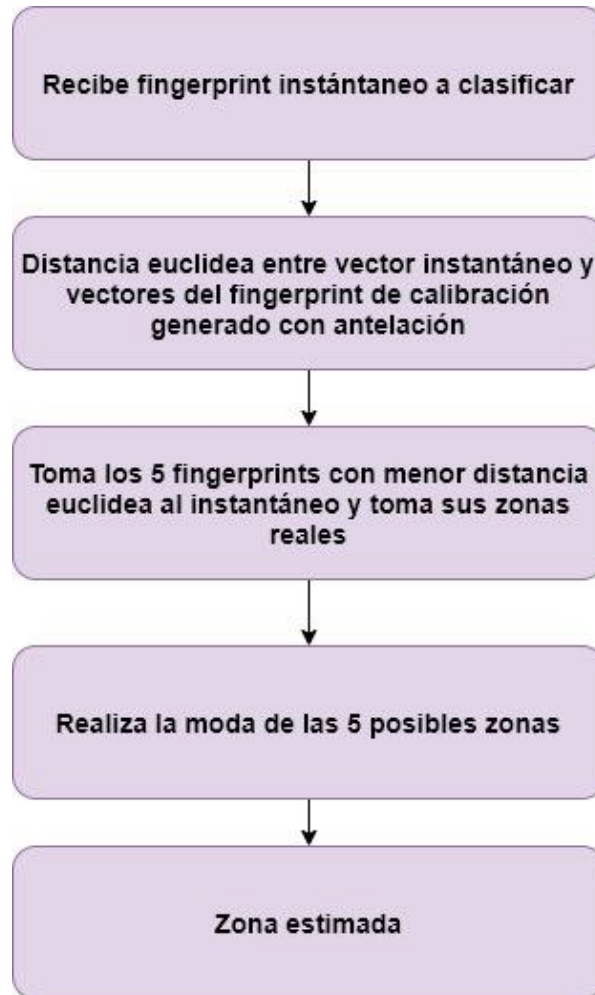


Figura 41. Diagrama de flujo de motor KNN

#### 4.2.1. Modo de operación online del motor KNN

El modo de operación online es el modo normal de funcionamiento del motor de localización, es decir, que a partir de un fingerprint recibido en tiempo real se estime una posición de manera instantánea.

Para ello se utiliza una rutina de recepción a través del protocolo MQTT, la cual está suscrita a los eventos BLE para recibir todos los mensajes del gestor de datos, a partir de los cuales se generará el fingerprint instantáneo y ejecutará el motor KNN comparando este vector con todos los fingerprints de calibración.

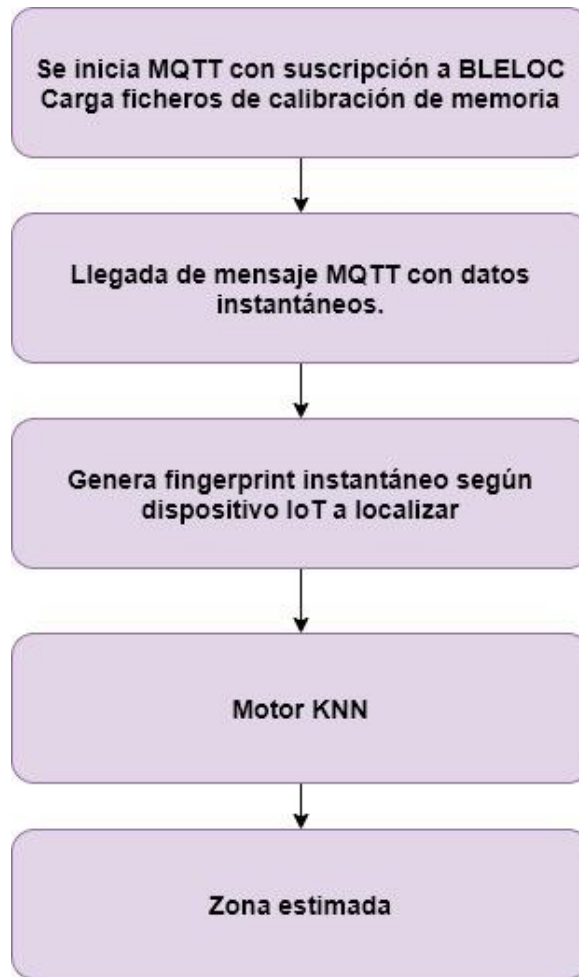


Figura 42. Diagrama de bloques de motor KNN online

#### 4.2.2. Modo de operación offline y evaluación del desempeño del sistema

En cuanto al modo offline de funcionamiento, su principal función es la de poder estudiar el desempeño del sistema y poder probar distintas configuraciones del motor de localización de una manera sencilla.

Al contrario que modo online, esta no tiene ninguna rutina de recepción de datos, sino que todas sus entradas son tomadas a partir de la memoria del dispositivo. Se distinguen dos ficheros de entrada a este módulo, los datos de entrenamiento y los datos de prueba.

- Datos de entrenamiento o calibración. Son aquellos datos a partir de los cuales el clasificador va a estimar una posición.
- Datos de prueba. Conjunto de datos etiquetados que serán clasificados con el fin de evaluar el desempeño del sistema.

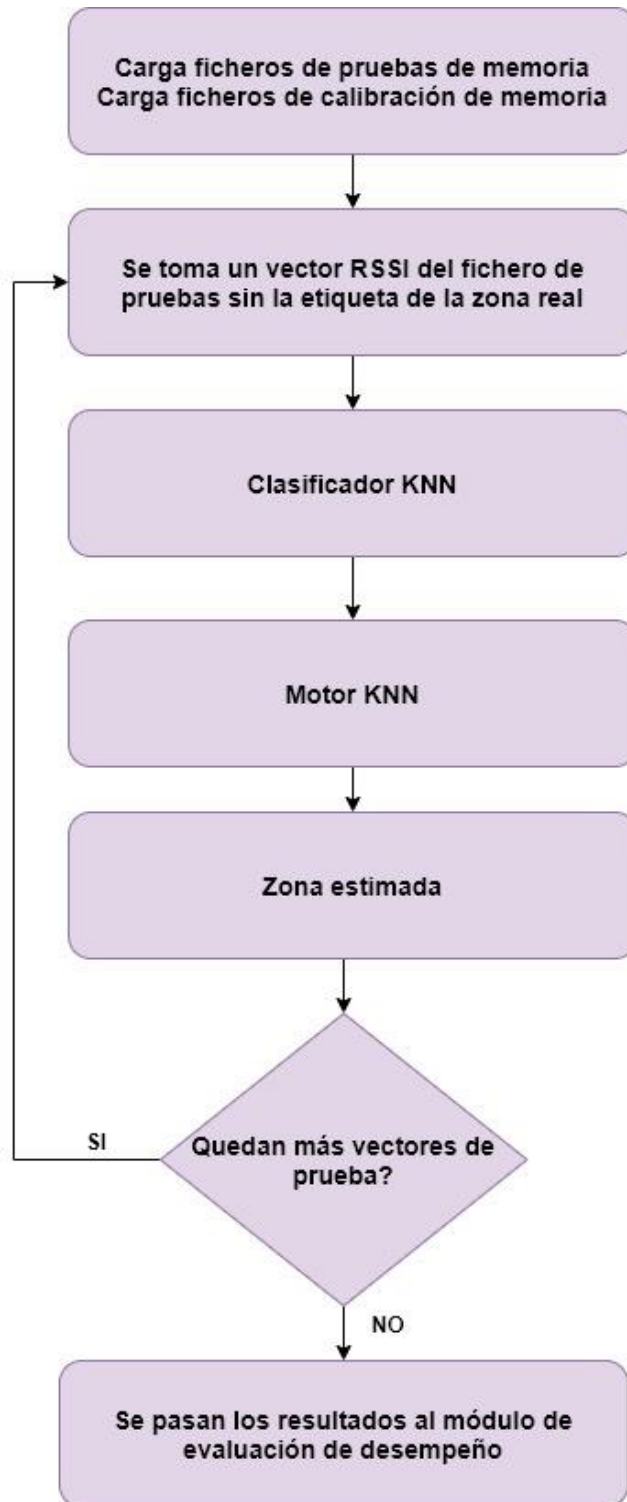


Figura 43. Diagrama de flujo de motor offline

El modo de funcionamiento offline nos proporcionará un fichero de salida con las estadísticas necesarias para evaluar el desempeño del sistema, como son el accuracy global y zonal. Además, este fichero de salida tendrá presente la fecha y hora a la que se ha realizado la prueba y el dispositivo IoT con el que se ha realizado con el fin de poder evaluar la diversidad en el tiempo y en la electrónica del hardware.

Estas estadísticas se obtendrán del módulo de evaluación del desempeño, el cual toma las zonas estimadas para cada uno de los fingerprint del fichero de pruebas y las compara con las zonas reales que contiene el mismo fichero, pudiendo así calcular el accuracy de manera sencilla.

FECHA	HORA	IDDISP	ACCGLC	ACCZON	ACCZON	ACCZON	ACCZON	ACCZON	ACCZON	ACCZON	ACCZON	ACCZON	ACCZON	ACCZON
08/10/2020	18:57:00	1	98%	100%	100%	100%	100%	100%	100%	100%	100%	83%	100%	100%
08/11/2020	14:55:00	1	94%	90%	100%	100%	100%	100%	100%	97%	97%	73%	97%	93%
08/17/2020	14:01:00	1	97%	100%	100%	100%	93%	93%	100%	100%	97%	100%	100%	100%
08/20/2020	15:19:00	1	97%	100%	100%	97%	97%	100%	100%	100%	97%	100%	100%	100%
08/10/2020	18:23:00	2	82%	65%	65%	72%	77%	85%	92%	97%	72%	73%	93%	
08/10/2020	18:57:00	2	87%	100%	97%	90%	60%	97%	90%	97%	77%	90%	100%	
08/11/2020	14:53:00	2	89%	87%	100%	93%	63%	87%	97%	100%	90%	80%	100%	
08/17/2020	14:01:00	2	88%	90%	97%	83%	63%	93%	93%	83%	83%	90%	90%	
08/20/2020	15:20:00	2	84%	77%	100%	87%	43%	83%	87%	100%	70%	90%	100%	

Tabla 9. Fichero de desempeño generado

## Capítulo 5. Resultados y conclusiones

Con el fin de probar el correcto funcionamiento del sistema desarrollado en este proyecto, se propone realizar una serie de medidas que generen datos ya etiquetados que permitan comprobar la robustez del sistema frente a diversos parámetros.

- **Robustez en el tiempo.** El espectro radioeléctrico varía durante el paso del tiempo, también varía la movilidad de posibles individuos existentes en el entorno, así como obstáculos. Todo esto puede afectar al patrón de la señal, por lo que se comprobará la robustez frente al paso del tiempo.
- **Robustez frente a distinto hardware.** El uso de otro hardware distinto al que ha realizado la calibración, es un factor muy determinante a la hora de estimar una posición, ya que disponen de distintos chips BLE con distintas antenas. Se comprobará la robustez frente a este fenómeno utilizando un portátil y una Raspberry 4B.

Además, para poder comparar los resultados de manera correcta, se toman como constantes los siguientes parámetros del sistema de localización.

- Beacons IBKS.
  - Periodo de advertising de los beacons IBKS de 200 ms.
  - Potencia de transmisión configurada en los beacons IBKS de -4 dBm.
  - Se despliega un total de 6 beacons en el escenario de pruebas.
- Tiempo de promediado. El tiempo de adquisición de todas las tramas ADV recibidas es de 2 segundos, es decir, cada 2 segundos se genera un vector de RSSI medios.
- Peso beacons no detectados. Cuando un beacon no se detecta, se le da un peso de -100 dBm.
- Motor KNN.
  - Se calculan los k vecinos más cercanos a través de la distancia euclídea.
  - Se atienden los 5 vecinos más cercanos,  $k=5$ .

En cuanto al hardware para la toma de datos, se utilizarán dos dispositivos distintos. Un portátil Dell con su chip BLE integrado y una Raspberry Pi 4B, también con su módulo BLE integrado.



*Figura 44. Dispositivos de medición utilizados.*

Las pruebas se realizarán siguiendo un movimiento natural y pausado por cada una de las zonas y etiquetando el vector de datos con la zona en la que se encuentra el individuo en ese instante.

Además, el tiempo establecido de captura de datos para cada una de las zonas será de 1 minuto, tiempo suficiente para obtener una cantidad de fingerprints considerables por zona. Para el fingerprint de calibración, el cual se obtiene utilizando el portátil, se toma un intervalo de 2 minutos por zona, teniendo así una mayor cantidad de datos para el fichero de calibración, lo que se traduce en una mayor resolución del sistema.

Una vez que se han realizado las pruebas, se calcula una serie de métricas de desempeño que permiten conocer el grado de acierto y robustez, tanto global como zonal, del sistema.

- **Confusion matrix.** (Matriz de confusión) Herramienta que permite la visualización del desempeño de un algoritmo que emplea aprendizaje supervisado. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias de la clase real.
- **Accuracy.** (Exactitud) Este parámetro se calcula como el número total de aciertos, entre el número total de fallos. Sirve para expresar el desempeño total del sistema de manera global. También se le conoce como exactitud, y responde a la pregunta ¿Qué porcentaje de predicciones fue correcta?



- **Precisión.** La precisión es la fracción de todas las instancias relevantes dividido entre las instancias obtenidas. En el caso que se nos presenta, este parámetro responde a la pregunta, ¿Qué porcentaje de predicciones, cuya zona real era la 1, fueron correctas?
- **Recall.** (Exhaustividad o sensibilidad) La sensibilidad es la fracción de instancias relevantes que se han obtenido sobre la cantidad total de instancias relevantes. En nuestro caso, esta métrica responde a la pregunta, ¿Qué porcentaje de predicciones, cuya estimación fue la zona 1, fueron correctas?
- **F1.** Esta métrica nos resume la precisión y sensibilidad en un sólo valor, por ello es de gran utilidad cuando la distribución de clases es desigual. Se calcula como el doble del producto de la precisión por la sensibilidad, partido de la suma de la sensibilidad y precisión. Este parámetro es muy útil a la hora de estudiar el rendimiento de un clasificador cuyas instancias están desbalanceadas, como en nuestro caso esto no es así, no será utilizado en el proyecto.

Estas métricas (Accuracy, Precision, Recall y F1) son muy utilizadas a la hora de analizar resultados para una máquina de aprendizaje supervisado, como es el caso que se nos presenta. Atendiendo a estos parámetros se puede conocer de manera fiable la robustez del sistema, así como las zonas de peor funcionamiento para una posible reestructuración de la arquitectura de beacons.

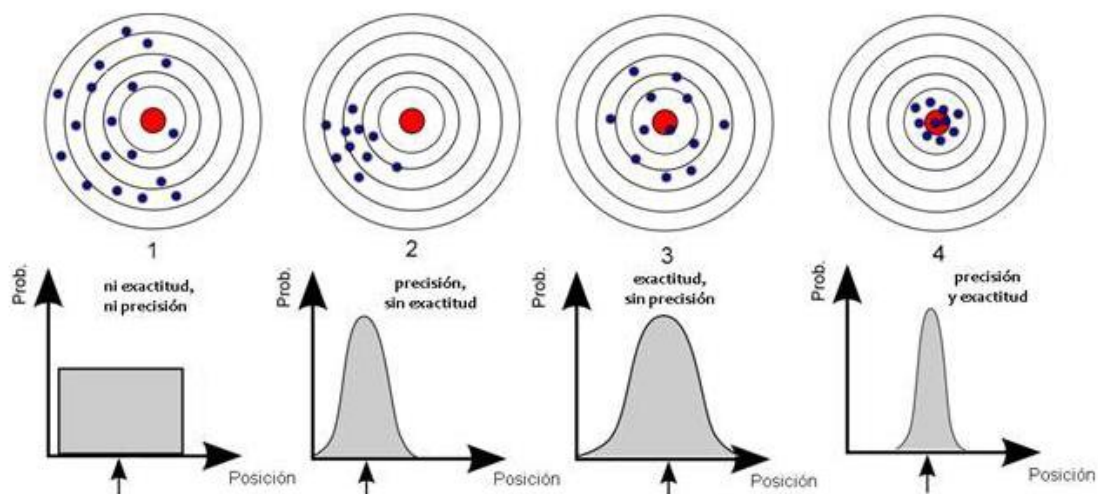


Figura 45. Compromiso de precisión y exactitud

## 5.1. Evaluación del desempeño de la estimación de la zona real

La calibración del sistema se llevó a cabo con el portátil Dell el día 10 de Agosto de 2020, a las 18:00 de la tarde, generando un fingerprint de calibración de referencia para el motor de localización. Para ello se realizó un movimiento natural sobre el entorno de pruebas, permaneciendo un tiempo mínimo de 2

minutos en cada una de las zonas y etiquetando cada fingerprint con el ID de la zona en la que se encontraba el dispositivo.

Se estudió la posibilidad de realizar otro tipo de movimiento a la hora de calibrar el sistema, como puede ser la metodología centroide, que consiste en permanecer estático en el centro de la zona durante el periodo de calibración. Finalmente, se decidió realizar un movimiento natural ya que de este modo se tienen en cuenta los distintos factores que pueden entorpecer la señal RF, como son la orientación del dispositivo, la distancia y la variabilidad del espectro frecuencial.

Además, de este modo se obtienen puntos de calibración sobre toda la superficie de la zona en cuestión, en cambio, con la metodología del centroide únicamente se obtienen puntos de calibración en el centro de la zona, lo que puede resultar en una menor resolución a la hora de estimar una posición.

Para la evaluación del sistema se ha decidido realizar sesiones de test en días diferentes al de calibración, pudiendo así observar la posible variación del espectro radioeléctrico en función del día de la semana. Se etiquetan todos los datos de test para tener una certeza de la zona real (*ground truth*).

Una sesión de test emula el procedimiento de calibración y se recorren todas las zonas siguiendo un movimiento natural durante al menos un minuto por zona. Tomando una zona por minuto, siendo 10 zonas, se obtienen en torno a unos 300 fingerprints, una cantidad de datos razonable para poder obtener estadísticas con un grado de confianza elevado.

En cada sesión de test se generan fingerprints con dos tipos de dispositivos:

- **Portátil Dell latitude.** ID Dispositivo 1
- **Raspberry Pi 4B.** ID Dispositivo 2

Ambos dispositivos hacen uso de su chip BLE integrado. De este modo se reduce el tiempo de testeo considerablemente. Como promedio, se tardan 15 minutos para una sesión de test de 10 zonas (160 m<sup>2</sup>), un tiempo bastante breve para la calibración de un sistema de localización.

A continuación, se muestran los resultados obtenidos para las medidas tomadas durante un periodo de 5 días.

Día	Fecha	Hora	Tipo
0	09/08/2020	18:23	CALIBRACIÓN
1	10/08/2020	18:57	TEST
2	11/08/2020	14:55	TEST
3	17/08/2020	14:01	TEST
4	20/08/2020	15:20	TEST

Tabla 10. Días de testeo y calibración

5.1.1. Confusion matrix de las sesiones de test por cada dispositivo.

		ZONA PREDICHA									
		Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10
ZONA REAL	Z1	30	0	0	0	0	0	0	0	0	0
	Z2	0	30	0	0	0	0	0	0	0	0
	Z3	0	0	30	0	0	0	0	0	0	0
	Z4	0	0	0	30	0	0	0	0	0	0
	Z5	0	0	0	0	30	0	0	0	0	0
	Z6	0	0	0	0	0	30	0	0	0	0
	Z7	0	0	0	0	0	0	30	0	0	0
	Z8	0	0	1	0	0	0	0	25	4	0
	Z9	0	0	0	0	0	0	0	0	30	0
	Z10	0	0	0	0	0	0	0	0	0	30

Tabla 11 . Confusion matrix día 1, dispositivo 1

		ZONA PREDICHA									
		Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10
ZONA REAL	Z1	27	0	0	0	0	0	3	0	0	0
	Z2	0	30	0	0	0	0	0	0	0	0
	Z3	0	0	30	0	0	0	0	0	0	0
	Z4	0	0	0	30	0	0	0	0	0	0
	Z5	0	0	0	0	30	0	0	0	0	0
	Z6	0	0	0	0	1	30	0	0	0	0
	Z7	1	0	0	0	0	0	29	0	0	0
	Z8	0	0	0	0	0	0	0	22	4	0
	Z9	0	0	0	0	1	0	0	0	29	0
	Z10	2	0	0	0	0	0	0	0	0	28

Tabla 12. Confusion matrix día 2, dispositivo 1

		ZONA PREDICHA									
		Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10
ZONA REAL	Z1	30	0	0	0	0	0	0	0	0	0
	Z2	0	30	0	0	0	0	0	0	0	0
	Z3	0	0	30	0	0	0	0	0	0	0
	Z4	0	0	2	28	0	0	0	0	0	0
	Z5	0	0	0	2	28	0	0	0	0	0
	Z6	0	0	0	0	0	30	0	0	0	0
	Z7	0	0	0	0	0	0	30	0	0	0
	Z8	0	0	0	0	0	0	0	29	1	0
	Z9	0	0	0	0	0	0	0	0	30	0
	Z10	0	0	0	0	0	0	0	0	0	30

Tabla 13. Confusion matrix día 3, dispositivo 1

		ZONA PREDICHA									
		Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10
ZONA REAL	Z1	30	0	0	0	0	0	0	0	0	0
	Z2	0	30	0	0	0	0	0	0	0	0
	Z3	0	0	29	1	0	0	0	0	0	0
	Z4	0	0	1	29	0	0	0	0	0	0
	Z5	0	0	0	0	30	0	0	0	0	0
	Z6	0	0	0	0	0	30	0	0	0	0
	Z7	0	0	0	0	0	0	30	0	0	0
	Z8	0	0	0	0	0	0	0	29	1	0
	Z9	0	0	0	0	0	0	0	0	30	0
	Z10	0	0	0	0	0	0	0	0	0	30

Tabla 14. Confusion matrix día 4, dispositivo 1

		ZONA PREDICHA									
		Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10
ZONA REAL	Z1	30	0	0	0	0	0	0	0	0	0
	Z2	0	29	0	0	0	0	0	0	0	1
	Z3	0	0	27	1	0	0	0	2	0	0
	Z4	0	0	8	18	4	0	0	0	0	0
	Z5	0	0	0	0	29	1	0	0	0	0
	Z6	0	0	0	0	3	27	0	0	0	0
	Z7	0	1	0	0	0	0	29	0	0	0
	Z8	0	0	1	0	0	2	0	23	4	0
	Z9	0	0	0	0	0	0	0	3	27	0
	Z10	0	0	0	0	0	0	0	0	0	30

Tabla 15. Confusion matrix día 1, dispositivo 2

		ZONA PREDICHA									
		Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10
ZONA REAL	Z1	26	0	0	0	0	2	0	0	0	2
	Z2	0	30	0	0	0	0	0	0	0	0
	Z3	0	0	28	2	0	0	0	0	0	0
	Z4	0	0	6	19	2	0	0	1	2	0
	Z5	0	0	0	0	26	0	0	4	0	0
	Z6	0	0	0	0	1	29	0	0	0	0
	Z7	0	0	0	0	0	0	30	0	0	0
	Z8	0	0	0	0	0	0	0	27	3	0
	Z9	0	0	1	0	3	0	0	2	24	0
	Z10	0	0	0	0	0	0	0	0	0	30

Tabla 16. Confusion matrix día 2, dispositivo 2

		ZONA PREDICHA									
		Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10
ZONA REAL	Z1	27	0	0	0	0	2	3	0	0	0
	Z2	0	29	0	0	0	0	0	0	0	1
	Z3	0	0	25	5	0	0	0	0	0	0
	Z4	0	0	6	19	5	0	0	0	0	0
	Z5	0	0	0	1	28	0	0	0	0	0
	Z6	0	0	1	0	1	29	0	0	0	0
	Z7	0	0	5	0	0	0	25	0	0	0
	Z8	0	0	3	0	0	0	0	25	2	0
	Z9	0	0	0	1	0	0	0	2	27	0
	Z10	0	0	0	0	0	0	0	0	0	30

Tabla 17. Confusion matrix día 3, dispositivo 2

		ZONA PREDICHA									
		Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10
ZONA REAL	Z1	23	0	0	0	0	2	0	0	0	5
	Z2	0	30	0	0	0	0	0	0	0	0
	Z3	0	0	26	3	0	0	0	0	1	0
	Z4	0	0	15	13	2	0	0	0	0	0
	Z5	0	0	3	2	25	0	0	0	0	0
	Z6	0	0	0	0	4	26	0	0	0	0
	Z7	0	0	0	0	0	0	30	0	0	0
	Z8	0	0	0	0	0	0	0	21	9	0
	Z9	0	0	0	0	0	0	0	3	27	0
	Z10	0	0	0	0	0	0	0	0	0	30

Tabla 18. Confusion matrix día 4, dispositivo 2

### 5.1.2. Resumen del desempeño en las sesiones de test

Día	ID Disp.	Acc. Global	Acc. Z1	Acc. Z2	Acc. Z3	Acc. Z4	Acc. Z5	Acc. Z6	Acc. Z7	Acc. Z8	Acc. Z9	Acc. Z10
1	1	98%	100%	100%	100%	100%	100%	100%	100%	83%	100%	100%
2	1	95%	90%	100%	100%	100%	100%	97%	97%	73%	97%	93%
3	1	98%	100%	100%	100%	93%	93%	100%	100%	97%	100%	100%
4	1	99%	100%	100%	97%	97%	100%	100%	100%	97%	100%	100%
1	2	90%	100%	97%	90%	60%	97%	90%	97%	77%	90%	100%
2	2	90%	87%	100%	93%	63%	87%	97%	100%	90%	80%	100%
3	2	88%	90%	97%	83%	63%	93%	93%	83%	83%	90%	100%
4	2	84%	77%	100%	87%	43%	83%	87%	100%	70%	90%	100%

Tabla 19. Accuracy global y zonales

Día	ID Disp.	Prec. Global	Prec. Z1	Prec. Z2	Prec. Z3	Prec. Z4	Prec. Z5	Prec. Z6	Prec. Z7	Prec. Z8	Prec. Z9	Prec. Z10
1	1	98%	100%	100%	100%	100%	100%	100%	100%	83%	100%	100%
2	1	95%	90%	100%	100%	100%	100%	97%	97%	73%	97%	93%
3	1	98%	100%	100%	100%	93%	93%	100%	100%	97%	100%	100%
4	1	99%	100%	100%	97%	97%	100%	100%	100%	97%	100%	100%
1	2	90%	100%	97%	90%	60%	97%	90%	97%	77%	90%	100%
2	2	90%	87%	100%	93%	63%	87%	97%	100%	90%	80%	100%
3	2	88%	90%	97%	83%	63%	93%	93%	83%	83%	90%	100%
4	2	84%	77%	100%	87%	43%	83%	87%	100%	70%	90%	100%

Tabla 20. Precisión global y zonal

Día	ID Disp	Rec. Global	Rec. Z1	Rec. Z2	Rec. Z3	Rec. Z4	Rec. Z5	Rec. Z6	Rec. Z7	Rec. Z8	Rec. Z9	Rec. Z10
1	1	99%	100%	100%	97%	100%	100%	100%	100%	100%	88%	100%
2	1	95%	90%	100%	100%	100%	94%	100%	91%	100%	78%	100%
3	1	98%	100%	100%	94%	93%	100%	100%	100%	100%	97%	100%
4	1	99%	100%	100%	97%	97%	100%	100%	100%	100%	97%	100%
1	2	90%	100%	97%	75%	95%	81%	90%	100%	82%	87%	97%
2	2	90%	100%	100%	80%	90%	81%	94%	100%	79%	83%	94%
3	2	89%	100%	100%	63%	73%	82%	97%	89%	93%	93%	97%
4	2	85%	100%	100%	59%	72%	81%	93%	100%	88%	73%	86%

Tabla 21. Recall global y zonal

## 5.2. Análisis de los resultados de desempeño de las sesiones de test

A partir de los resultados mostrados anteriormente podemos llegar a una serie de conclusiones.

En primer lugar, atendiendo al accuracy global de la tabla 5.11, se puede confirmar que al trabajar con un dispositivo distinto al que ha realizado la calibración afecta al desempeño del sistema. Para el dispositivo 1, con el que se ha realizado la calibración, se tiene un accuracy global por encima del 95%, en cambio, para el dispositivo 2 (Raspberry PI 4b) se tiene un desempeño global en torno al 88%. Con esto se comprueba la robustez del sistema frente al cambio de hardware, ya que se reduce el desempeño a la hora de estimar una posición con un hardware distinto, pero estamos hablando de una reducción del 7%, por lo que no se trata de una reducción considerable.

Por otro lado, también atendiendo al accuracy global de la tabla 5.1, se puede confirmar la robustez frente al tiempo, ya que no se aprecia una diferencia considerable entre las métricas de los distintos días para un mismo dispositivo. Esto quiere decir que la variación del espectro radioeléctrico no afecta al desempeño del sistema, independientemente del día o la hora.

Además, atendiendo a la tabla 5.11, en la cual se expresa la precisión para cada una de las zonas, se puede afirmar que la zona con una peor precisión es la 4, llegando a tener un nivel del 43% para el dispositivo número 2. Para explicar esto, se ha de observar la matriz de confusión correspondiente a ese resultado, que se encuentra en la tabla 5.9. Si observamos la fila número 4, se ve que de los 30 fingerprints cuya zona real era la número 4, únicamente se ha predicho de manera correcta esta ubicación en 14 de las estimaciones, habiendo predicho la zona 3 un total de 15 veces. Esto quiere decir que el sistema no discierne de manera correcta entre la zona 3 y la zona 4, pero se puede comprobar que esto únicamente ha ocurrido para el dispositivo número 2.



Por último, cabe destacar la relación entre precisión y recall, ya que el punto explicado anteriormente se traduce en una bajada considerable del recall en la zona 3, ya que en la prueba realizada el día 4 para el dispositivo 2, se clasifica de manera incorrecta la zona 3 un total de 18 veces, esto se puede observar en la tercera columna de la matriz de confusión de la tabla 5.11. Esto quiere decir, que de un total de 44 predicciones cuyo resultado fue la zona 3, únicamente 26 fueron acertadas.

### 5.3. Conclusiones y trabajos futuros

En el presente TFM se ha diseñado y desarrollado una solución integral para un sistema de localización de interiores en una vivienda particular, teniendo en cuenta los requisitos de usar tecnologías estándares con productos comerciales disponibles en el mercado, reduciendo el coste del sistema y ofreciendo una solución cuya puesta en marcha fuese lo más sencilla posible.

Se ha comprobado que la tecnología Bluetooth Low Energy utilizada en el proyecto es idónea para este tipo de servicios de localización en interiores debido, en gran medida, a la simplicidad del mecanismo de advertisement. Los estudios de propagación y cobertura realizados en el escenario de este proyecto, confirman el buen funcionamiento de la tecnología BLE para cualquier entorno de este tipo. Dado los planos y zonificación de una vivienda, realizando la adecuada planificación RF, se puede optimizar el número de beacons y su ubicación logrando una cobertura total en todas las zonas objetivo.

Se ha sintetizado una arquitectura y solución a medida para una vivienda familiar minimizando el coste de instalación, así como el cableado, siendo un sistema “plug and play”. Además, se ha desarrollado un protocolo de calibración sencillo para que cualquier individuo, independientemente de sus conocimientos informáticos, pueda poner en marcha el sistema de localización.

Se ha diseñado un motor de localización propio a través de algoritmos de machine learning y puesto a prueba de manera exhaustiva con distintos parámetros de configuración y ficheros test.

- Se ha puesto a prueba la diversidad temporal del sistema, realizando pruebas en distintos días y horarios con el fin de comprobar la variación del espectro frecuencial en función del tiempo.
- Se ha testado el motor de localización utilizando distintos dispositivos IoT con el objetivo de conocer la robustez del sistema frente a la variación del hardware utilizado, con resultados muy satisfactorios.
- Se ha desarrollado un sistema muy depurado de evaluación del desempeño a partir del cual se obtienen métricas de interés utilizadas comúnmente en machine learning, pudiendo así evaluar resultados de manera sencilla y eficiente.

En resumen, los resultados obtenidos en el TFM son una solución completa, desde la capa física hasta el nivel de aplicación, de un sistema de localización en interiores diseñado para viviendas familiares utilizando la tecnología Bluetooth Low Energy.

### 5.3.1. Trabajos futuros

Las líneas de trabajo a seguir para mejorar y poder industrializar este sistema son:

- Desarrollo de un dispositivo IoT único capaz de albergar los distintos módulos de manera compacta y que pueda ser transportada de manera sencilla por un individuo (RaspiZero, ESP32, Smartphone...)
- Desarrollo de motor de localización en Python con el fin de unificar las tecnologías del sistema y que sean fácilmente exportables a cualquier hardware compacto.
- Planificador RF mediante simulación, que permita de manera sencilla conocer el número óptimo y la ubicación de los beacons a instalar en un escenario concreto.
- Mejora de las prestaciones del motor de localización añadiendo otro tipo de entradas como funciones monopolso o beacon diferencial. Esto requerirá de un despliegue un poco más complejo en cuanto a la distribución RF se refiere.
- Revisión de la implementación software para gestión de errores y excepciones y validar el mismo para producción.

## Bibliografía y enlaces

- [1] <https://locatify.com/blog/indoor-positioning-systems-ble-beacons>.
- [2] P. Davidson and R. Piche; A Survey of Selected Indoor Positioning Methods for Smartphones; IEEE Commun. Surv. Tutorials, vol. 19, no. 2, pp. 1347-1370, 2017.
- [3] S. He and S.-H. G. Chan; Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons; IEEE Commun. Surv. Tutorials, vol. 18, no. 1, pp. 466-490, 2016.
- [4] <https://www.python.org/>
- [5] <https://geekytheory.com/que-es-mqtt>
- [6] [https://es.wikipedia.org/wiki/Baliza\\_electr%C3%B3nica](https://es.wikipedia.org/wiki/Baliza_electr%C3%B3nica)
- [7] <https://accent-systems.com/support/knowledge/ibks-config-tool-user-manual/>
- [8] <https://www.bluetooth.com/>