

UNIVERSITAT  
JAUME•I

GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO DE FINAL DE GRADO

---

**Sensorización de interiores para  
monitorizar personas u objetos**

---

*Autor:*  
Dídac Fibla Escoto

*Supervisor:*  
Joaquín Torres Sospedra  
*Tutora académica:*  
María Asunción Castaño  
Álvarez

Curso académico 2020 / 2021  
Fecha de lectura: 15 de Septiembre de 2021



*A mi familia, por todo su esfuerzo durante tantos años para que a mí nunca me falte de nada,  
a mi pareja Lucía, por estar conmigo y apoyarme incluso en los momentos más difíciles,  
a mi supervisor Joaquín, por haberme enseñado tanto durante la estancia en prácticas,  
a mi tutora Asun, por ser una profesora con gran vocación, sobretodo por tu apoyo  
incondicional y dedicación.*



# Resumen

Las tecnologías geoespaciales son de gran utilidad para empresas y administraciones ya que permiten obtener información de gran valor. Se trata un campo que se encuentra en constante crecimiento en la actualidad.

Este proyecto está orientado a realizar los análisis pertinentes para alcanzar una solución de sensorización óptima de un Sistema de Localización en Interiores (SLI) basado en tecnología inalámbrica de bajo consumo, más concretamente *Bluetooth Low Energy (BLE)*. El bajo coste de los dispositivos BLE, sumado a su bajo consumo, les permite estar operativos durante meses o incluso años con una pequeña batería. Colocando una baliza en una persona u objeto nos permitiría localizar su ubicación en tiempo real en un entorno interior donde las tecnologías de geolocalización no son eficaces.

En este documento se recogen los resultados de los estudios para determinar cuál es el hardware de sensorización más adecuado, así como su configuración, para lograr una solución de localización de interiores óptima de manera que maximicemos la eficiencia de la monitorización y minimicemos, al mismo tiempo, el coste de la solución.

## Palabras clave

Posicionamiento en interiores, bluetooth, BLE, baliza, iBeacon.

## Keywords

Indoor positioning, bluetooth, BLE, beacon, iBeacon.



# Índice general

<b>1. Introducción.....</b>	<b>7</b>
1.1. Contexto y motivación del proyecto.....	7
1.2. Objetivos del proyecto .....	8
1.3. Estructura de la memoria .....	9
<b>2. Descripción del proyecto .....</b>	<b>11</b>
2.1. Estado del arte .....	11
2.2 Metodología.....	12
2.3. Planificación inicial .....	12
2.4. Planificación final.....	13
2.5. Diferencias entre la planificación inicial y final .....	15
2.6. Estimación de recursos y costes del proyecto .....	16
<b>3. Introducción a los sistemas de localización .....</b>	<b>19</b>
3.1. Técnicas de estimación de distancias .....	20
3.1.1. Fuerza de la señal recibida .....	20
3.1.2. Tiempo de llegada.....	21
3.1.3. Diferencia en tiempo entre llegadas .....	22
3.1.4. Ángulo de llegada.....	22
3.2. Técnicas de estimación de la localización basados en RSS .....	23
3.2.1. <i>Fingerprinting</i> .....	23
3.2.2. Trilateración y triangulación.....	24
<b>4. Sistemas de localización en exterior .....</b>	<b>27</b>
4.1. Tecnología GPS.....	27
4.2. Tecnología GSM .....	28
4.3. Tecnología LPWAN .....	28
4.4. Comparación de las tecnologías de exterior .....	28

<b>5. Sistemas de localización en interior.....</b>	<b>31</b>
5.1. Tecnologías para SLI .....	32
5.1.1. Tecnología <i>Bluetooth</i> .....	33
5.1.1.1. Bluetooth Low Energy y Bluetooth 5.....	33
5.1.1.2. Arquitectura BLE.....	34
5.1.1.3. Procedimiento de advertising .....	37
5.1.2. Tecnología <i>Ultra Wide Band</i> .....	38
5.1.3. Tecnología Wi-fi.....	38
5.1.4. Tecnología ZigBee.....	39
5.1.5. Tecnología ultrasonidos.....	39
5.2. Protocolos para las balizas .....	39
5.2.1. Protocolo <i>iBeacon</i> .....	39
5.2.2. Protocolo <i>Eddystone</i> .....	40
5.2.3. Comparación entre los protocolos .....	41
<b>6. Requisitos, análisis y diseño .....</b>	<b>43</b>
6.1. Requisitos .....	43
6.2. Análisis sobre los sensores .....	45
6.3. Diseño.....	50
<b>7. Análisis de la señal.....</b>	<b>53</b>
7.1. De RSSI a metros .....	57
7.2. Filtro estadístico.....	59
7.3. Filtro de Kalman.....	62
7.4. Filtro estadístico vs filtro Kalman .....	64
7.5. Filtro de Kalman extendido.....	65
7.6. Conclusiones sobre el análisis de la señal .....	66
<b>8. Ubicación de los sensores y configuración de las balizas.....</b>	<b>67</b>
8.1. Ubicación de los sensores.....	67
8.2. Configuración de las balizas.....	72
<b>9. Conclusiones .....</b>	<b>75</b>
<b>Bibliografía .....</b>	<b>77</b>



# Capítulo 1

## Introducción

### 1.1. Contexto y motivación del proyecto

Este proyecto se ha desarrollado durante la estancia en prácticas del Grado de Ingeniería Informática de la Universidad Jaume I (UJI), concretamente en la empresa Ubik Geoespatial Solutions S.L. [1] (Ubik en adelante), ubicada en el Parque Científico, Tecnológico y Empresarial de la UJI.

La empresa Ubik se dedica a desarrollar soluciones web y aplicaciones móviles que requieren integración de información geoespacial (*Smart Cities*, *Web Mapping*) y aplicaciones basadas en la localización del usuario (*Geo Games*).

Ubik conoce el gran valor que tiene la información geoespacial para aquellas empresas u organizaciones que podrían aprovechar la localización de personas u objetos. Poder obtener la ubicación en tiempo real de forma precisa permitiría localizar o realizar un seguimiento en algunos entornos, por ejemplo, mantener localizados ciertos objetos o personas en entornos industriales, o bien en hospitales o residencias de la tercera edad, donde conocer la ubicación del personal sanitario o mantener localizadas a ciertas personas con enfermedades como Alzheimer podría ayudar a mejorar su calidad de vida y su seguridad.

Así pues, en este proyecto se han realizado una serie de estudios para poder implementar un Sistema de Localización en Interiores (SLI) que permita localizar una persona u objeto en tiempo real y también realizar un seguimiento. El objetivo de estos estudios consiste en analizar diferentes tecnologías, componentes *hardware* y su configuración para optimizar el funcionamiento y minimizar el coste del sistema. Además de realizar todos los estudios previos necesarios, también se pretendía poner en funcionamiento un prototipo en un entorno real para verificar su funcionamiento. Sin embargo, debido a la situación de emergencia sanitaria no ha sido posible probar el prototipo en el entorno deseado, por lo que ha sido probado en un entorno alternativo igual de válido.

La motivación de este proyecto reside en la gran utilidad que puede aportar a ciertas empresas u organizaciones ya que ahorrar tiempo a la hora de localizar ciertos objetos móviles como pueden ser, por ejemplo, robots de almacenamiento. Realizar un seguimiento de estos objetos móviles también permitiría detectar si su funcionamiento es correcto o no y corregirlo u optimizarlo.

Otra fuente de motivación reside en la gran utilidad que tiene en el área de la salud, existen estudios que afirman que ciertos patrones de movimiento pueden ayudar a detectar enfermedades como alzhéimer y demencia, sobre todo en las personas mayores. Además, mantener localizadas a personas con estas enfermedades, sobre todo con alzhéimer, ayudaría a mejorar su seguridad ya que es muy habitual que estas personas se pierdan y sufran daños.

## **1.2. Objetivos del proyecto**

El objetivo del proyecto consiste en realizar los estudios previos necesarios para poder implementar un sistema de localización en interiores de manera óptima.

El objetivo principal se puede desglosar en estos subobjetivos:

- Análisis previo de características de diversas plataformas hardware factibles para la realización del proyecto, incluyendo requisitos específicos de conectividad y operación. Selección de las 2 soluciones más prometedoras.
- Análisis completo de las alternativas seleccionadas considerando el coste energético, calidad de las señales recibidas, robustez del HW, tolerancia a fallos, actualización remota, entre otros.
- Optimización de la localización de los sensores a través de análisis teóricos de la distribución de los sensores para maximizar la eficiencia de la monitorización y minimizar, al mismo tiempo, el coste de la solución.
- Optimización de las balizas a través de análisis teóricos de la configuración óptima para las balizas (frecuencia y potencia de transmisión).

- Realizar pruebas empíricas para evaluar la solución con respecto a los siguientes indicadores: número de muestras recibidas por segundo para cada baliza; número máximo de balizas *BLE* (Bluetooth Low Energy) que se pueden monitorizar de forma simultánea por cada sensor, capacidad de preprocesamiento de las muestras y coste económico por m<sup>2</sup>.

Inicialmente los siguientes objetivos no formaban parte de los objetivos o alcance del proyecto, sin embargo, durante la realización del mismo me he dado cuenta de que eran de gran importancia para el desarrollo del mismo, de manera que se añadieron tras la aprobación del supervisor.

- Investigación sobre tecnologías alternativas de localización
- Investigación sobre protocolos para balizas alternativas.
- Análisis del comportamiento de la señal y de diversas técnicas de filtrado para optimizar los resultados y la precisión de la localización.

### **1.3. Estructura de la memoria**

Este documento está formado por un total de 9 capítulos que recogen todos los detalles del proyecto.

En el capítulo 1 (en el que se encuentra este apartado) se describe el contexto, la motivación y los objetivos del proyecto.

En el capítulo 2 se detalla la metodología utilizada, la planificación y la estimación de los costes y recursos.

En el capítulo 3 se introducen los sistemas de localización, se explica cómo funcionan los sistemas de localización, tanto de exterior como de interior. Finalmente se explican diversas técnicas de localización, entre ellas la escogida para este trabajo.

En el capítulo 4 se comparan las diversas tecnologías de localización en exterior y se comenta si estas tecnologías podrían ser aptas para utilizarlas en interiores.

En el capítulo 5 se explican algunas tecnologías específicas para ser utilizadas en un SLI, también los protocolos de comunicación que se van a utilizar y los motivos por los cuales se ha seleccionado estos frente a otros.

En el capítulo 6 se encuentran los requisitos que debe cumplir el sistema que se desea implementar. A continuación, podemos encontrar el análisis y el diseño del mismo. También se hace un análisis y selección del hardware más recomendado para el SLI en función de los requisitos y necesidades.

En el capítulo 7 se analiza el comportamiento de la señal emitida por las balizas, se implementan diversos métodos de filtrado de la señal y se estudian los resultados obtenidos.

En el capítulo 8 se presenta un análisis preliminar acerca la ubicación y el número de sensores necesarios. Además, se presenta una posible configuración para las balizas..

En el capítulo 9 se plantean las conclusiones y propuestas de continuación y mejora del proyecto.

## Capítulo 2

# Descripción del proyecto

Este proyecto se ha desarrollado para estudiar ciertas características (mencionadas en los objetivos del proyecto del capítulo 1) que permitan implementar un SLI utilizando balizas con tecnología *BLE* como base. En este capítulo se presenta la metodología seguida en dicho proyecto, la planificación a priori y a posteriori de este, los recursos utilizados en él y el coste estimado para desarrollarlo.

### 2.1. Estado del arte

Definimos *beacon* o baliza [2] como un dispositivo electrónico que emite una señal de forma constante (a una potencia e intervalo de emisión prefijado) que otros dispositivos electrónicos compatibles que estén dentro del alcance del *beacon* pueden localizar. Por definición, las balizas no interactúan directamente con los dispositivos electrónicos, si no que estos, al localizar un *beacon* son capaces de conseguir una localización o ejecutar una acción.

Un sistema de localización de interiores (SLI) es una red de dispositivos sensores utilizados para localizar inalámbricamente a personas u objetos en el interior de un edificio. En entornos de exterior la solución más utilizada es el GPS, prácticamente universal, que permite obtener una localización precisa y rápida en cualquier punto del planeta. Sin embargo, en el interior de los edificios, la atenuación de las señales es uno de los mayores inconvenientes que presentan el GPS y otras tecnologías de localización de exterior. Por otro lado, en los SLI la precisión es muy importante y esta depende de las señales utilizadas y, por tanto, de la tecnología en la que se basan.

Las aplicaciones que se benefician de los SLI incluyen:

- Almacén
- Aparcamientos cubiertos, para localizar donde se dejó el vehículo.
- Campus académicos.
- Deportes
- Hospitales

- Mapas de centro comercial y aeropuertos.
- Navegación de tiendas.
- Publicidad dirigida.
- Realidad aumentada.
- Redes sociales
- Robótica
- Turismo, hoteles y visitas guiadas a los museos

Podemos encontrar más información acerca de los sistemas de localización de interior en [3] y en [4] y [5] se realiza un estado del arte profundo sobre los SLI.

## **2.2 Metodología**

Ubik ha dado total libertad para seleccionar la metodología que mejor se adapte al desarrollo del proyecto y a las circunstancias. Puesto que la dimensión del proyecto no es excesivamente grande, se ha optado por seguir un modelo tradicional o predictivo, concretamente el modelo de desarrollo en cascada [6] combinado con una metodología ágil como es Kanban [7]. Se ha elegido dicha combinación porque la metodología de desarrollo en cascada presenta algunos inconvenientes como es la dificultad de realizar cambios después que el proceso ha avanzado. De esta manera las tareas principales se han desarrollado a partir de la metodología en cascada, pero cada subtarea se ha desarrollado siguiendo una metodología Kanban dando así mayor flexibilidad al desarrollo del proyecto.

## **2.3. Planificación inicial**

En este apartado se enumeran las tareas que se han identificado inicialmente para el desarrollo del proyecto y su duración estimada ajustándose a los requisitos y objetivos establecidos por el tutor de prácticas. Esta planificación se puede observar en detalle en la tabla 2.1.

<b>N.º</b>	<b>Proyecto</b>	<b>Tiempo</b>	<b>Dependencias</b>
<b>1</b>	<b>Contexto del proyecto</b>	<b>25 h</b>	
<b>1.1</b>	Estudio del caso	5 h	-
<b>1.2</b>	Investigación previa	20 h	1.1
<b>2</b>	<b>Selección del hardware de sensorización</b>	<b>80 h</b>	
<b>2.1</b>	Análisis previo de características y requisitos para el hardware	15 h	1.2
<b>2.2</b>	Análisis de diversos hardware	20 h	2.1
<b>2.3</b>	Selección del hardware	20 h	2.2
<b>2.4</b>	Análisis completo de las alternativas seleccionadas	25 h	2.3
<b>3</b>	<b>Optimización de la localización de los sensores</b>	<b>45 h</b>	
<b>3.1</b>	Investigación previa	15 h	2.4
<b>3.2</b>	Análisis teórico de la distribución de los sensores	30 h	3.1
<b>4</b>	<b>Optimización de las balizas</b>	<b>30 h</b>	
<b>4.1</b>	Investigación previa	15 h	1.1
<b>4.2</b>	Análisis teórico de la configuración óptima para las balizas	15 h	3.2

**Tabla 2.1:** Planificación inicial del proyecto.

## **2.4. Planificación final**

Durante el desarrollo del proyecto he sugerido tareas que previamente no estaban planificadas. Tras notificarlo al supervisor y después de su aprobación, he procedido a añadirlas derivando así en la planificación mostrada en la tabla 2.2. Los detalles acerca de las modificaciones de las tareas y los tiempos dedicados a estas se pueden ver en el apartado 2.5.

<b>N.º</b>	<b>Proyecto</b>	<b>Tiempo</b>	<b>Dependencias</b>
<b>1</b>	<b>Contexto del proyecto</b>	<b>28 h</b>	
<b>1.1</b>	Estudio del caso	5 h	-
<b>1.2</b>	Investigación previa	20 h	1.1
<b>1.3</b>	Investigación de tecnologías de localización alternativas	3 h	1.2
<b>1.4</b>	Investigación de protocolos de <i>beacon</i> alternativos	2,5 h	1.2
<b>2</b>	<b>Selección del hardware de sensorización</b>	<b>102 h</b>	
<b>2.1</b>	Análisis previo de características y requisitos para el hardware	5 h	1.2
<b>2.2</b>	Análisis de diversos hardware	5 h	2.1
<b>2.3</b>	Selección del hardware	1 h	2.2
<b>2.4</b>	Análisis completo de las alternativas seleccionadas	5 h	2.3
<b>2.5</b>	Análisis de la señal	25 h	2.4
<b>2.6</b>	Conversión del RSSI a metros	1 h	2.5
<b>2.7</b>	Investigación sobre distintas técnicas de filtrado	5 h	2.6
<b>2.8</b>	Implementación de un filtro estadístico	20 h	2.7
<b>2.9</b>	Implementación del filtro de Kalman	35 h	2.7
<b>2.10</b>	Selección de la mejor técnica de filtrado implementada	0 h	2.9
<b>3</b>	<b>Optimización de la localización de los sensores</b>	<b>30 h</b>	
<b>3.1</b>	Investigación previa	10 h	2.9
<b>3.2</b>	Análisis preliminar de la distribución de los sensores	20 h	3.1
<b>4</b>	<b>Optimización de las balizas</b>	<b>20 h</b>	
<b>4.1</b>	Investigación previa	10 h	1.1
<b>4.2</b>	Análisis preliminar de la configuración óptima para las balizas	10 h	3.2
<b>5</b>	<b>Pruebas empíricas para evaluar la solución</b>	<b>120 h</b>	<b>2.2</b>

**Tabla 2.2:** Planificación final del proyecto.



## 2.5. Diferencias entre la planificación inicial y final

Durante el desarrollo del proyecto ha surgido la necesidad de añadir nuevas tareas no previstas en la planificación inicial que, tras su aprobación, se han añadido a la planificación. A continuación, se describen todas las modificaciones que ha sufrido la planificación inicial:

- **Tarea 2.5. Análisis de la señal:** Inicialmente esta tarea no estaba prevista. Durante el desarrollo del proyecto he visto la necesidad de analizar la señal durante las pruebas de funcionamiento para verificar el correcto funcionamiento de las balizas y sensores seleccionados. Le he dedicado un tiempo de 25 horas debido a que interpretar una señal correctamente puede llegar a ser bastante complejo.
- **Tarea 2.6. Conversión de RSSI a metros:** Tras analizar la señal he visto que era necesario convertir los valores de las muestras de la señal que son capturados en unidades RSSI (medida de potencia de la señal) a metros para verificar la precisión de los sensores. Le he dedicado 1 hora debido a que la complejidad de esta tarea era relativamente sencilla.
- **Tarea 2.7. Investigación sobre distintas técnicas de filtrado:** Después de analizar la señal y convertirla a metros he llegado a la conclusión de que el sistema no puede funcionar correctamente si no se aplica un filtrado a la señal debido a que el ruido afecta notablemente a la señal y, por ende, los resultados obtenidos no son nada fiables. Por lo tanto, he añadido esta nueva tarea con el objetivo de encontrar una o varias técnicas que permitan filtrar la señal y obtener resultados precisos. A esta tarea le he dedicado 5 horas.
- **Tarea 2.8. Implementación de un filtro estadístico:** Tras investigar sobre distintos métodos de filtrado he añadido esta tarea con el objetivo de implementar un filtro estadístico para mejorar la precisión del sistema. A esta tarea le he dedicado 20 horas debido a que la implementación y la interpretación de los resultados es una tarea compleja.
- **Tarea 2.9. Implementación del filtro de Kalman:** Tras investigar sobre distintos métodos de filtrado he añadido esta tarea con el objetivo de implementar el filtro de Kalman para mejorar la precisión del sistema. A esta tarea le he dedicado 35 horas debido a que la técnica de filtrado de Kalman es compleja.
- **Tarea 2.10. Selección de la mejor técnica de filtrado implementado:** Después de haber implementado las distintas técnicas de filtrado he de comparar los resultados y escoger la que proporcione mejor resultados.

Como se puede ver también en la planificación final (tabla 2.2), se han modificado las horas de las demás tareas previstas inicialmente para que se ajusten al tiempo real dedicado. Al surgir las tareas imprevistas comentadas arriba, no hubo tiempo para realizar la optimización de la ubicación de los sensores y la configuración de las balizas y, en su lugar, se llevó a cabo un estudio preliminar.

## 2.6. Estimación de recursos y costes del proyecto

En este apartado se muestra la estimación de recursos y costes del proyecto. Han sido clasificados en costes de recursos humanos y costes debidos a los recursos físicos (hardware y software).

El coste de recursos humanos está formado por las horas trabajadas, los gastos de contratación (25% del salario) y los gastos indirectos (20 % del salario). A partir de la información dada por el tutor en función de los contratos de prácticas, se ha hecho una estimación, que se encuentra en la tabla 2.3 y donde se muestra el desglose de los costes.

El coste de los recursos hardware corresponde al coste de los sensores y balizas utilizados para realizar los estudios. En la tabla 2.4 pueden verse en detalle estos costes. En el último campo de la tabla se encuentra el coste prorrateado calculado a partir de la ecuación 2.1. En esta ecuación  $C_t$  hace referencia a coste total y  $V_{ut}$  es la vida útil en meses del producto. Finalmente,  $V_{us}$  corresponde al tiempo en meses que se ha utilizado el recurso, en este caso, 2 meses.

$$C_t = \frac{C_t * V_{us}}{V_{ut}} \quad (2.1)$$

La vida útil de los recursos ha sido proporcionada por Ubik, como una estimación aproximada. Los costes han sido extraídos de a partir de consultas propias en páginas web.

El coste de los recursos software para el desarrollo del proyecto y los estudios correspondientes ha sido de 0€ dado que se ha utilizado software con licencia libre. En la tabla 2.5 se puede ver en detalle de dichos recursos.

Finalmente, como resumen, el coste estimado del proyecto puede verse en la tabla 2.6, donde se observa que el coste total asciende a 8.750 €.

RECURSOS HUMANOS			
Recurso	Horas	Coste	Coste total
Horas de trabajo	300	20 €/h	6.000 €
Gastos de contratación	300	5 €/h	1.500 €
Gastos indirectos	300	4 €/h	1.200 €
<b>Coste total</b>			<b>8.700 €</b>

**Tabla 2.3:** Desglose de los costes de recursos humanos.

HARDWARE					
Recurso	Unidades	Coste	Total	Vida útil	Coste prorrateado
Baliza Accent Systems iBKS 105	1	15 €	15 €	48 meses	0,62 €
Raspberry Pi 4 4GB RAM	1	65 €	65 €	60 meses	2,16 €
Packs accesorios raspberry pi 4	1	15 €	15 €	60 meses	0,50 €
Raton USB	1	10 €	10 €	60 meses	0,3 €
Pantalla	1	99 €	99 €	96 meses	2,06 €
<b>Coste total</b>			<b>95 €</b>	<b>-</b>	<b>5,64 €</b>

**Tabla 2.4:** Desglose de los costes de recursos hardware.

SOFTWARE			
Recursos	Uso	Coste	Total
LibreOffice	Documentación de los resultados	0€	0€
IDE PyCharm	Programación	0€	0€
Raspbian	S.O. para la Raspberry Pi 4	0€	0€

**Tabla 2.5:** Desglose de los costes de recursos software.

<b>COSTE TOTAL</b>	
<b>Concepto</b>	<b>Coste</b>
<b>Recursos humanos</b>	8.700 €
<b>Recursos hardware</b>	95 €
<b>Recursos software</b>	0 €
<b>Coste total</b>	<b>8.795 €</b>

**Tabla 2.6:** Costes totales estimados del proyecto.

## Capítulo 3

# Introducción a los sistemas de localización

En este capítulo se expone una introducción a los SLI. Además, se explican diversas técnicas que se pueden utilizar para estimar distancias en tecnologías inalámbricas y estimar la localización en base a la potencia de la señal recibida.

Hoy en día la mayoría de personas utilizamos los sistemas de localización a diario. Todos estamos familiarizados con estos sistemas ya que están presentes, por ejemplo, en nuestro *Smartphone*, como es el caso del Sistema de Posicionamiento Global (GPS) [8].

La rápida evolución de este tipo de tecnologías ha convertido la necesidad de saber en todo momento dónde se encuentra alguien o algo en un requisito importante para múltiples campos, como pueden ser el caso de localizar a personas, ya sean familiares, mascotas o en llamadas de emergencia y rescates. También en el mundo de la logística y en sectores donde se alquila o presta maquinaria o diferentes objetos puede resultar de mucha utilidad conocer su ubicación para posicionar flotas de vehículos.

A continuación, vamos a hablar de diferentes técnicas existentes para estimar distancias y para estimar la localización. También se presentarán las distintas tecnologías utilizadas para localizar un dispositivo. Primero debemos distinguir los dos tipos de localización existentes, que son la localización en exteriores y en interiores.

La localización en exterior o geolocalización es la capacidad para obtener la ubicación geográfica de un objeto o dispositivo.

En la localización en interiores o SLI se pretende ubicar un objeto o dispositivo en el interior de un edificio. La localización en exteriores e interiores pueden parecer muy similares, pero hay una gran diferencia en las tecnologías y métodos utilizados para conseguir el objetivo de cada una.

### 3.1. Técnicas de estimación de distancias

En los últimos años se han desarrollado diversas técnicas para estimar distancias. A continuación, se van a describir algunas una de ellas y a explicar cuál ha sido la escogida en este proyecto y por qué.

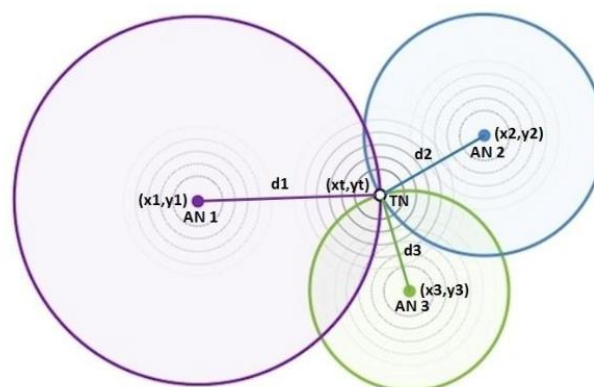
#### 3.1.1. Fuerza de la señal recibida

Esta técnica hace uso de la fuerza de la señal recibida o *RSS (Received Signal Strength)* [9] para determinar la ubicación. El fundamento de este método se basa en la atenuación que sufre la señal a medida que la distancia entre el emisor y el receptor aumenta. En base a la fuerza de la señal podemos estimar la distancia de forma precisa en distancias cortas (inferiores a 20 metros).

Esta es la técnica que se recomienda utilizar para el SLI ya que las principales ventajas son que se trata de la técnica más sencilla de implementar, no precisa de sincronización entre el emisor y el receptor, y, además, es la técnica de menor coste al no necesitar ningún hardware adicional. El principal problema es que la atenuación no es constante, y, por lo tanto, la señal debe ser postprocesada dado que en distancias mayores presenta inconvenientes propios de las ondas de radio como son el multitrayecto, reflexiones y difracciones, entre otros.

Hay dos técnicas que utilizan el *RSS* para estimar la posición: la triangulación y el *fingerprinting*, que se verán más adelante en el apartado 3.2.

En la figura 3.1 se puede ver un ejemplo de cómo se utiliza la fuerza de la señal recibida para estimar la localización mediante triangulación.



**Figura 3.1:** Ejemplo de estimación de la localización basado en la fuerza de la señal recibida mediante la técnica de triangulación<sup>1</sup>.

<sup>1</sup> **Figura 3.1:** Extraída de [9]

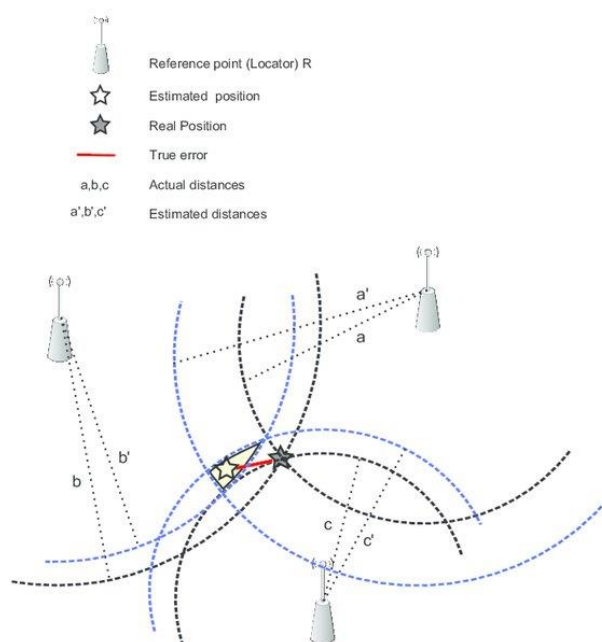
### 3.1.2. Tiempo de llegada

También llamado Tiempo de vuelo, o en inglés *Time of arrival (ToA)* [10], se refiere al tiempo que tarda una señal de radio en llegar desde un transmisor hasta un solo receptor. Se trata de un método para estimar distancias que se utiliza entre otros, en los GPS.

Esta técnica se basa en la relación lineal que hay entre el tiempo de propagación de la señal y la distancia recorrida, dado que la velocidad de las ondas de radio es similar a la de la luz. A partir de esto, mediante la técnica de triangulación, explicada en el apartado 3.2, se puede estimar la distancia entre el emisor y el receptor.

Presenta algunos inconvenientes para aplicarla en un SLI. El tiempo de propagación en interiores es más elevado que en un espacio abierto, por lo tanto, las mediciones de la distancia no son lo suficientemente precisas. Además, este sistema requiere un reloj muy preciso y robusto para sincronizar el emisor con el receptor con el fin de minimizar errores.

En la figura 3.2 se puede ver un ejemplo de cómo se utiliza la técnica *ToA* para estimar la localización.



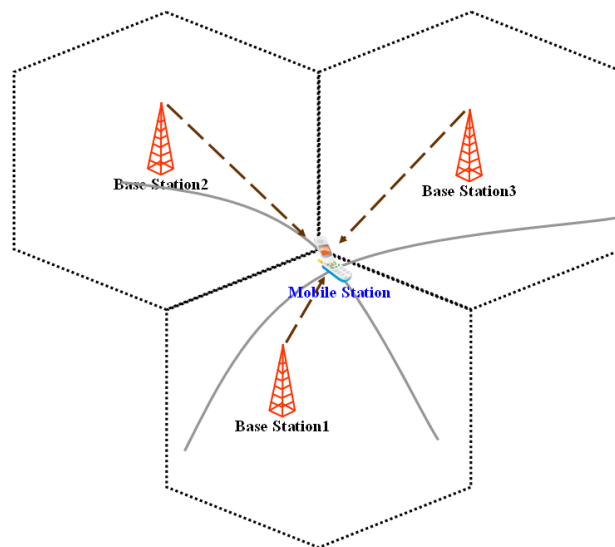
**Figura 3.2:** Ejemplo de estimación de localización utilizando la técnica *ToA*.<sup>2</sup>

<sup>2</sup> **Figura 3.2:** Extraída de [10]

### 3.1.3. Diferencia en tiempo entre llegadas

Esta técnica, también llamada *TDoA (Time Difference of Arrival)* [11], consiste en determinar la posición relativa 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 esto elimina 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.

En la figura 3.3 se puede ver un ejemplo de cómo se utiliza esta técnica para estimar la localización.



**Figura 3.3:** Ejemplo de estimación de la localización utilizando la técnica TDoA.<sup>3</sup>

### 3.1.4. Ángulo de llegada

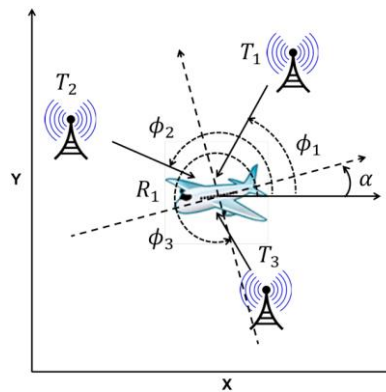
Esta técnica, también llamada en inglés *Angle of Arrival (AoA)* [12], requiere usar diversas antenas de manera que se pueda inferir el ángulo de llegada de la señal, mediante la triangulación de los ángulos se puede estimar la localización. Su principal ventaja consiste en el bajo número de nodos satélites necesarios para el posicionamiento, dos para el plano en dos dimensiones y tres para localizar en tres dimensiones. Sin embargo, su principal desventaja es que requiere hardware adicional (antenas) que incrementa notablemente el coste, además, emisor y receptor

<sup>3</sup> **Figura 3.3:** Extraída de [11]



tienen que estar en línea de visión directa para evitar la angulación incorrecta de la señal de llegada, lo cual es difícil de conseguir en entornos interiores. De esta manera, esta técnica es más adecuada para localizaciones despejadas y al aire libre como los entornos rurales.

En la figura 3.4 se puede ver un ejemplo de cómo se utiliza esta técnica para estimar la localización.



**Figura 3.4:** Estimación de la localización utilizando la técnica ángulo de llegada.<sup>4</sup>

## 3.2. Técnicas de estimación de la localización basados en RSS

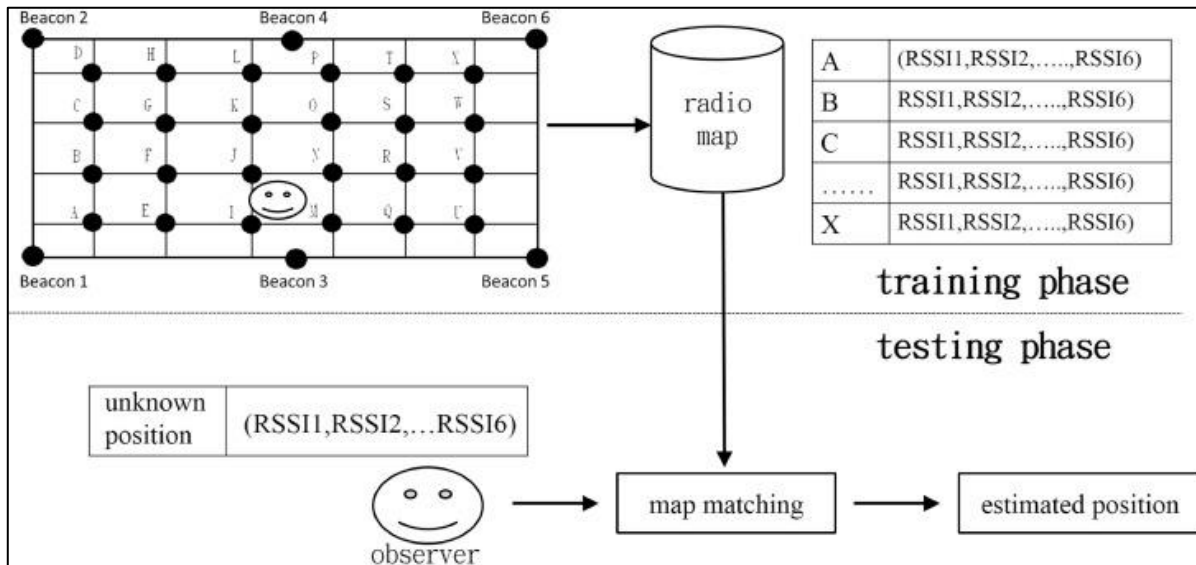
### 3.2.1. *Fingerprinting*

La técnica de *fingerprinting* [13] requiere una base de datos *RSS* bien sólida que se utilizará para generar mapas de intensidad de la señal. Cada punto de referencia incluye la intensidad de la señal medida desde todos los sensores accesibles. Los datos *RSS* emitidos por el dispositivo que se desea localizar se pueden comparar con la coincidencia más cercana de la base de datos que almacena la ubicación de cada punto de referencia para encontrar la ubicación.

Esta técnica comprende dos etapas. En la primera etapa, también llamada *offline*, la intensidad de diferentes señales recibidas es medida en las diferentes ubicaciones del escenario y se almacenan en una base de datos. En la segunda etapa, conocida como etapa *online*, se estima la posición del objeto a localizar consultando la intensidad de la señal asociada a este en la base de datos para encontrar la localización que tenga el valor de la intensidad de la señal más aproximado al recibido. El método de estimación más utilizado es el método del vecino más cercano [14].

<sup>4</sup> **Figura 3.4:** Extraída de [14]

En la figura 3.5 se puede ver un ejemplo de cómo se estima la localización utilizando la técnica de *fingerprinting*.

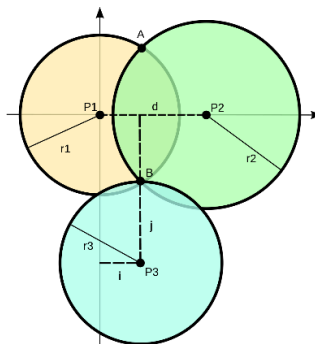


**Figura 3.5:** Ejemplo de estimación de posición utilizando la técnica de *fingerprinting*.<sup>5</sup>

### 3.2.2. Trilateración y triangulación

La técnica de la trilateración [15] utiliza la distancia del nodo hasta diversas balizas para calcular la posición del nodo en dos dimensiones. Como mínimo se necesitan conocer 3 distancias de 3 balizas no colineales. Para calcular la distancia en 3 dimensiones necesitamos como mínimo 4 balizas.

En la figura 3.6 se puede ver un ejemplo de cómo se estima la localización utilizando la técnica de trilateración.



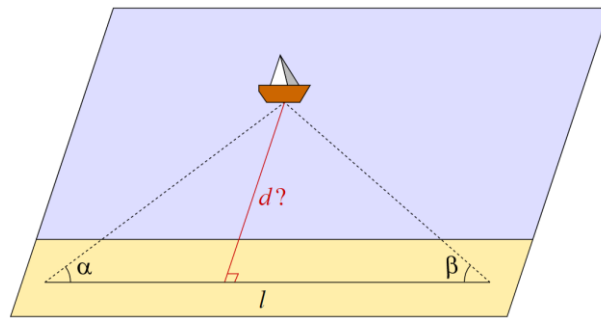
**Figura 3.6:** Ejemplo de trilateración en 3 dimensiones.<sup>6</sup>

<sup>5</sup> **Figura 3.5:** Extraída de [13]

<sup>6</sup> **Figura 3.6:** Extraída de [15]

La triangulación [16] es muy similar a la trilateración, sin embargo, la triangulación utiliza los ángulos para determinar la posición. Como mínimo necesitaremos dos ángulos y la distancia entre un nodo y la baliza. Su funcionamiento para la localización se basa en la suposición de que las antenas generan un patrón de radiación en forma de esfera desde la antena en el centro. Con el radio estimado en base a la potencia recibida por las antenas podemos estimar la localización al intersecar las esferas.

En la figura 3.7 se puede ver un ejemplo de cómo se estima la localización utilizando la técnica de trilateración.



**Figura 3.7:** Ejemplo de triangulación en 3 dimensiones.<sup>7</sup>

---

<sup>7</sup> **Figura 3.7:** Extraída de [16]



## Capítulo 4

# Sistemas de localización en exterior

La precisión de los sistemas de localización de exterior depende mucho de la calidad de la señal que obtengamos de su infraestructura como son las antenas, los repetidores o los satélites. Con ellos se puede llegar a obtener una precisión de metros.

En este capítulo vamos a analizar si es posible utilizar estas tecnologías para un SLI. Nos centraremos en analizar las tecnologías de localización en exteriores más utilizadas, entre las cuales, podemos destacar las siguientes:

- GPS
- GSM
- LPWAN

### 4.1. Tecnología GPS

El Sistema de Posicionamiento Global (GPS) [8] es un sistema de posicionamiento que fue desarrollado por el departamento de defensa de EEUU. Utiliza satélites para encontrar la ubicación de un dispositivo.

El GPS recibe soporte de una constelación de hasta 24 satélites que orbitan por todo el globo terrestre. El objeto o persona a localizar necesita recibir como mínimo la señal de 4 satélites para ser localizado. Cuanto mayor sea el número de satélites mejor será la precisión.

De cada uno de los satélites se obtiene la posición (latitud, longitud y altitud) y el tiempo de envío de cada muestra de señal recibida. Con estos datos es posible calcular la localización del objetivo mediante triangulación.

La precisión de este sistema es de entre 5 y 10 metros. El mayor inconveniente es que requiere de visión directa con los satélites y, por lo tanto, solo funciona adecuadamente en espacios abiertos, de manera que, no la consideramos apta para un SLI.

## **4.2. Tecnología GSM**

El Sistema Global para las comunicaciones Móviles, más conocido como GSM (*Global System for Mobile Comunicación*) [17], utiliza las antenas móviles para localizar un dispositivo. Todas estas antenas tienen un identificador único y, además, tienen almacenada su latitud y su longitud, de manera que, gracias a la potencia de la señal telefónica, podemos determinar la ubicación de un dispositivo triangulando la señal con varias antenas.

La principal desventaja de esta tecnología para un SLI es que actualmente su evolución y mejora se centra en la transmisión de datos en vez de aumentar su precisión para la localización, siendo esta de entre 50 metros y 500 metros. Por lo tanto, no se considera una tecnología apta para un SLI debido a su baja precisión.

## **4.3. Tecnología LPWAN**

*LPWAN (Low Power Area Network)*, también llamadas Redes de Extensa Área y Baja Potencia [18], se está implantando como estándar para comunicaciones entre dispositivos IoT. Sin embargo, esta tecnología no aporta gran precisión para sistemas de localización interior y exterior, ya que es de kilómetros. Su ventaja radica que llega a zonas difíciles donde el GPS o GSM no llegan tan fácilmente y, por lo tanto, se está utilizando para transmitir datos principalmente en ámbitos rurales.

Existen varias implementaciones de LPWAN, entre las que podemos destacar LoRa [19] y SigFox [20] como las más conocidas.

## **4.4. Comparación de las tecnologías de exterior**

En la tabla 4.1 se comparan las prestaciones de las anteriores tecnologías mencionadas. Podemos observar que la tecnología GPS es la que más precisión aporta, aunque es la que mayor tiempo de adquisición tiene. Por otro lado, es la única con la capacidad de informar sobre el rumbo. También hay que destacar que GSM tiene un consumo ligeramente menor que GPS. LPWAN, sin embargo, aunque tiene la menor precisión, también presenta un consumo muy bajo, con lo cual la convierte en una tecnología muy interesante para dispositivos que se alimenten a través de baterías.

<b>Tecnología</b>			
	<b>GPS</b>	<b>GSM</b>	<b>LPWAN</b>
<b>Precisión</b>	5-10 metros.	50-500 m.	Kms.
<b>T. adquisición</b>	12 min.	< 1 min.	<1 min.
<b>Info. Velocidad</b>	Sí + rumbo	No	No
<b>Conectividad</b>	No	Si	Si
<b>Consumo</b>	Alto	Alto/Medio	Bajo
<b>Interior</b>	No	Si	Si
<b>Urbano</b>	++	++	++
<b>Rural</b>	++	+	+++

**Tabla 4.1:** Comparación de tecnologías de geolocalización.<sup>8</sup>

Después de haber analizado las tecnologías de localización exterior más utilizadas, podemos terminar este capítulo con la siguiente conclusión:

Estas tecnologías no aportan la suficiente precisión como para ser utilizadas en un SLI. Esto es debido a que presentan una atenuación de la señal, causada principalmente por los edificios. Por lo tanto, no las considero adecuadas ni aptas para un SLI.

---

<sup>8</sup>Tabla 4.1: Extraída de [21]





## Capítulo 5

# Sistemas de localización en interior

A continuación, vamos a hablar de las diferentes formas que hay para localizar un objeto o persona dentro de un edificio.

Para poder localizar un objeto o una persona en el interior de un edificio vamos a necesitar crear una infraestructura específica dentro de dicho edificio. Está es la principal desventaja frente a los sistemas de localización exterior, en los que se puede aprovechar una infraestructura ya instalada. La instalación de esta infraestructura va a suponer un aumento del coste, sin embargo, podemos lograr precisiones de hasta centímetros.

El funcionamiento de los sistemas de localización en interiores es bastante similar al de los sistemas de geolocalización, aunque utilizan balizas en vez de satélites u otras infraestructuras. Actualmente es habitual referirse a baliza como nodo, *beacon*, *anchor* o *tag*, por lo tanto, para no confundir al lector, en este documento se va a utilizar la palabra baliza.

Existen principalmente dos modalidades de balizas en función de la ubicación de estas, esto es, dependiendo de si está fija y su ubicación es conocida, o bien si la baliza es móvil.

- **Baliza fija:** Diversas balizas se encuentran desplegadas en una ubicación fija y conocida, y su función es emitir una señal. El objeto o persona que deseamos localizar lleva equipado un sensor que capta la señal de las balizas y, mediante las técnicas mencionadas en los apartados 3.1 y 3.2, es posible determinar la localización del objetivo.
- **Baliza móvil:** La persona u objeto que deseamos localizar lleva equipado una baliza que emite una señal y, a medida que se va moviendo por el interior del edificio, se encuentra con sensores que la captan. Con la potencia de la señal recibida a través de varios sensores es posible determinar la ubicación del objetivo.

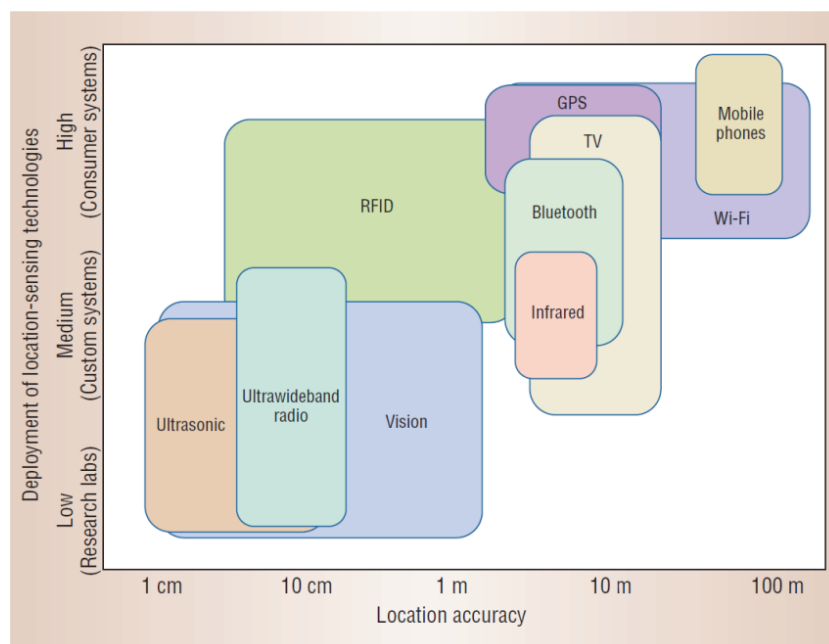
Para nuestro SLI en concreto utilizaremos balizas móviles.

## 5.1. Tecnologías para SLI

El SLI que queremos implementar debe utilizar la tecnología *bluetooth* [22], concretamente la versión *Bluetooth Low Energy (BLE)* [23] ya que se trata de un requisito del mismo establecido por la empresa, como veremos más adelante. De todas maneras, en este apartado vamos a analizar *BLE* y algunas de las tecnologías alternativas más utilizadas para localización en interiores.

Cada una de estas tecnologías tienen sus ventajas y sus inconvenientes que las hacen más o menos adecuadas dependiendo de los requisitos específicos del sistema de localización de interiores.

Como se puede apreciar en la figura 5.1, tecnologías como ultrasonidos, *UWB*, *Visión* y *RFID* pueden alcanzar precisiones muy altas, sin embargo, su uso está más especializado para laboratorios y para sistemas a medida. Por otro lado, vemos que entre las tecnologías más utilizadas por los consumidores estándar se encuentran el *GPS*, *Bluetooth*, infrarrojos, *TV*, *Wifi* y *GSM*.



**Figura 5.1:** Comparación de la precisión y uso de diversas tecnologías de localización.<sup>9</sup>

<sup>9</sup> **Figura 5.1:** Extraída de [24]

### 5.1.1. Tecnología *Bluetooth*

*Bluetooth (BT)* [22] es una tecnología inalámbrica para dispositivos móviles en un área de extensión personal. Permite la transferencia de voz y datos entre dispositivos a través de radiofrecuencia en la banda ISM de 2.4 GHz. Es ampliamente utilizado en dispositivos que pertenecen a sectores de telecomunicaciones y a la informática personal.

Actualmente existen cuatro clases de dispositivos *bluetooth* clasificados por su potencia de emisión. En la tabla 5.1 podemos ver esta clasificación.

	Potencia máxima permitida (mW)	Potencia máxima permitida (dBm)	Alcance aproximado
Clase 1	100 mW	20 dBm	100 metros
Clase 2	2.5 mW	4 dBm	5-10 metros
Clase 3	1 mW	0 dBm	1 metro
Clase 4	0.5 mW	-3 dBm	0.5 metro

**Tabla 5.1:** Clases de los dispositivos BT en relación a la potencia.<sup>10</sup>

Hay distintas versiones de *Bluetooth*, como V1.1, V1.2, V2.0, V2.1, V3.0, V4.0, V5.0, V5.1 y V5.2. En la actualidad los dispositivos utilizan la versión V4.0 o superior. Es muy importante mencionar que, a partir de la versión 4.0, el núcleo del *bluetooth*, además del *bluetooth* clásico que ya incorporaba, añadió *bluetooth* de alta velocidad y *bluetooth* de bajo consumo (V4.2), más conocido *Bluetooth Low Energy*, *BLE* o también *BT Smart*.

#### 5.1.1.1. *Bluetooth Low Energy* y *Bluetooth 5*

*BLE* [23] es un subconjunto de *bluetooth* 4.0 dirigido a las aplicaciones de bajo consumo. En la actualidad se han optimizado las prestaciones de *BLE* dando como resultado *bluetooth* 5, que ha superado a *BLE* en un factor de 2 en la velocidad, 4 en el rango y 8 en los datos, además de tener menor consumo.

En la figura 5.3, podemos ver una tabla donde se ha recogido y se comparan las prestaciones del *bluetooth* clásico con sus respectivas versiones de bajo consumo (v4.2 y v5).

<sup>10</sup> **Tabla 5.1:** Extraída de [22]

	Bluetooth Clásico	Bluetooth Low Energy (4.2)	Bluetooth 5
<b>Distancia interior</b>	~40 metros	~10 metros	~40 metros
<b>Distancia exterior</b>	~100 metros	~30-50 metros	~100 metros
<b>Ancho de banda</b>	1-3 Mbps	1 Mbps	2 Mbps
<b>Retrocompatibilidad</b>	Si	Si	No
<b>Tamaño mensaje</b>		32 by	255 by
<b>Latencia entre paquetes</b>	100 ms	<6 ms	<6 ms
<b>Consumo</b>	~1 W	~0.01 – 0.5 W	~0.01 - 0.5 W
<b>Bluetooth beacons</b>	-	++	+++
<b>Seguridad</b>	56/128 bit	128 bit AES	128 bit AES
<b>Nodos</b>	7	ilimitados	ilimitados

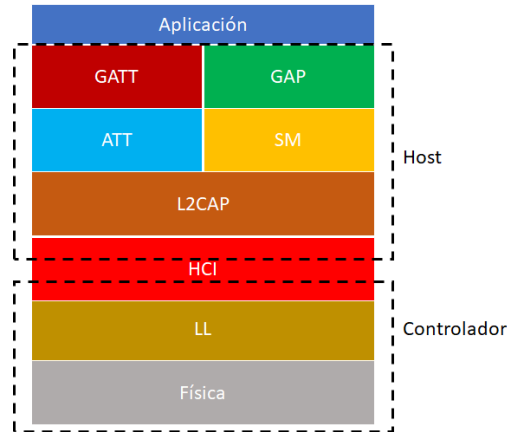
**Tabla 5.2:** Comparativa *BT* clásico, *BLE V4.2* y *BT V5*<sup>11</sup>.

### 5.1.1.2. Arquitectura BLE

La pila de protocolos de la arquitectura BLE consta de tres bloques: controlador, anfitrión y aplicación. En la figura 5.2 podemos ver la agrupación de dichos bloques con sus respectivas capas.

En este apartado no vamos a entrar en detalle en cada una de las capas, solo se van a comentar aquellos aspectos de estas que estén relacionados de manera directa con el desarrollo de este proyecto.

<sup>11</sup> **Tabla 5.2:** Fuente propia

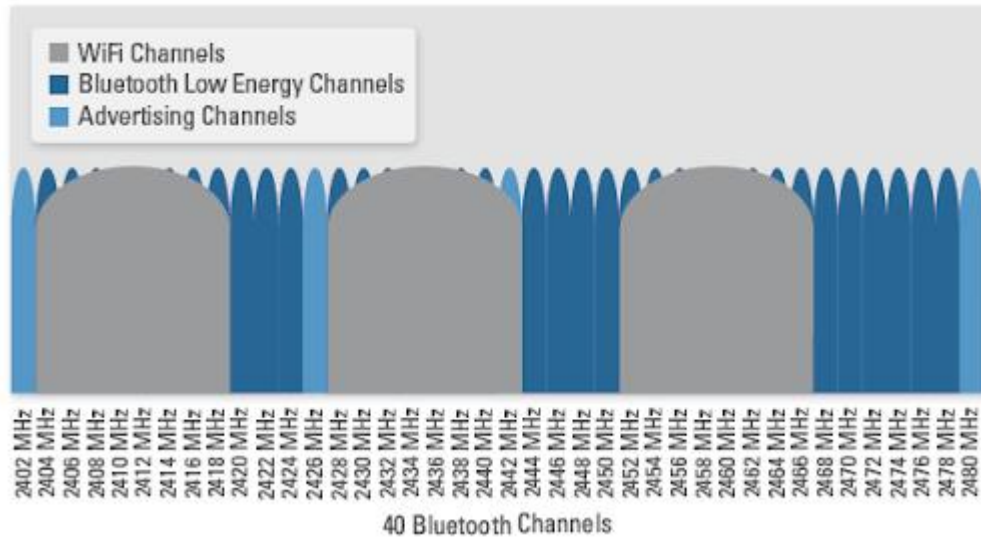


**Figura 5.2:** Bloques y capas de la arquitectura *BLE*.<sup>12</sup>

- Bloque controlador: Está formado por la capa física y la capa de enlace (*link layer*).
  - Capa física: Esta capa contiene la circuitería necesaria para realizar los procesos de modulación de las señales analógicas de radio y convertirlas en información digital.
    - *BLE* tiene 40 canales de 2MHz en la banda ISM de 2,4 GHz, que se muestran en la figura 5.3.
    - Cada canal tiene un ancho de 1MHz y una separación de 2MHz.
    - Emplea la técnica de saltos de frecuencia.
    - 3 canales están dedicados a *advertising* (este concepto se verá más adelante en el apartado 5.1.1.3) y los otros 37 para transmisión de datos.

Estos canales de *advertising* son una de las razones por las que *BLE* es una tecnología apropiada para ser utilizada en las balizas, ya que son los únicos canales que utilizarán las balizas. Estos canales están situados de manera estratégica para evitar interferencias con otras tecnologías que también utilizan la banda de 2,4 GHz, como *Wi-fi* o *ZigBee*.

<sup>12</sup> **Figura 5.2:** Extraída de [25]



**Figura 5.3:** Canales de 2,4 GHz empleados en BLE y WIFI.<sup>13</sup>

- Capa de enlace: Gestiona los requerimientos temporales del estándar, comprueba los mensajes, su reenvío y los errores. Además, ofrece la definición de roles que permiten identificar de forma lógica el rol de cada dispositivo en el proceso de comunicación.
  - En esta capa se definen los siguientes roles: *advertiser*, *scanner*, *master* y *slave*.
- Anfitrión o host: Está formado por las siguientes capas:
  - HCI (*Host Controller Interface*): Define la conexión física entre el host y el controlador.
  - L2CAP (*Logic Link Control and Adaptation Protocol*): BLE utiliza el mismo L2CAP que el bluetooth clásico.
  - ATT (*Attribute Protocol*): Define la comunicación entre dos dispositivos con los roles cliente y servidor.
  - GATT (*Generic Attribute Profile*): Define el marco en el que ATT trabaja para descubrir servicios.

<sup>13</sup> **Figura 5.3:** Extraída de [26]

- GAP (*Generic Acces Profile*): Define cómo los dispositivos interactúan entre ellos en un nivel inferior.
    - GAP define 4 roles: *Broadcaster, Observer, Peripheral y Central*.
  - SM (*Security Manager*): Proporciona un mecanismo para encriptar los datos con el cifrado AES-128 estándar y, además, proporciona autenticación de datos.
- Aplicación:
    - Formada por las aplicaciones de usuario encargadas de interactuar con BT.

### **5.1.1.3. Procedimiento de *advertising***

Los dispositivos BT pueden enviar paquetes de *advertising* [27] (*PDU* o *Protocol Data Units*) para transmitir datos o para permitir que otros dispositivos *BT* los encuentren y se conecten a ellos.

*BLE* utiliza 40 canales de radiofrecuencia diferentes. Tres de estos canales se denominan canales de *advertising* primarios y se utilizan para la comunicación fuera de las conexiones. Esto incluye:

- *Advertisements*.
- Solicitudes de escaneo.
- Respuestas de escaneos.
- Solicitudes de conexión.

Los 37 canales restantes se utilizan principalmente para intercambios de datos dentro de las conexiones *BLE*, tal y como se ilustra en la figura 5.3. Los canales de *advertising* primarios se colocan para evitar los canales Wi-Fi, ya que las transmisiones Wi-Fi causan ruido en los canales *BLE*.

Con Bluetooth 5 se agregan nuevos tipos de *advertising* que amplían las posibilidades de enviar datos en paquetes de *advertisiment*. Bluetooth 5 tiene 37 canales de datos para *advertising*. Cuando se envía *advertising* en los canales de datos, se denominan canales *advertising* secundarios.

## Parámetros de *advertising*

La tabla 5.3. resume los parámetros que se pueden configurar para *advertising*.

Parámetro	Descripción	Rango
Intervalo de <i>advertising</i>	Tiempo entre el inicio de dos eventos <i>advertising</i> consecutivos	[20 ms, 10-24 s]
Tipo de <i>advertising</i>	Se envían diferentes <i>PDU</i> para diferentes tipos de <i>advertising</i> .	
Canal de <i>advertising</i>	Los paquetes de <i>advertising</i> se envían en tres canales	Diferentes combinaciones de canales 37, 38 y 39.

**Tabla 5.3:** Parámetros del paquete de *advertising* de BLE.<sup>14</sup>

### 5.1.2. Tecnología *Ultra Wide Band*

*Ultra Wide Band* o *UWB* [28] es una tecnología para los sistemas de posicionamiento que se basa en la transmisión de pulsos cortos y, utilizando diversas técnicas, provoca la dispersión de la energía de radio en una frecuencia de banda ultra amplia.

Su principal ventaja es la robustez ante cambios de entorno, la alta velocidad de transmisión y el alto rango de distancia. Otra ventaja es que emite pulsos en una baja frecuencia, lo cual facilita que la señal pase a través de obstáculos. Su mayor inconveniente es que no está regulado de tal forma que se pueda usar libremente a escala masiva.

### 5.1.3. Tecnología Wi-fi

Wi-fi [29] es la tecnología de comunicación más extendida en la actualidad. Entre sus ventajas tenemos que, aporta una gran tasa de transmisión y es capaz de suministrar una cobertura de rango de hasta 100 metros. Además, en la actualidad muchos edificios ya cuentan con una infraestructura instalada que se puede aprovechar para el SLI. También hay que tener en cuenta que, además de permitirnos localizar un objeto o persona, puede aportar conectividad a la red.

Su principal desventaja es el consumo energético (medio/alto) y el coste de la infraestructura. Además, en zonas con mucha densidad de población es posible que la frecuencia de transmisión se encuentre saturada.

<sup>14</sup> **Tabla 5.1:** Fuente propia



## 5.1.4. Tecnología ZigBee

*ZigBee* [30] es una especificación de un conjunto de protocolos de comunicación de redes inalámbricas de área personal. Sus principales ventajas son su bajo consumo y su bajo coste. Sin embargo, dado que su alcance es más reducido que la tecnología Wi-fi y su ancho de banda es bajo, en la actualidad se opta por utilizar Wifi antes que ZigBee.

## 5.1.5. Tecnología ultrasonidos

Los ultrasonidos [31], también conocidos como sonar, se utilizan en los sistemas de localización de alta precisión ya que pueden ofrecer precisiones de entre 0,1 y 50 cm. Sin embargo, dado que la señal no se propaga a través de las paredes u obstáculos, cubre un área muy pequeña y es muy susceptible a factores ambientales, por lo que en un SLI habría que incluir una gran cantidad de emisores. Para cada área se necesitan más de 3 emisores con visión directa para garantizar un mínimo de precisión. Por lo tanto, el alto coste de esta infraestructura supone un gran inconveniente para implementar un SLI.

## 5.2. Protocolos para las balizas

Aunque existen otros protocolos, *iBeacon* [32] y *Eddystone* [33] son considerados actualmente como los protocolos estándar para los SLI. La gran mayoría de balizas comerciales son compatibles con estos protocolos.

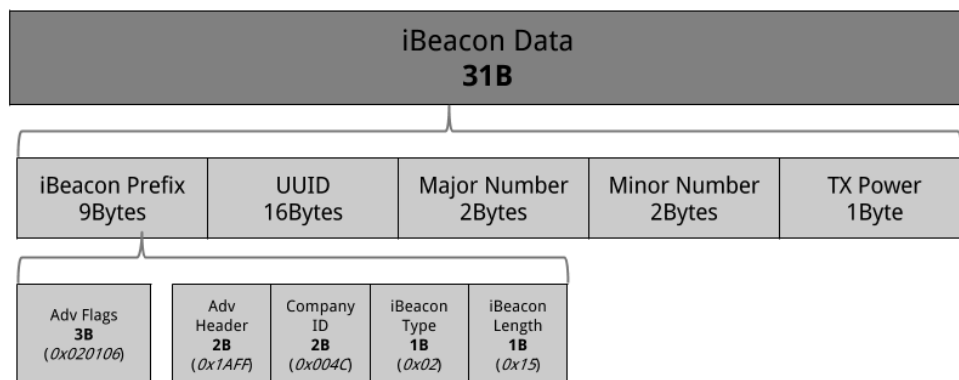
### 5.2.1. Protocolo *iBeacon*

Es un protocolo creado por Apple en 2013 para sistemas de posicionamiento en interiores. Fue el primer protocolo que se desarrolló para este fin y, además, fue el que introdujo la tecnología *BLE* en los SLI. Esta soportado de forma nativa en iOS y también es compatible con Android, aunque no de forma nativa.

El protocolo consta de cuatro partes de información, que podemos ver en detalle en la figura 5.4:

- *UUID*: Es un identificador de 16 bytes que sirve para identificar un grupo de balizas implementadas en un lugar, como puede ser un edificio o una empresa.
- *Major*: Es un identificador de 2 bytes que sirve para identificar un subgrupo dentro del grupo *UUID*. El *major* podría servir, por ejemplo, para indicar la planta de un edificio o departamento.

- *Minor*: Es un identificador de 2 bytes que sirve para identificar una baliza concreta dentro de un grupo.
- *Tx Power*: Se trata de 1 byte, que representa la intensidad de la señal (potencia *tx*) que emite la baliza a un metro.



**Figura 5.4:** Estructura de un paquete *iBeacon*<sup>15</sup>.

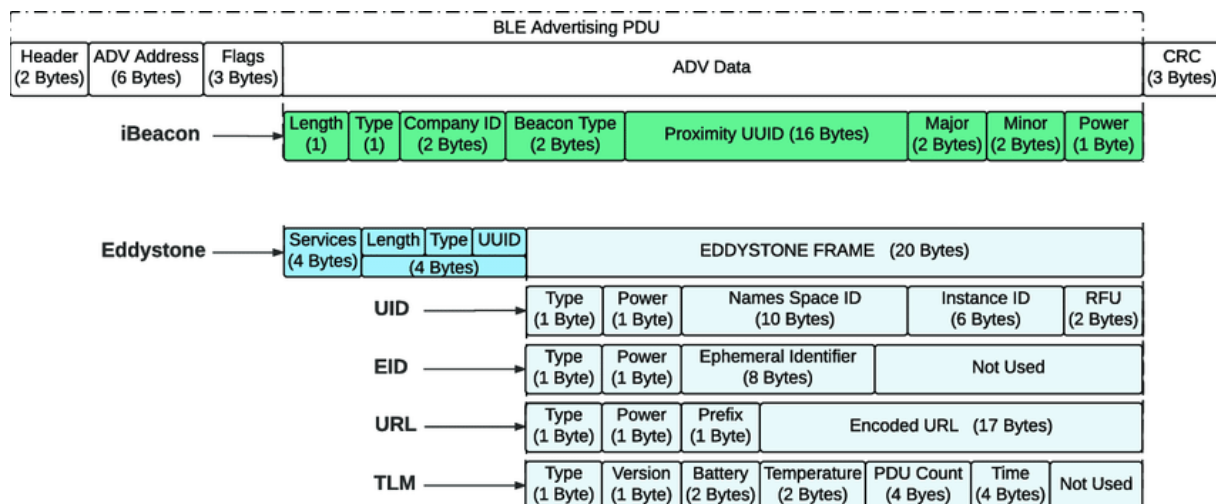
### 5.2.2. Protocolo *Eddystone*

Es un protocolo desarrollado por Google en 2015 y un futuro candidato a ser el protocolo estándar para los SLI. Se trata de un protocolo de código abierto y multiplataforma. Su funcionamiento es muy similar al de *iBeacon*, pero tiene más funcionalidades. Existen tres diferentes tipos de protocolos *Eddystone*, por lo que es más complejo de implementar:

- *Eddystone-UID*: Su funcionamiento es similar al de *iBeacon*, contiene un *UID* para la identificación de la baliza.
- *Eddystone-URL*: Transmite una URL de hasta 18 caracteres.
- *Eddystone-TLM*: Transmite información sobre la baliza, como el nivel de batería, datos de los sensores y otra información relevante.

En la figura 5.5 podemos ver todos los campos de los distintos tipos de protocolo *Eddystone* y también los campos del protocolo *iBeacon*.

<sup>15</sup> **Figura 5.4:** Extraída de [34]



**Figura 5.5:** Campos de datos de un paquete *BLE Advertising* según el protocolo.<sup>16</sup>

### 5.2.3. Comparación entre los protocolos

En la tabla 5.4 podemos ver una comparación de las características de los dos protocolos analizados: *iBeacon* y *Eddystone*.

	<b>iBeacon</b>	<b>Eddystone</b>
<b>Licencia</b>	Propietario	Código abierto
<b>Facilidad</b>	+++	++
<b>Seguridad</b>	No cuenta con seguridad, la señal que se transmite no tiene ningún tipo de encriptación, por lo que puede ser leída y emulada.	Sí cuenta con seguridad, la funcionalidad EID permite transmitir una señal que solo puede ser leída por clientes autorizados
<b>Compatibilidad</b>	iOS: De forma nativa Android: De forma no nativa	Multiplataforma
<b>Tipo de paquetes</b>	Un único tipo: - <i>UUID</i>	Tres tipos distintos: - <i>Eddystone UID</i> - <i>Eddystone URL</i> - <i>Eddystone TLM</i>

**Figura 5.4:** Comparación protocolos *iBeacon* y *Eddystone*.<sup>17</sup>

A la vista del análisis arriba realizado, se decidió utilizar en este proyecto el protocolo *iBeacon* por su simplicidad y facilidad, así como por la larga experiencia que tenía la empresa en este protocolo.

<sup>16</sup> **Figura 5.5:** Extraída de [35]

<sup>17</sup> **Tabla 5.4:** Extraída de [36]



## Capítulo 6

# Requisitos, análisis y diseño

En este capítulo se detallan los requisitos específicos que debería tener el SLI con balizas móviles que deseamos desarrollar. También se plantea en análisis y diseño del SLI a desarrollar.

### 6.1. Requisitos

Algunos de los requisitos del SLI a realizar han sido establecidos por la empresa Ubik, que se presentan a continuación:

- **Tecnología de localización:** La tecnología que debe utilizar el SLI debe ser *BLE*, tal y como se indicó en el capítulo anterior.
- **Protocolo de baliza:** El protocolo para la comunicación entre baliza y sensor debe ser *iBeacon*. Aunque la empresa estableció inicialmente este requisito, en el capítulo anterior se plantea un análisis que confirmó la utilización de este protocolo.
- **Rendimiento y prestaciones:**
  - Se debe maximizar el número de muestras por segundos que puedan captar los sensores.
  - Los sensores deben tener suficiente capacidad de computación para procesar<sup>18</sup> la señal recibida de las balizas.

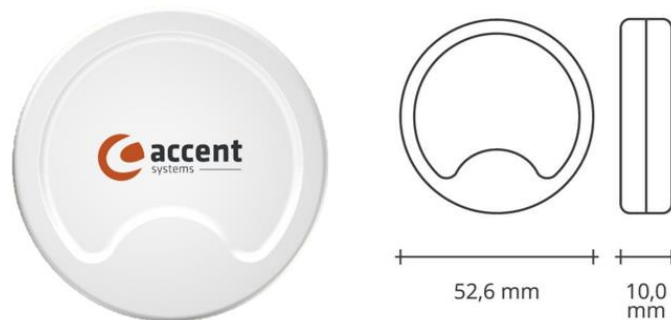
A continuación, se han añadido otros requisitos que se deben tener en cuenta a la hora de seleccionar el hardware para el SLI y que han sido aprobados por el supervisor:

- **Alimentación y consumo energético:**
  - Los sensores deben funcionar directamente a través de la red eléctrica.
  - Es de gran interés minimizar el consumo de los sensores para reducir todo lo posible el impacto medioambiental.
- **Número de sensores:** Se ha de minimizar el número de sensores para reducir el coste total del SLI.

---

<sup>18</sup> Procesar la señal hace referencia a aplicar un filtrado a esta.

- **Escalabilidad:** El sistema ha de ser escalable, las prestaciones del hardware han de permitir que el SLI siga funcionando de manera correcta ante un aumento de la cantidad de balizas.
- **Baliza:** Para el desarrollo de este trabajo se utilizó la baliza iBKS 105 [42], mostrada en la figura 6.1. Se trata de una baliza *BLE* basada en semiconductores *Nordic nrf105* creada por Accent Systems [43]. Este dispositivo ha sido proporcionado por la empresa Ubik. Se trata de un dispositivo de bajo consumo y tamaño reducido. En la tabla 6.1 se muestran sus especificaciones técnicas.



**Figura 6.1:** Baliza iBKS 105 de Accent Systems.<sup>19</sup>

Especificaciones baliza iBKS 105	
<b>Dimensiones</b>	11,3 x Ø52,6 mm
<b>Batería (mAh)</b>	1000
<b>Duración batería</b>	30-40 meses con un intervalo Tx de 1s 3-4 meses con un intervalo de Tx de 100 ms
<b>Resistencia al agua</b>	Opcional
<b>Temperatura de funcionamiento</b>	-30°C hasta +60°C
<b>Protocolos</b>	iBeacon Eddystone Actualización de firmware de forma inalámbrica Servicio GATT
<b>Otras funcionalidades</b>	Reemplazo de batería rápido Kit opcional a prueba de agua

**Tabla 6.1:** Especificaciones técnicas de la baliza iBKS 105 de Accent Systems.<sup>20</sup>

<sup>19</sup> **Figura 6.1:** Extraída de [42]

<sup>20</sup> **Tabla 6.1:** Extraída de [42]

## 6.2. Análisis sobre los sensores

En este apartado se han analizado los distintos dispositivos hardware que nos ofrece el mercado para determinar cuál de ellos es el adecuado para nuestro SLI, teniendo en cuenta los requisitos y funciones que deben desempeñar los sensores. El mercado dispone de muchos sensores que son opciones muy interesantes. Tras estudiar las hojas de especificaciones técnicas de cada uno de ellos y filtrar la información más relevante para nuestro SLI, estos han sido los preseleccionados:

- **Particle Argon**

Particle Argon [37] es un kit de desarrollo de Wi-Fi y *Bluetooth*. Equipado con los procesadores Nordic nRF52840 y Espressif ESP32, tiene un circuito de carga de batería incorporado que facilita la conexión de una batería Li-Po y 20 GPIO de señal mixta para interactuar con sensores, actuadores y otros componentes electrónicos.

Previamente a mi incorporación, la empresa Ubik había realizado pruebas de rendimiento sobre este hardware, los resultados obtenidos fueron que este componente era capaz de capturar como máximo 7 señales simultáneas por segundo emitidas por diferentes balizas *BLE*.

La imagen de este sensor se encuentra en la figura 6.2.



**Figura 6.2:** Kit de desarrollo Particle Argon.<sup>21</sup>

---

<sup>21</sup> **Figura 6.2:** Extraída de [37]

- **ESP32-DevKitC V4**

El módulo ESP32 [38] es una solución que aporta Wi-Fi y Bluetooth todo en uno, integrada y certificada, que proporciona no solo comunicación inalámbrica, sino también un procesador integrado con interfaces para conectarse con varios periféricos. En la actualidad es uno de los componentes más famosos del mercado gracias a su versatilidad, bajo consumo y reducido precio.

La imagen de este sensor se encuentra en la figura 6.3.



**Figura 6.3:** ESP32 de AZ Delivery.<sup>22</sup>

- **Adafruit Feather MO Bluefruit LE**

Adafruit Feather [39] se caracteriza por ser delgada y liviana, sin embargo, no incorpora conectividad Wifi ni BLE de forma nativa. Es la nueva versión de un 'todo en uno' compatible con Arduino + Bluetooth Low Energy con USB integrado y carga de batería.

La imagen de este sensor se encuentra en la figura 6.4.



**Figura 6.4:** Adafruit Feather MO Bluefruit LE.<sup>23</sup>

---

<sup>22</sup> **Figura 6.3:** Extraída de [38]

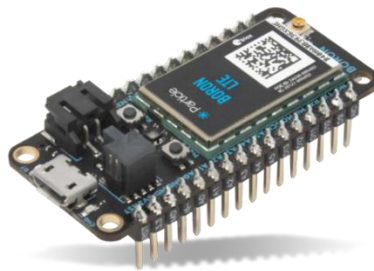
<sup>23</sup> **Figura 6.4:** Extraída de [39]



- **Particle Boron**

Particle Boron [40] es un kit de desarrollo habilitado para celulares, tiene un circuito de carga de batería incorporado que facilita la conexión de una batería Li-Po y 20 GPIO de señal mixta a la interfaz. Tiene sensores, actuadores y otros componentes electrónicos. Es ideal para conectar proyectos existentes a Particle Device Cloud o como puerta de enlace para conectar un grupo completo de puntos finales locales donde falta Wi-Fi o no es confiable.

La imagen de este sensor se encuentra en la figura 6.5.



**Figura 6.5:** Kit de Desarrollo Particle Boron.<sup>24</sup>

- **Raspberry Pi 4**

Raspberry Pi 4 [41] es una computadora de bajo costo y con un tamaño compacto, del tamaño de una tarjeta de crédito. Es lo suficientemente potente para su uso como ordenador de escritorio funcional y dispone de sensores que pueden ser utilizados en nuestro SLI.

En la figura 6.6. se encuentra a imagen de una Raspberry Pi 4.



**Figura 6.6:** Raspberry Pi 4.<sup>25</sup>

---

<sup>24</sup> **Figura 6.5:** Extraída de [40]

<sup>25</sup> **Figura 6.6:** Extraída de [41]

También se han estudiado las especificaciones técnicas de otros sensores, como la variedad de placas de desarrollo de Arduino, Raspberry pi 3, ESP8266 y otros que han sido descartados por no cumplir con los requisitos necesarios.

Tras estudiar en detalle las especificaciones técnicas de cada uno de los sensores seleccionados, se ha realizado una selección de las tres plataformas que más se ajustan a nuestras necesidades:

- Particle Argon, ESP32-DevKitC V4 y Raspberry Pi 4

Particle Argon ha sido preseleccionado por su bajo consumo y facilidad de programación, ya que tiene la ventaja de que es posible programar el componente a través de la nube, permitiendo así modificar su funcionamiento una vez instalado sin tener acceso físico a él, o bien, sin tener que desmontar el sistema una vez instalado.

ESP32 ha sido preseleccionado ya que es un dispositivo que cuenta con un consumo muy bajo y en la actualidad es uno de los dispositivos más utilizados de su categoría, contando con mucha documentación y un gran apoyo por parte de su comunidad.

Raspberry Pi 4 ha sido preseleccionado ya que dentro de los componentes estudiados es el que más prestaciones ofrece. Además, tiene la posibilidad de acceso remoto al componente a través de protocolos como SSH u otros, de manera que lo convierten en una opción muy versátil y potente. La principal desventaja que tiene es que es el que tiene el mayor consumo dentro de las opciones analizadas, sin embargo. Aun así, su consumo sigue siendo muy bajo.

La tabla 6.2 recoge un análisis de los 3 dispositivos seleccionados teniendo en cuenta los requisitos que se desea que cumpla el SLI a desarrollar. Tras el análisis de las prestaciones de cada hardware realizado en la tabla 6.1, se ha llegado a la siguiente conclusión:

Tanto *Particle Argon* como *ESP32* son hardware especializado en el bajo consumo, con lo cual sus prestaciones también se ven más reducidas para estar acorde con el consumo. Nuestro SLI debe tener suficientes prestaciones para maximizar el número de muestras y procesarlas (filtrarlas), además de ser escalable. Con lo cual, de las plataformas hardware analizadas, la única que cumple con todos los requisitos en la *Raspberry Pi 4*. Consecuentemente, ha sido la elegida para el desarrollo del trabajo.

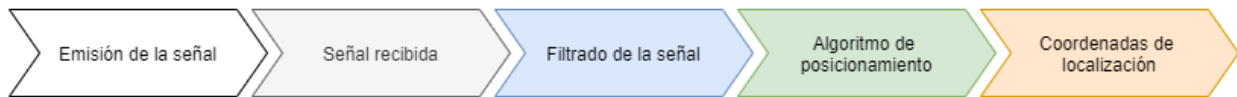
	<b>Particle Argon</b>	<b>ESP32</b>	<b>Raspberry Pi 4</b>
<b>Procesador</b>	ARM Cortex M4F 32b 64Mhz	Tensilica Xtensa DualCore 32b 160Mhz o 240Mhz	Broadcom BCM2711, Quad core Cortex-A72 64-bit 1.5GHz
<b>RAM</b>	256 KB	448 kB	2GB, 4GB, 8GB
<b>ROM / Flash</b>	1MB	520 KB	Max SD 256 GB
<b>Bluetooth</b>	Versión 5.0	Versión 4.2	Versión 5.0
<b>BLE Power transmisión</b>	Desde -20 dBm a +8 dBm	Desde -12 dBm a +9 dBm	Desde -20 dBm a +10 dBm
<b>Consumo BLE</b>	De 340 uA hasta 3420 uA	Recibir: 100mA, Transmitir 130mA a 0 dBm	Scan: 180 uA a 10 dBm
<b>Consumo</b>	De 3.1 hasta 261 mA	De 1.5 a 30 mA de mínimo (sleep) 80mA de promedio, 225 mA de máximo	6W
<b>Sueño profundo</b>	Sí	El modo profundo deshabilita el wifi y el bt.	No
<b>Ethernet</b>	No	No	Si, 1 Gbps
<b>Wifi</b>	ESP32-DoWD 2.4G 4MB Flash 802.11 bgn 150 Mbps	HT40 2,4G 4MB Flash 802.11 bgn 150 Mbps	2.4 GHz y 5 GHz 300 Mbps
<b>Alimentación</b>	Micro USB	Micro USB	
<b>Temperatura</b>	-30°C hasta 75°C	-40°C hasta 125°C	De 0°C a 50°C Throttle a 65°C (requiere disipación) Max 80°C
<b>GPIO</b>	20	11	40
<b>Precio</b>	Entre 23€ y 28\$	Entre 6€ y 8€	2GB RAM – 41€ 4GB RAM – 61€ 8GB RAM – 83€

**Tabla 6.2:** Comparativa de los sensores preseleccionados.

<sup>26</sup> **Tabla 6.2:** Fuente propia, elaborada a partir de las fichas de especificaciones técnicas de cada sensor y fuentes varias.

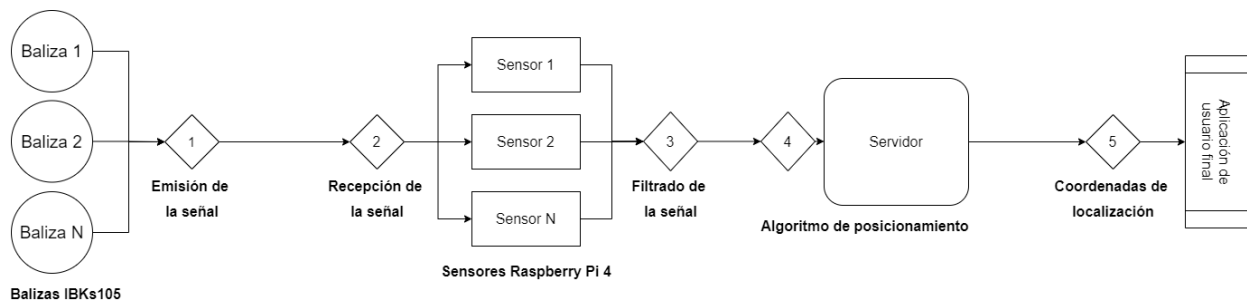
## 6.3. Diseño

En la figura 6.7 se ilustra un diagrama que muestra el comportamiento del sistema mencionado en el apartado anterior.



**Figura 6.7:** Fases del funcionamiento del SLI.<sup>27</sup>

Dado que el diagrama de las fases no es suficiente para describir con profundidad el funcionamiento del sistema, se ha creado otro diagrama para describir en detalle las fases. Este diagrama se puede ver en detalle en la figura 6.8.



**Figura 6.8:** Fases del funcionamiento del SLI.<sup>28</sup>

Fase 1:

- Las balizas IBKS 105 emiten las señales de forma continua.

Fase 2:

- Los sensores capturan la señal de las balizas que se encuentren dentro de su alcance.

Fase 3:

- Los sensores identifican el protocolo de baliza de la señal capturada.
- Los sensores descomponen el paquete iBeacon y extraer los siguientes campos: *uuid*, *major*, *minor* y *rssí*.
- Los sensores filtran la señal (las técnicas de filtrado las veremos en el capítulo 7).
- Los sensores envían la señal filtrada al servidor.

<sup>27</sup> **Figura 6.7:** Fuente propia.

<sup>28</sup> **Figura 6.8:** Fuente propia.

Fase 4:

- El servidor sincroniza las señales filtradas recibidas.
- El servidor aplica el algoritmo de posicionamiento para obtener la estimación de las coordenadas de localización.
- El servidor envía las coordenadas de localización estimadas a la aplicación final del usuario.

Fase 5:

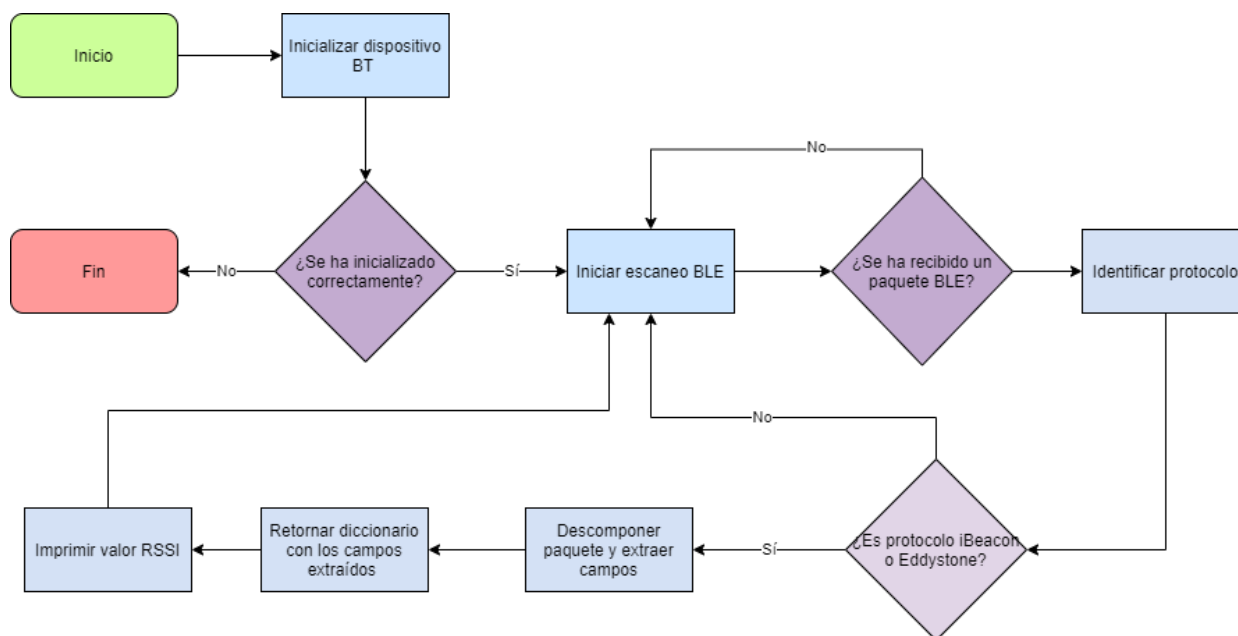
- La aplicación final de usuario muestra las coordenadas en el mapa.



# Capítulo 7

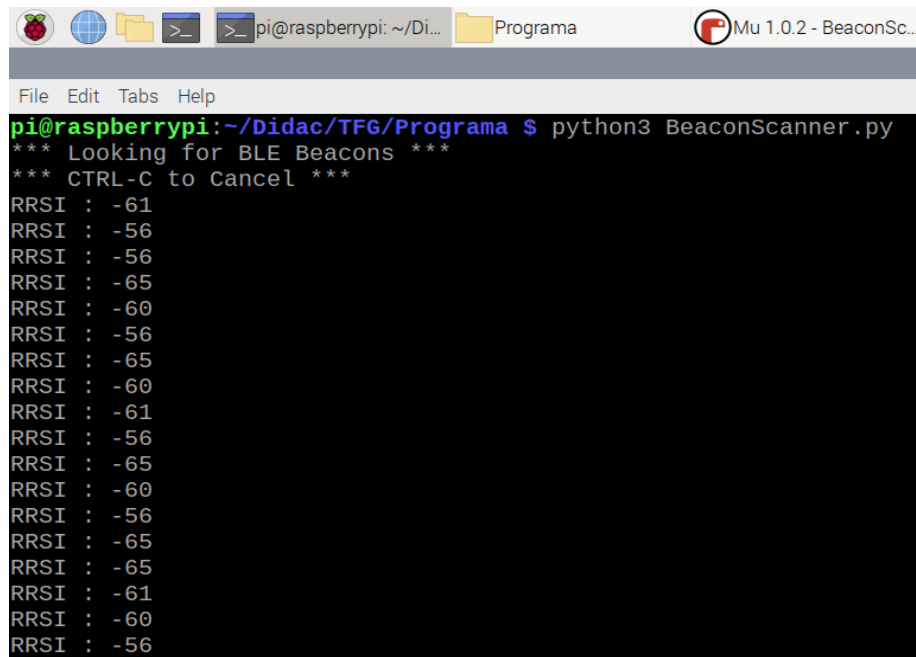
## Análisis de la señal

En este capítulo se analiza el comportamiento que tiene la señal de *BLE* emitida por la baliza IBKS 105 y recibida por el sensor de la Raspberry Pi 4. Para ello, se ha diseñado un programa implementado en lenguaje Python y utilizando la librería *BLE iBeacon Scanner* [44]. En la figura 7.1 se detalla el funcionamiento de este.



**Figura 7.1:** Diagrama de flujo del programa que toma muestras de la señal de las balizas.

Inicialmente se configuró la baliza con una potencia de transmisión de -14 dBm y una frecuencia de envío de la señal de 100 ms, la baliza se situó a 1 metro del sensor y se dejó fija en esa localización. A continuación, se ejecutó el programa y se empezaron a capturar muestras de la señal. En la figura 7.2 se puede ver en detalle los valores RSSI que se obtuvieron.



```
pi@raspberrypi: ~/Didac/TFG/Programa $ python3 BeaconScanner.py
*** Looking for BLE Beacons ***
*** CTRL-C to Cancel ***
RRSI : -61
RRSI : -56
RRSI : -56
RRSI : -65
RRSI : -60
RRSI : -56
RRSI : -65
RRSI : -60
RRSI : -61
RRSI : -56
RRSI : -65
RRSI : -60
RRSI : -56
RRSI : -65
RRSI : -65
RRSI : -61
RRSI : -60
RRSI : -56
```

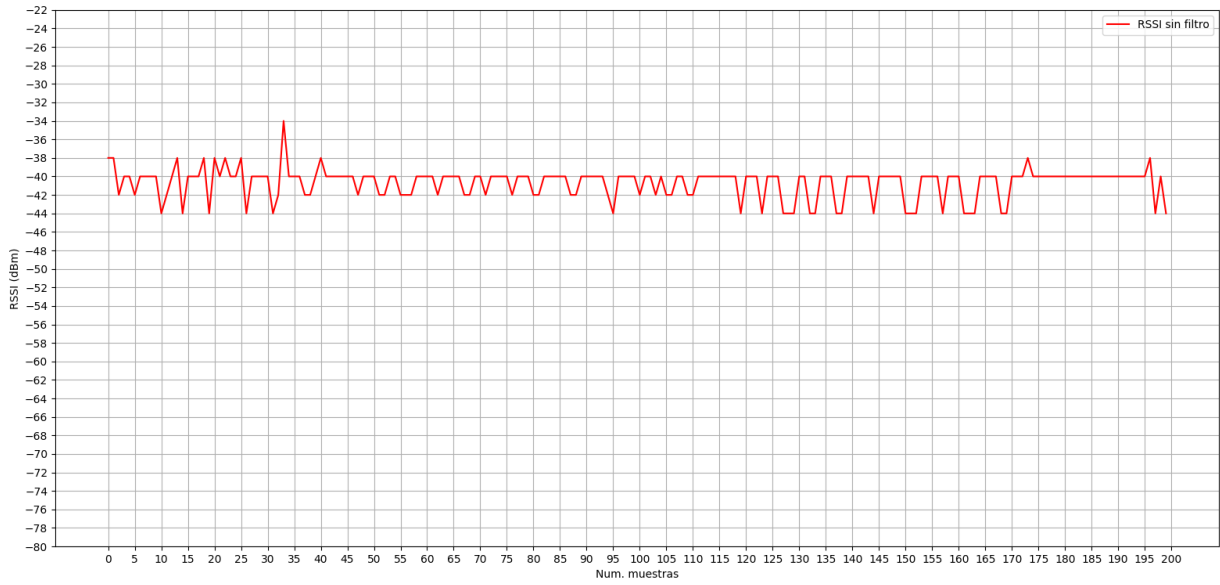
**Figura 7.2:** Valores RSSI obtenidos en una medición con 1 metro de distancia entre la baliza y el sensor.

Se ha podido observar que, aunque la baliza y el sensor se mantengan en una posición fija, el valor RSSI proporcionado, que debería ser muy similar en todos los casos, sufre unas variaciones muy altas, lo que dificulta el cálculo de la distancia con precisión. Cabe matizar que cuanto mayor sea el valor de RSSI recibido, mayor debe ser la intensidad de la señal. Existe una relación entre el RSSI y la distancia: A mayor distancia entre el emisor y el receptor, menor será la intensidad de la señal.

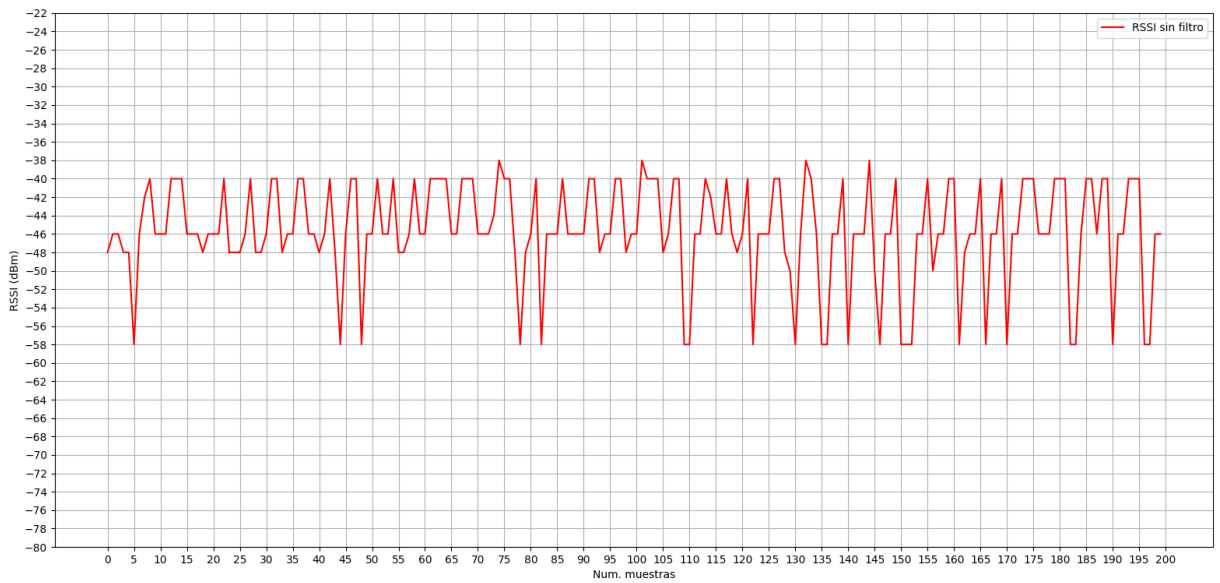
Partiendo del programa anterior y utilizando la librería *MathPlotLib* [45] [46] se diseñó un nuevo programa en lenguaje Python que, tras tomar  $N$  muestras de la señal, proporciona los valores RSSI de la señal capturada en forma de gráfico de líneas para facilitar su interpretación.

Para el siguiente análisis se mantuvo la misma configuración en la baliza que se indicó arriba. A continuación, esta se situó a una serie de distancias del sensor (0,5 metros, 1 metro, 2 metros y 3 metros) y se tomaron 200 muestras de la señal para cada distancia. Los resultados obtenidos se pueden ver en detalle en las figuras 7.3, 7.4, 7.5 y 7.6.

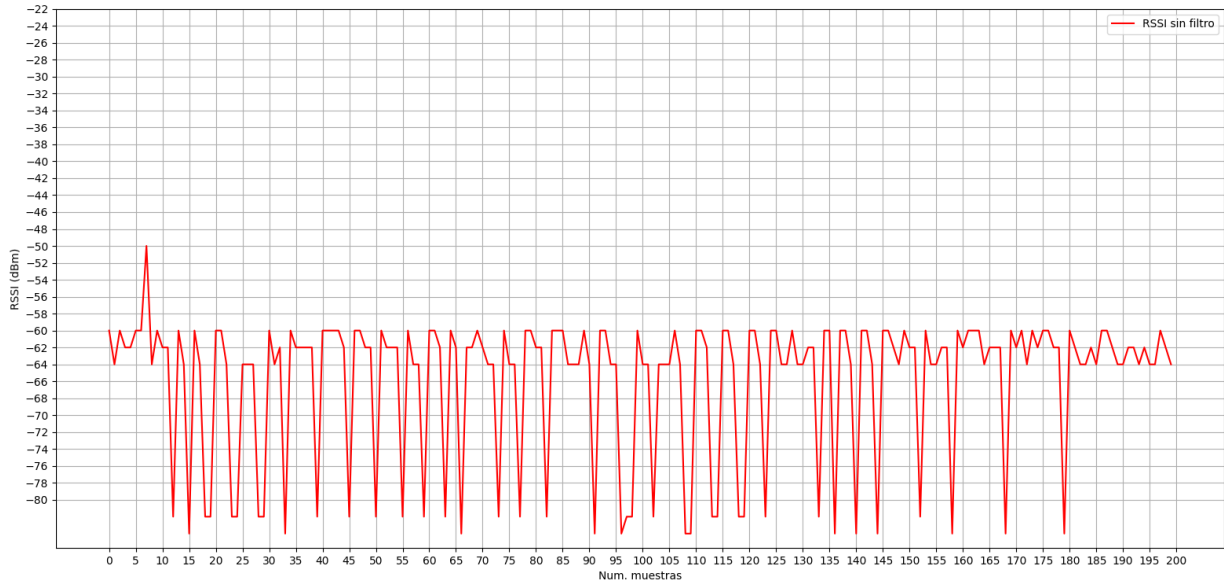




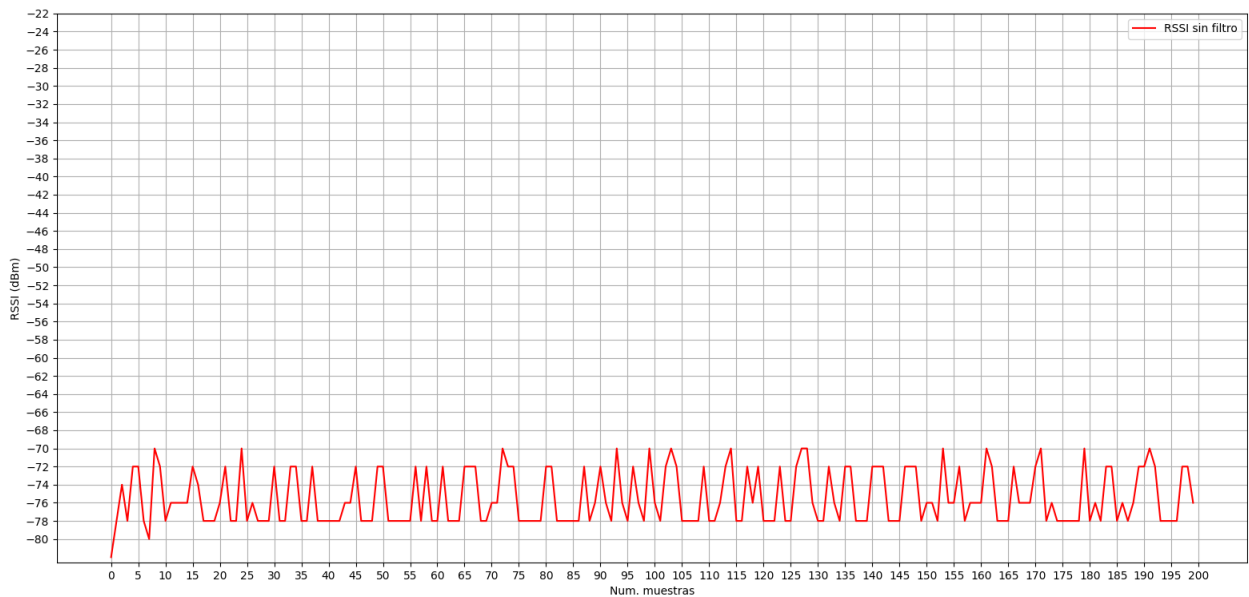
**Figura 7.3:** Evolución de la señal a 0,5 metros de distancia.



**Figura 7.4:** Evolución de la señal a 1 metro de distancia.



**Figura 7.5:** Evolución de la señal a 2 metros de distancia.



**Figura 7.6:** Evolución de la señal a 3 metros de distancia.

En estas figuras se observa que, si medimos el RSSI a lo largo del tiempo, la intensidad de la señal recibida por el receptor está claramente influenciada por la distancia. Además, deducimos que la cantidad de ruido es sustancial dada la variación que sufren los valores RSSI para una misma distancia. Por otro lado, analizando las figuras, no se observa en ninguno de los casos que la señal siga un patrón determinado. Teniendo esto en cuenta se ha llegado a la conclusión de que debemos aplicar un filtrado a la señal para poder estabilizarla. De esta manera, deberíamos poder aumentar la precisión de nuestro SLI.

## 7.1. De RSSI a metros

Habitualmente se suele confundir los términos RSS y RSSI. A continuación, se presenta una breve aclaración de los conceptos mencionados para dejarlos claros al lector:

**RSSI:** *Received Signal Strength Indicator* [47]

**RSS:** *Received Signal Strength* [48]

**dBm:** decibelio-milivatio [49]

RSSI es un indicador y RSS es el valor real. Los valores de radiofrecuencia se expresan en dBm y suelen ser valores negativos la mayor parte del tiempo. Para facilitar la comprensión, estos valores negativos se convierten en positivos mediante una escala.

dBm y RSSI son unidades de medida diferentes que representan lo mismo: intensidad de la señal. La diferencia es que RSSI es un índice relativo, mientras que dBm es un número absoluto que representa los niveles de potencia en mW (milivatios).

El RSSI se mide en dBm, pero en su forma original no es realmente útil en una aplicación (aparte de ser una medida de diagnóstico). Lo bueno de RSSI es que podemos traducir las medidas a estimaciones de distancia en metros. De manera más precisa, podemos describir la relación entre RSSI y la distancia usando el modelo de pérdida de trayectoria de larga distancia, expresado en la fórmula 7.1.

$$RSSI = -10n \log_{10} \left( \frac{d}{d_0} \right) + A_0 \quad 29$$

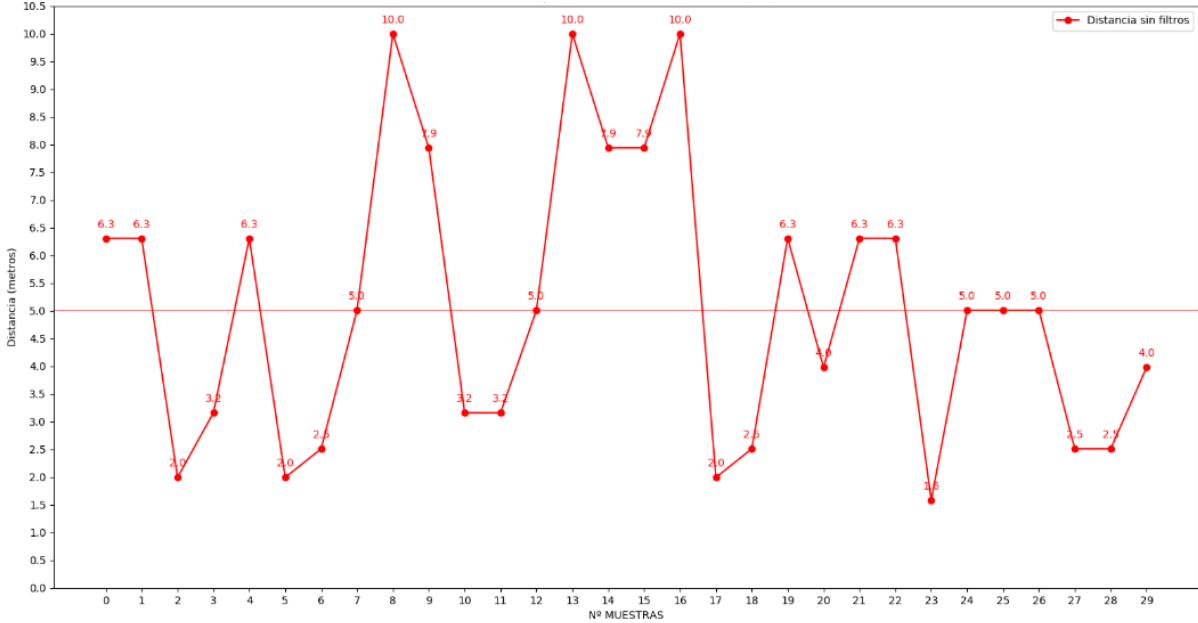
En este modelo,  $d$  describe la distancia entre el emisor y el receptor y  $n$  es el exponente de propagación de la señal.  $A_0$  es el valor obtenido para la intensidad a una distancia  $d_0$ . Por lo general,  $d_0$  vale 1 metro, de manera que  $A_0$  se convierte en la intensidad de la señal medida a una distancia de 1 metro del dispositivo.

El valor de  $A_0$ , se puede obtener desde el propio dispositivo, ya que a menudo forma parte del mensaje de difusión. Aunque también se puede extraer de la ficha de especificaciones técnicas de la baliza. El exponente de propagación de la señal  $n$  es una constante que difiere de un entorno a otro. Para aplicaciones en interiores,  $n$  a menudo se establece en 2.

---

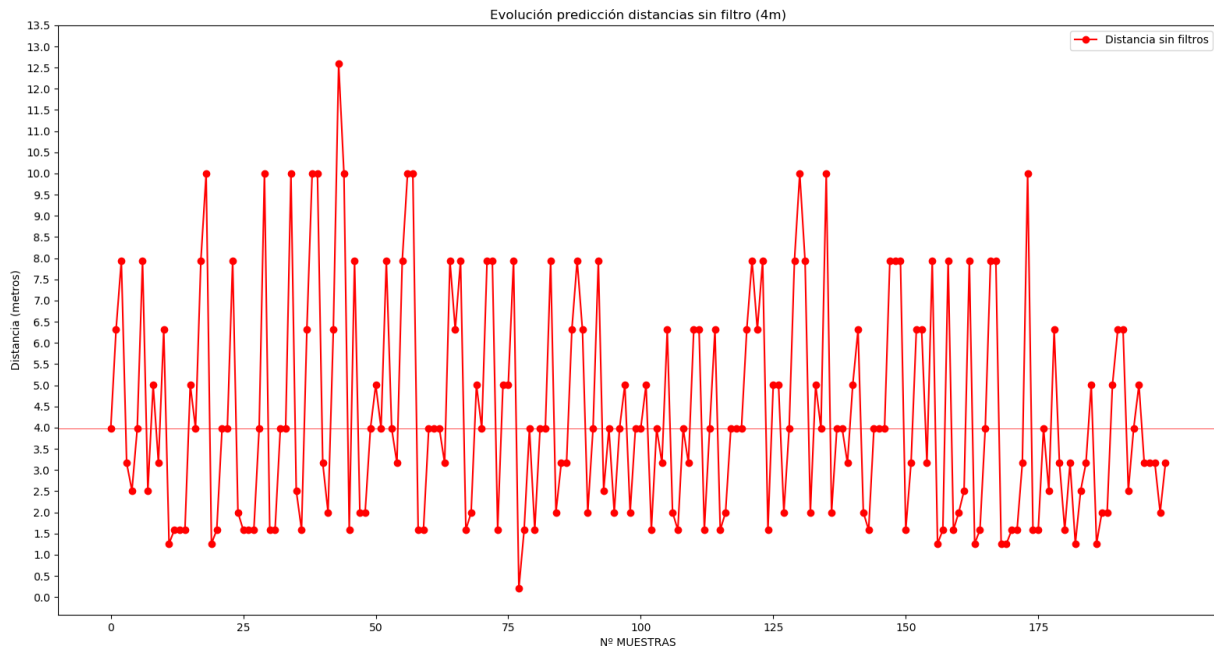
<sup>29</sup> **Fórmula 7.1:** Fuente [50], [51], [52]

Para el siguiente análisis, se mantuvo la configuración de la baliza y esta fue posicionada a 4 metros del sensor. Se realizó una modificación al programa anterior para aplicar la fórmula 7.1 al valor RSSI capturado. A continuación, se ejecutó el programa y se capturaron 30 y 200 muestras de la señal para convertirlas a distancia en metros. En la figura 7.7 y 7.8 podemos ver en detalle las respectivas distancias aproximadas obtenidas. Vemos que la media de todas las distancias en cada figura (línea horizontal) es de 5 m y 4 m respectivamente, cuando debería ser de 4 metros en ambos casos.



**Figura 7.7:** RSSI obtenido sobre 30 muestras para una distancia de 4 metros convertido a distancia en metros utilizando el modelo de pérdida de trayectoria de distancia logarítmica.

Aunque el sensor y la baliza se encontraban a una distancia fija de 4 m, la distancia aproximada que se obtuvo variaba en exceso debido a la inestabilidad de la señal causada por el ruido. Por lo tanto, vimos de nuevo la necesidad de filtrar la señal para mejorar la precisión del SLI.



**Figura 7.8:** RSSI obtenido sobre 200 muestras para una distancia de 4 metros convertido a distancia en metros utilizando el modelo de pérdida de trayectoria de distancia logarítmica

Como acabamos de ver, podemos traducir el RSSI a distancia, pero nuestras estimaciones aún no son muy buenas. Como se muestra en la figura 7.8, en algún momento (aproximadamente en la muestra número 41) nuestra estimación basada en RSSI establece que el dispositivo debería estar a 13 metros de distancia, cuando en realidad está a 4 metros de distancia. ¿De dónde viene esta discrepancia? Siguiendo nuestro modelo, en un mundo ideal, el valor RSSI solo depende de la distancia entre los dos dispositivos. Cuando en realidad, sin embargo, los valores RSSI están fuertemente influenciados por el medio ambiente y, en consecuencia, tienen altos niveles de ruido. Este ruido es causado, por ejemplo, por reflejos de múltiples trayectorias: las señales rebotan contra los objetos del entorno, como paredes y muebles. Entonces, tenemos que luchar contra el ruido. Aquí entran en juego el filtro estadístico y el filtro de Kalman que veremos en detalle a continuación.

## 7.2. Filtro estadístico

En los apartados anteriores hemos visto cómo se comporta la señal. A continuación, se ha analizado si, aplicando un método de filtrado a la señal, se puede lograr mejorar el funcionamiento y la precisión del SLI. Para ello, hemos diseñado un programa que, tras capturar la señal, aplica una media móvil [55] con el objetivo de suavizar los valores RSSI recibidos.

Para diseñar este programa se ha utilizado el módulo de Python *bottleneck* [53]. “El módulo *bottleneck* es una compilación de métodos *numpy* [54] rápidos. Este módulo tiene la función *move\_mean()*, que puede devolver la media móvil de algunos datos.” En la figura 7.9 podemos ver el método en cuestión desarrollado para ello. A continuación, en la figura 7.10 se muestra el resultado de aplicar este método a estos datos de ejemplo: [10, 5, 88, 9, 15, 22, 26, 11, 15, 16, 18, 7]. Dado que el intervalo de la ventana de tiempo (el parámetro **n**) es 4, hay tres valores *nan* al principio porque no se puede calcular la media móvil para ellos.

```
import bottleneck as bn
import numpy as np
def rollavg_bottleneck(a,n):
    return bn.move_mean(a, window=n,min_count = None)

data = np.array([10,5,8,9,15,22,26,11,15,16,18,7])

print(rollavg_bottleneck(data, 4))
```

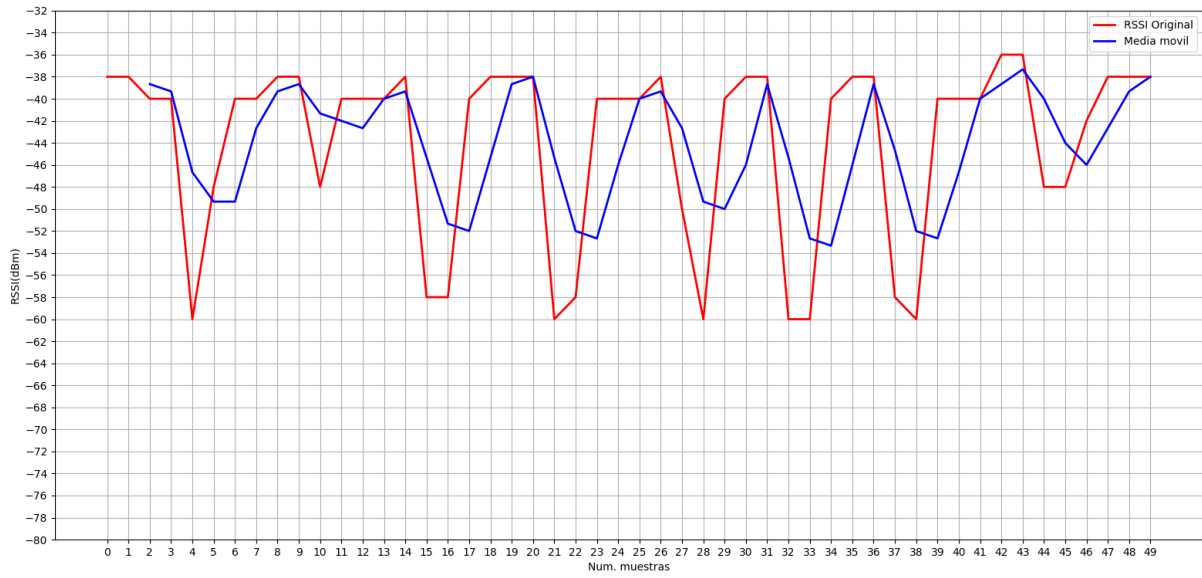
**Figura 7.9:** Método *rollavg\_bottleneck*<sup>30</sup> encargado de calcular la media móvil sobre el vector de enteros [10, 5, 88, 9, 15, 22, 26, 11, 15, 16, 18, 7].

```
[ nan nan nan  8.   9.25 13.5 18.   18.5 18.5 17.   15.  14. ]
```

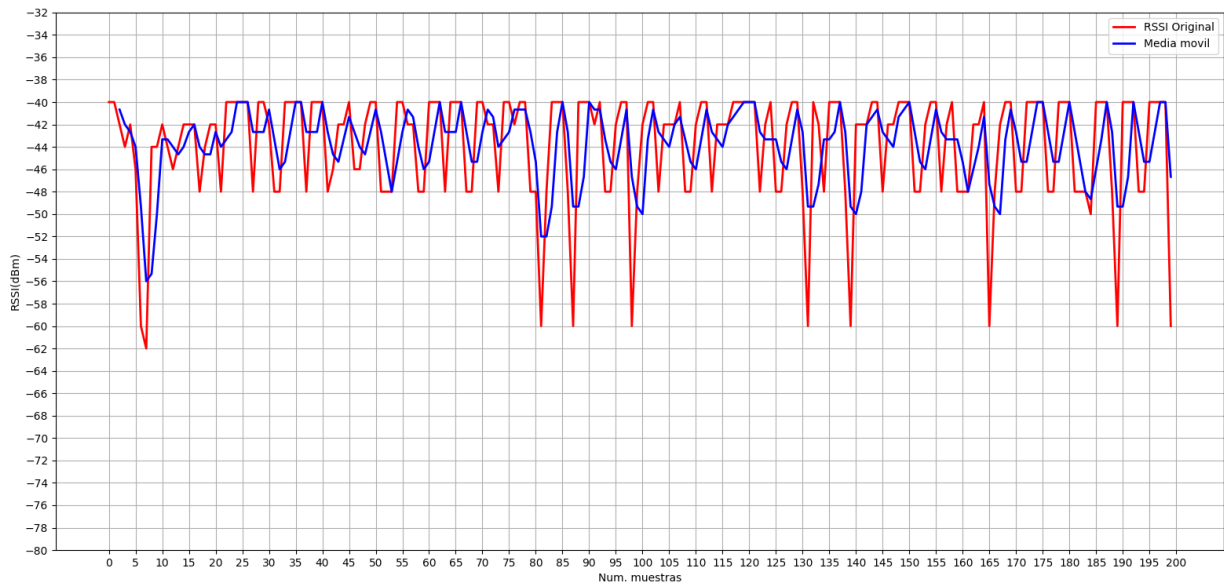
**Figura 7.10:** Resultados obtenidos tras aplicar la media móvil a los datos de ejemplo de la figura 7.10.

Para hacer estas mediciones se ha aprovechado el programa de la figura 7.1 y se le ha añadido el método de la figura 7.9. Además, se ha mantenido la configuración anterior de la baliza. Esta ha sido situada a 2 metros de distancia del sensor. Después de capturar 200 muestras de la señal y aplicar la media móvil, hemos mostrado en la figura 7.11 y 7.12 la señal original y la señal filtrada en forma de gráfico para su interpretación. En estas figuras podemos observar que los resultados obtenidos tras capturar 50 y 200 muestras indican que se ha logrado suavizar la señal. Aunque se logra estabilizar la señal, no parece aún lo suficientemente estable. De manera que, en el siguiente apartado aplicaremos el filtro de Kalman a las señales obtenidas para estudiar si se puede mejorar la estabilidad de estas.

<sup>30</sup> **Figura 7.9:** Método extraído de [56]



**Figura 7.11:** Comparación entre la señal original y la señal filtrada mediante una media móvil (50 muestras).

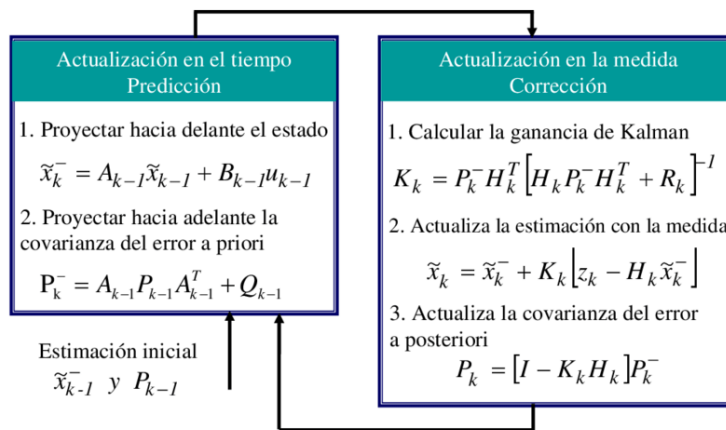


**Figura 7.12:** Comparación entre la señal original y la señal filtrada mediante una media móvil (200 muestras).

### 7.3. Filtro de Kalman

El filtro de Kalman [57] es un algoritmo desarrollado por Rudolf E. Kalman en 1960 que sirve para poder identificar el estado no medible de un sistema dinámico lineal. Es un algoritmo recursivo que puede ser ejecutado en tiempo real utilizando únicamente las mediciones de entrada actuales, el estado calculado previamente y una matriz de incertidumbre. No requiere información adicional.

El filtro de Kalman tiene muchas aplicaciones en tecnología. La aplicación más común es la guía, navegación y control de vehículos, especialmente de naves espaciales. Además, se utiliza ampliamente en campos como el procesamiento de señales y la econometría. En la figura 7.13 podemos ver las etapas del filtro de Kalman.



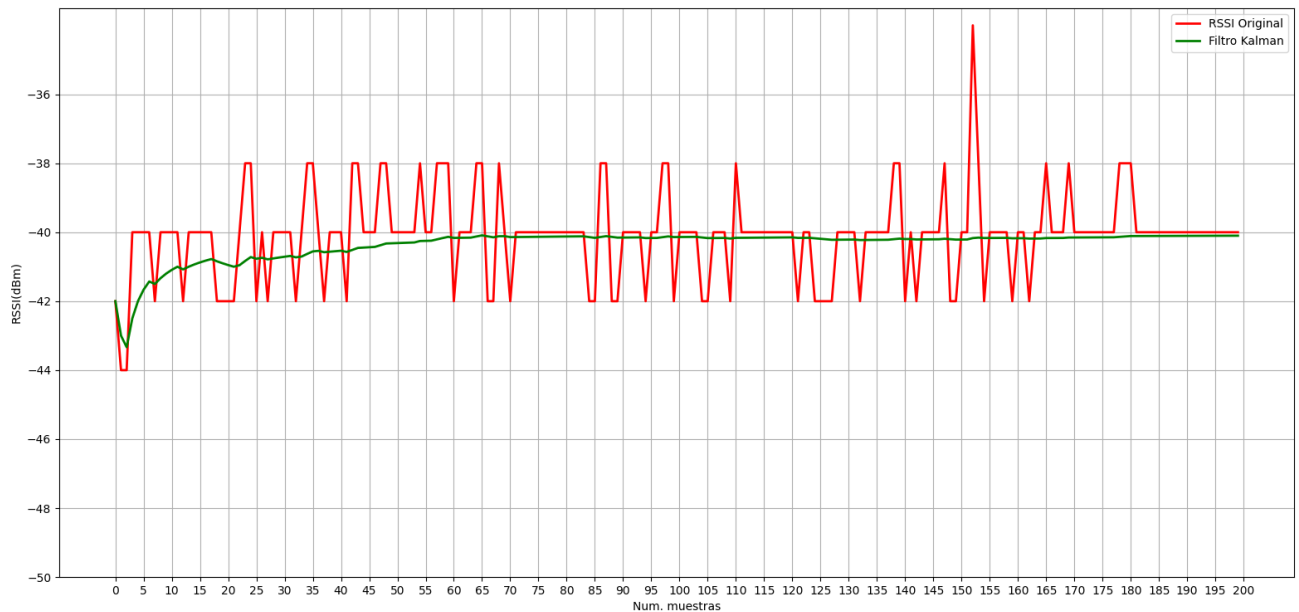
**Figura 7.13:** Etapas del filtro de Kalman.<sup>31</sup>

Para analizar la señal filtrada con el filtro de Kalman se ha desarrollado un programa a partir de la figura 7.1 y la librería *FilterPy* [58]. Este programa aplica el filtrado a la señal a medida que va capturando muestras y, una vez tomadas todas las muestras deseadas, proporciona en forma de gráfico de líneas los valores RSSI obtenidos para su interpretación.

<sup>31</sup> **Figura 7.13:** Extraída de [59]

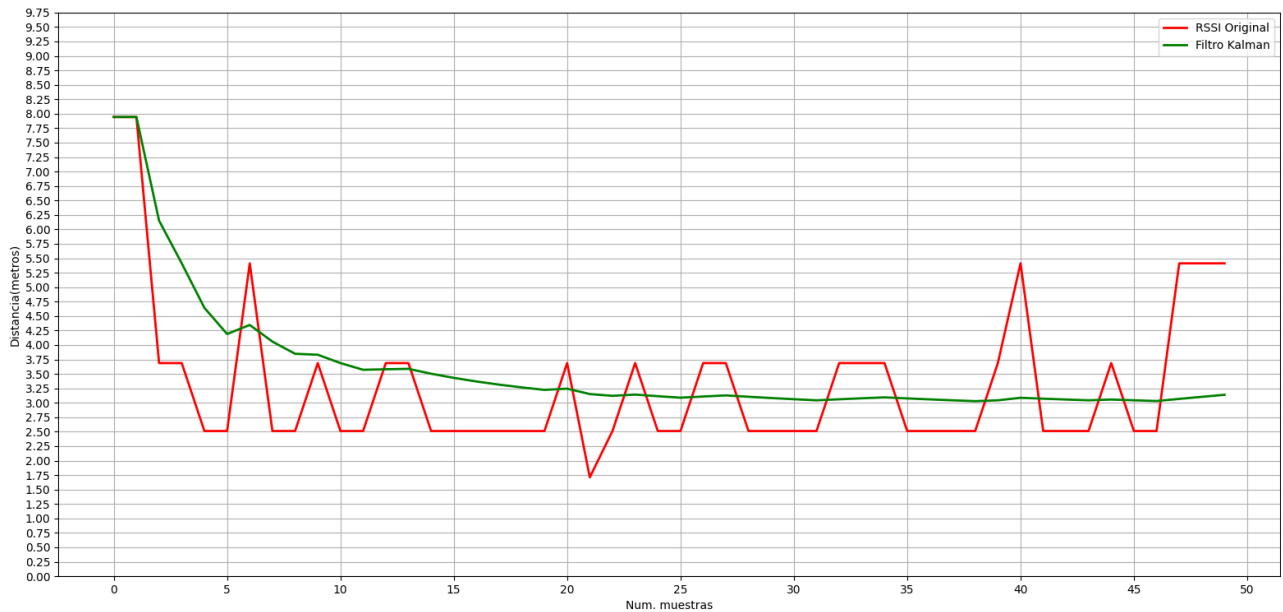


De igual manera que en los análisis anteriores, se ha mantenido la configuración de la baliza indicada al inicio del capítulo 7. Sin embargo, la distancia entre la baliza y el sensor ha sido en esta ocasión de 3 metros. A continuación, después de haber tomado 200 muestras de la señal podemos apreciar en detalle los resultados obtenidos en la figura 7.14. En ella podemos observar que hay una suavización y una clara mejora en la estabilidad de la señal.



**Figura 7.14:** Comparación entre RSSI original y RSSI filtrado con Kalman (200 muestras).

Después de haber visto estos buenos resultados, se modificó el anterior programa para utilizar la fórmula 7.1 y, de esta manera, convertir la señal filtrada a distancia en metros. Después de ejecutar nuevamente el programa, en esta ocasión, configurado para tomar solamente 50 muestras, obtuvimos los resultados que pueden verse en la figura 7.15.

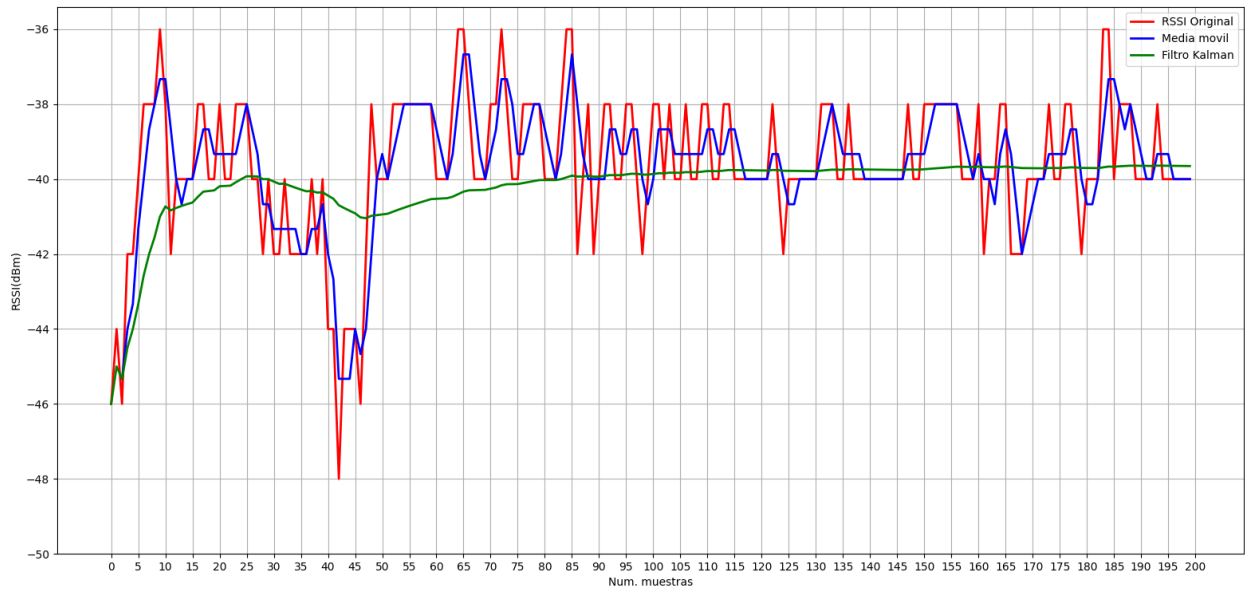


**Figura 7.15:** Evolución de la distancia aproximada con la señal filtrada con el filtro de Kalman para una distancia real de 3 metros (50 muestras).

Como podemos ver en la figura 7.15, los resultados de convertir la señal filtrada a metros fueron muy buenos, ya que las distancias obtenidas a partir de la señal filtrada se mantienen muy cerca de la distancia real de 3 metros.

## 7.4. Filtro estadístico vs filtro Kalman

En este apartado he realizado una comparación de los resultados obtenidos con los distintos filtros que he aplicado. Para ello, se ha creado un programa, a partir de los programas anteriores, que aplica los dos filtros que hemos visto anteriormente y muestra los resultados en un gráfico de líneas. En la figura 7.16 podemos ver los resultados obtenidos para 200 muestras. En esta, podemos ver en color rojo los valores RSSI originales, en color azul y verde podemos ver los valores de la señal filtrada con el filtro estadístico y el filtro de Kalman, respectivamente. En esta figura se observa que, aunque la baliza se encuentra **estática a una distancia fija**, ambos filtros funcionan adecuadamente y logran suavizar la señal y estabilizarla, aunque los resultados del filtro de Kalman son notablemente mejores.



**Figura 7.16:** Comparación entre la señal RSSI original (rojo) y las señales obtenidas tras utilizar el filtro estadístico y el filtro de Kalman (azul y verde respectivamente) en 200 muestras de la señal.

Aunque hemos obtenidos buenos resultados, el filtro de Kalman no es la elección correcta para un sistema con balizas móviles como es el nuestro, ya que no tiene un vector de control adicional para predecir el movimiento. Si las balizas están estáticas, el filtro de Kalman sí que logra un obtener un buen rendimiento, tal y como hemos visto en las pruebas anteriores, pero, para casos dinámicos se necesita tener un vector de control para poder predecir adecuadamente. Para solucionar esto en nuestro futuro SLI, se sugiere estudiar el funcionamiento del filtro de Kalman extendido, mencionado en el apartado 7.5.

## 7.5. Filtro de Kalman extendido

El filtro de Kalman normal está diseñado para generar estimaciones del estado al igual que el filtro de Kalman extendido [60]. Sin embargo, el filtro de Kalman solo funciona cuando el modelo de espacio de estado (es decir, la función de transición de estado) es lineal; es decir, la función que gobierna la transición de un estado al siguiente se puede trazar como una línea en un gráfico). El filtro de Kalman extendido se desarrolló para permitir que el filtro de Kalman se aplique a sistemas que tienen dinámicas no lineales como un robot móvil. Es decir, el filtro de Kalman funciona mejor cuando las balizas están estáticas en una posición y el filtro de Kalman extendido está desarrollado para balizas móviles.

Sería muy interesante analizar los resultados obtenidos a través del filtro de Kalman y el filtro de Kalman extendido, sin embargo, debido a la falta de tiempo, el estudio del filtro de Kalman extendido se ha dejado para estudiarlo en un futuro.

## **7.6. Conclusiones sobre el análisis de la señal**

Como conclusión de este capítulo, podemos decir que con la tecnología *BLE* es realmente difícil estimar la distancia de forma precisa utilizando un solo sensor, aplicando simplemente filtros y realizando una conversión a distancia. La solución a esto podría ser utilizar muchos sensores, tomar medidas en ellos, aplicar uno de los filtros analizados y emplear, por ejemplo, la técnica *fingerprinting* comentada en el apartado 3.2.1. Ahora bien, si la distancia entre los sensores es grande, es difícil obtener buenos resultados debido a la reflexión y la absorción de la señal, por lo que se recomienda poner los sensores próximos como veremos en el siguiente capítulo.

Como en este trabajo no estaba previsto realizar inicialmente el filtrado de la señal, no ha dado tiempo a estudiar el funcionamiento del SLI con la técnica *fingerprinting*, de manera que se plantea como trabajo futuro.

## Capítulo 8

# Ubicación de los sensores y configuración de las balizas

Un aspecto fundamental que determina la obtención de resultados fiables en este tipo de sistemas de localización y sin utilizar demasiada infraestructura, es la correcta ubicación de un número mínimo de sensores y la correspondiente configuración adecuada de las balizas. Este apartado presenta un estudio preliminar sobre la ubicación y la cantidad de los sensores necesarios para el SLI, así como su configuración requerida para la baliza.

### 8.1. Ubicación de los sensores

Para determinar la ubicación y el número de sensores debemos conocer previamente el área en  $m^2$  que es capaz de monitorizar cada sensor. La primera prueba que hemos realizado al respecto consiste en determinar la distancia máxima a la que los sensores pueden captar la señal de las balizas. En la figura 8.1 podemos ver el sensor y la baliza montados en un trípode listos para empezar las pruebas.

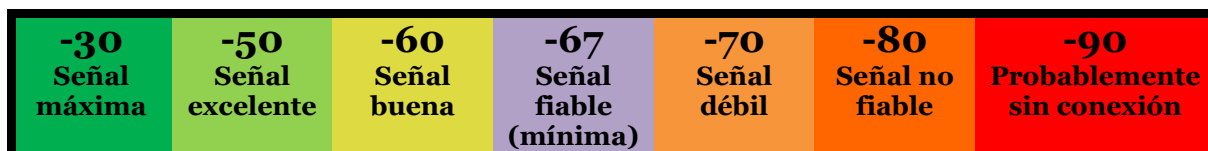


**Figura 8.1:** Sensor elaborado con una Raspberry Pi 4 y baliza iBKS 105 montados en sus respectivos trípodes.

La configuración de las balizas no se ha visto alterada para este análisis. El programa desarrollado a partir de la figura 7.3 se ha modificado para que, después de tomar 50 muestras de RSSI, calcule la media aritmética de todos los valores obtenidos. Para realizar este análisis se ha colocado el sensor en una posición estática y se han tomado tres veces muestras para cada distancia con el objetivo de encontrar la distancia máxima efectiva. En la tabla 8.1 podemos ver los resultados de las mediciones. La tercera columna de la tabla (“RSSI medio”) nos indica la media aritmética de los valores de las 50 muestras RSSI de cada medición.

Distancia	Numero de muestras	RSSI medio
1 metro	50	-54.1 dBm
	50	-54.8 dBm
	50	-54.7 dBm
2 metros	50	-68.2 dBm
	50	-65.1 dBm
	50	-64.9 dBm
5 metros	50	-66.6 dBm
	50	-66.8 dBm
	50	-66.0 dBm
6 metros	50	-70.4 dBm
	50	-70.6 dBm
	50	-69.2 dBm
7 metros	50	-65.5 dBm
	50	-65.2 dBm
	50	-65.4 dBm
8 metros	50	-63.4 dBm
	50	-62.8 dBm
	50	-63.3 dBm
10 metros	50	-67.9 dBm
	50	-67.1 dBm
	50	-67.6 dBm
10 metros	50	-66.4 dBm
	50	-66.1 dBm
	50	-66.3 dBm

**Tabla 8.1:** Resultados de las mediciones para encontrar la distancia máxima a la que los sensores pueden captar la señal de la baliza.



**Figura 8.2:** Valores de intensidad de la señal *BT* (*dBm*) en función de la calidad de esta.

Teniendo en cuenta la escala de valores de la figura 8.2 que muestra la calidad de la señal en función de la intensidad de esta, damos por buenos todos los valores de la señal superiores a -75 dBm. En la figura 8.3 se muestran la media aritmética de las 3 mediciones en cada distancia en formato de gráfico de barras para facilitar su interpretación.



**Figura 8.3:** Resultados en forma de gráfico de barras de las mediciones de la tabla 8.1 para encontrar la distancia máxima a la que los sensores pueden captar la señal de la baliza.

Después de todas las pruebas realizadas en el capítulo 7 habíamos llegado a la conclusión de que, a medida que la distancia aumenta, la señal disminuye. Sin embargo, en esta ocasión nos encontramos con un fenómeno extraño, como podemos ver la figura 8.3, cuando se aumenta la distancia de 5 metros a 6 metros la señal disminuye, como se espera. Sin embargo, al aumentar la distancia de 6 metros a 7 metros la señal mejora en vez de disminuir.

Para entender este fenómeno nos hemos fijado en el entorno donde se han realizado las mediciones. Da la casualidad de que, a la distancia de 6 metros, la baliza estaba junto a una estantería de madera, tal y como se puede apreciar en la figura 8.4.



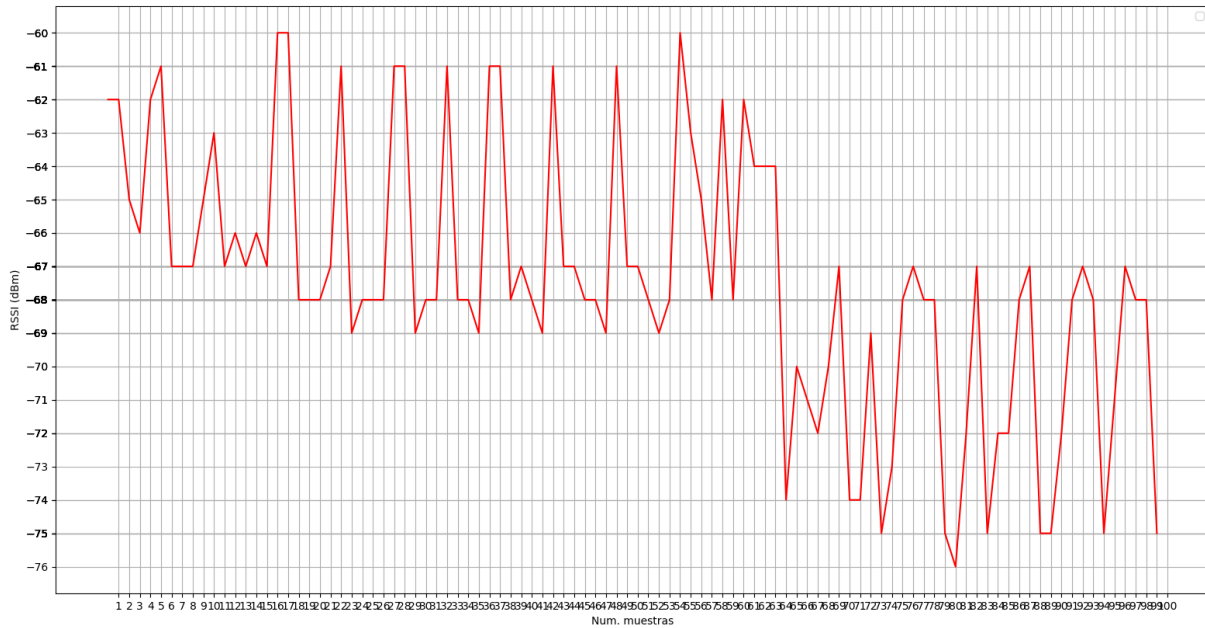
**Figura 8.4:** Baliza situada junto a una estantería de madera, a 6 metros de distancia del sensor.

Por un lado, es posible que aumenten las reflexiones de la señal y empeoren los resultados al estar tan cerca de la superficie del mueble. Por otro lado, debido a que la madera contiene mayor humedad que otros materiales, podría afectar en mayor medida a la señal. Esto se debe a que la tecnología *BT* emite a una frecuencia de 2,4 GHz. Hay mucha proximidad entre esta frecuencia y la supuesta frecuencia natural del agua. Debido a esto, las moléculas de agua entrarían en resonancia absorbiendo mucha más energía que en otras frecuencias.

Si un mueble es capaz de alterar la señal, ¿qué pasará con un cuerpo humano formado por un 70% de agua? Para responder a esto, realizamos una simple prueba donde se mantiene la baliza encerrada entre los dedos de las manos en una distancia de 1,5 metros del sensor.

En la figura 8.5 podemos observar los resultados obtenidos tras capturar 100 muestras de la señal. En las muestras 1 hasta la 50 he dejado la baliza “libre”. A partir de la muestra 51 hasta la 100 se ha encerrado la baliza en la mano, como se muestra en la figura 8.5. En la figura 8.6 podemos observar cómo al encerrar la baliza en la mano la señal se reduce drásticamente, lo que demuestra, efectivamente, que a mayor cantidad de agua peor será la señal.





**Figura 8.5:** RSSI obtenido con la baliza encerrada en la mano (muestras 51 a 100) y fuera de la mano (muestras 1 a 50), a una distancia de 1,5 metros entre baliza y sensor.



**Figura 8.6:** Baliza fuera de la mano y en la mano para interferir en la señal que emite.<sup>32</sup>

Aunque en la tabla 8.1 vemos que el sensor es capaz de captar la señal de la baliza a más de 13 metros, después haber visto cómo la señal puede alterarse fácilmente solamente al aumentar 1 metro de distancia debido a los obstáculos propios del entorno, se recomienda que la distancia entre cada sensor de nuestro SLI no supere los 3 metros para no perder precisión. Sin embargo, cada entorno deberá ser estudiado de manera individual para encontrar la distancia más efectiva.

<sup>32</sup> **Figura 8.6:** Fuente propia

Para nuestro entorno en específico hemos recomendado utilizar una distancia de 1 a 3 metros entre cada sensor. Ahora podemos presentar las bases establecidas en [61], [62] y [63] de un algoritmo que sea capaz de calcular la posición y el número de sensores necesarios. Se van a presentar solo las bases del algoritmo, ya que la implementación del mismo no forma parte del alcance del proyecto.

La solución va a consistir en un listado de coordenadas, cada una de las cuales tendrían un formato  $X_i = (x_i, y_i, z_i)$  correspondiente a la posición del sensor, con  $i = 1, 2, 3, \dots, N$ , siendo  $N$  el número total de sensores de la solución que minimizaría el coste. La ubicación de estos sensores es la respuesta a un proceso que busca la optimización de los siguientes parámetros:

**Minimizar el número de sensores:** Para un área determinada habría que obtener el número mínimo de sensores necesarios para tener el cien por cien de cobertura.

**Maximizar la cobertura en el área de trabajo:** Se necesitaría encontrar la configuración de sensores que den la máxima cobertura cumpliendo con los demás parámetros.

El algoritmo debe buscar configuraciones de sensores cuya ubicación permita obtener resultados de posición confiables al realizar los cálculos, libres de singularidades y manteniendo la cobertura adecuada. Para ello, es fundamental determinar primero las características del área y la cobertura deseada donde implementar el sistema, con el fin de establecer los parámetros iniciales que definen el funcionamiento del algoritmo.

## 8.2. Configuración de las balizas

En todas las pruebas que se han realizado con las balizas, la configuración utilizada ha sido una potencia de transmisión de -14 dBm y una frecuencia de emisión de 100 ms. Inicialmente este proyecto pretendía encontrar la configuración óptima para las balizas de nuestro SLI en relación a la duración de la batería y al correcto funcionamiento del SLI. Debido a la falta de tiempo causada por las tareas imprevistas, solamente se va a presentar un estudio preliminar y una propuesta de configuración.

Para determinar esta configuración para las balizas se han tenido en cuenta las siguientes consideraciones:

- La frecuencia de emisión es muy importante para que el sensor detecte correctamente la señal ya que puede haber emisiones que no hayan sido detectadas. Por tanto, se sugiere utilizar una frecuencia de emisión de 100 milisegundos.
- La potencia de emisión de la señal no afecta de manera tan drástica al comportamiento de las señales.

En la figura 8.2 hemos considerado como señal aceptable todas las señales mayores que -75 dBm. En la hoja de especificaciones técnicas de la baliza [42] se sugiere que la potencia de emisión sea de -12 dBm. La figura 8.7 muestra la potencia recibida en dBm en comparación con la distancia en metros para todas las potencias TX configurables. En esta figura podemos observar que, para una distancia de 3 metros, la potencia de emisión -12 dBm nos proporcionaría -74 dBm de señal. Además, se sugiere establecer una potencia de emisión baja, como esos -12 dBm, para compensar el consumo de energía debido a la alta frecuencia de emisión que sugerimos, de 100 milisegundos.

Distance (m)	TX Power (dBm)							
	-30	-20	-16	-12	-8	-4	0	+4
0	-63	-47	-42	-40	-40	-34	-27	-23
1	-92	-79	-71	-68	-66	-58	-59	-57
3	-99	-84	-82	-74	-73	-70	-67	-62
5	-101	-90	-88	-84	-82	-74	-71	-67
10		-95	-95	-89	-87	-81	-79	-75
15		-98	-97	-95	-92	-90	-84	-82
20		-102	-102	-97	-94	-92	-87	-85
30				-102	-96	-93	-92	-88
40					-99	-99	-94	-91
50					-102	-102	-100	-99

**Figura 8.7:** Potencia recibida en relación a la potencia de emisión (TX) y a la distancia entre el sensor y la baliza iBKS 105.<sup>33</sup>

<sup>33</sup> **Figura 8.7:** Extraída de [42]



# Capítulo 9

## Conclusiones

En este proyecto se han desarrollado una serie de estudios orientados a poder implementar en el futuro de manera eficiente un sistema de localización de interior basado en el despliegue de balizas *bluetooth low energy*. Se ha realizado un análisis profundo sobre el *hardware* de sensorización más adecuado para el SLI y también se han llevado a cabo complicados estudios sobre las señales emitidas por una baliza para poder obtener posteriormente una buena precisión en la localización de esta. El desarrollo del proyecto ha sido un éxito ya que se han podido realizar la mayoría de los estudios planificados. No obstante, debido a la gran complejidad de factores que existen relacionados con la señal, deberían realizarse más estudios con el filtro de Kalman extendido junto a la técnica de *fingerprinting* para encontrar el método más óptimo de filtrado de la señal y de estimación de la posición del objeto a localizar.

Por otro lado, la gestión del proyecto, ha sido satisfactoria. Aunque la planificación inicial se ha visto modificada debido a que han surgido tareas no previstas inicialmente, la metodología elegida ha permitido adaptar estos imprevistos.

Por último, a nivel personal, considero que este proyecto y estancia me ha aportado muchos conocimientos tanto técnicos como de organización para enfrentarme a problemas y desafíos imprevistos que surgen en el desarrollo de proyectos de este tipo. Por otra parte, se ha echado en falta trabajar en equipo y la colaboración de manera cercana con los clientes debido a la situación de emergencia sanitaria causada por el COVID-19.



# Bibliografía

- [1] Ubik Geospatial Solutions. *UBIK Geospatial Solutions S.L. helps you manage your location-based information*. [Consulta: 22 de febrero de 2021]. (2021). <http://www.ubikgs.com/>
- [2] Colaboradores de Wikipedia. *Baliza electrónica*. *Wikipedia, la enciclopedia libre*. [Consulta: 22 de febrero de 2021]. (2020, 14 diciembre). [https://es.wikipedia.org/wiki/Baliza\\_electr%C3%B3nica](https://es.wikipedia.org/wiki/Baliza_electr%C3%B3nica)
- [3] Ligeró, R. (2021, 6 julio). *Beacons: el elemento fundamental del posicionamiento en interiores*. *Accent Systems*. [Consulta: 22 de febrero de 2021]. <https://accent-systems.com/es/beacons-elemento-fundamental-posicionamiento-interiores/>
- [4] Jesús Ruiz Ruiz, A. (2009) *Sistemas de localización geográfica en entornos de interior basados en tecnologías de comunicación inalámbricas*. [Consulta: 22 de febrero de 2021]. <https://webs.um.es/oceanovas/miwiki/lib/exe/fetch.php?media=indoor-location.pdf>
- [5] Díaz-Ambrona Tabernilla, Luis (2008). *Sistema de localización en interiores. Proyecto Fin de Carrera / Trabajo Fin de Grado, Facultad de Informática (UPM) [antigua denominación], Madrid, España*. [Consulta: 01 de marzo de 2021]. [http://oa.upm.es/947/1/PFC\\_LUIS\\_DIAZ\\_AMBRONA.pdf](http://oa.upm.es/947/1/PFC_LUIS_DIAZ_AMBRONA.pdf)
- [6] IONOS Digitalguide. (2021, 20 julio). *El modelo en cascada: desarrollo secuencial de software*. [Consulta: 01 de marzo de 2021]. <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>
- [7] Gilibets, L. (2020, 11 noviembre). *Qué es Kanban y cómo utilizarlo en el desarrollo de proyectos*. *Thinking for Innovation*. [Consulta: 01 de marzo de 2021]. <https://www.iebschool.com/blog/metodologia-kanban-agile-scrum/>
- [8] Colaboradores de Wikipedia. (2021, 22 julio). *GPS*. *Wikipedia, la enciclopedia libre*. [Consulta: 01 de marzo de 2021]. <https://es.wikipedia.org/wiki/GPS>
- [9] García Polo, Eva M. (2011, 11 diciembre). *Técnicas de Localización en Redes Inalámbricas de Sensores*. [Consulta: 01 de marzo de 2021]. <https://www.dsi.uclm.es/personal/EvaMariaGarcia/docs/2008-Curso%20Verano.pdf>

- [10] Wikipedia contributors. (2021, febrero 24). *Time of arrival*. *Wikipedia*. [Consulta: 01 de marzo de 2021]. [https://en.wikipedia.org/wiki/Time\\_of\\_arrival](https://en.wikipedia.org/wiki/Time_of_arrival)
- [11] (PubMed Central (PMC)). (2019, 1 Junio). *Time Difference of Arrival (TDoA) Localization Combining Weighted Least Squares and Firefly Algorithm*. [Consulta: 08 de marzo de 2021]. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6603714/>
- [12] Wikipedia contributors. (2021, Abril 17). *Angle of arrival*. *Wikipedia*. [Consulta: 08 de marzo de 2021]. [https://en.wikipedia.org/wiki/Angle\\_of\\_arrival](https://en.wikipedia.org/wiki/Angle_of_arrival)
- [13] Subedi, S. (2017, 31 diciembre). *Practical Fingerprinting Localization for Indoor Positioning System by Using Beacons*. *Hidawi - Journal of Sensors*. [Consulta: 08 de marzo de 2021]. <https://www.hindawi.com/journals/js/2017/9742170/>
- [14] IBM. (2021). *Análisis vecino más cercano*. [Consulta: 08 de marzo de 2021]. <https://www.ibm.com/docs/es/spss-statistics/SaaS?topic=features-nearest-neighbor-analysis>
- [15] Colaboradores de Wikipedia. (2021, 19, febrero 19). *Trilateración*. *Wikipedia, la enciclopedia libre*. [Consulta: 08 de marzo de 2021]. <https://es.wikipedia.org/wiki/Trilateraci%C3%B3n>
- [16] Wikipedia contributors. (2021, abril 8). *Triangulation*. *Wikipedia*. [Consulta: 15 de marzo de 2021]. <https://en.wikipedia.org/wiki/Triangulation>
- [17] Colaboradores de Wikipedia. (2021, julio 10). *Sistema global para las comunicaciones móviles*. *Wikipedia, la enciclopedia libre*. [Consulta: 15 de marzo de 2021]. [https://es.wikipedia.org/wiki/Sistema\\_global\\_para\\_las\\_comunicaciones\\_m%C3%B3viles](https://es.wikipedia.org/wiki/Sistema_global_para_las_comunicaciones_m%C3%B3viles)
- [18] Wikipedia contributors. (2021, marzo 15). *LPWAN*. *Wikipedia*. [Consulta: 15 de marzo de 2021]. <https://en.wikipedia.org/wiki/LPWAN>
- [19] Wikipedia contributors. (2021, julio 11). *LoRaWAN*. *Viquipèdia, l'enciclopèdia lliure*. [Consulta: 15 de marzo de 2021]. <https://ca.wikipedia.org/wiki/LoRaWAN>
- [20] Colaboradores de Wikipedia. (2021, abril 16). *Sigfox*. *Wikipedia, la enciclopedia libre*. [Consulta: 15 de marzo de 2021]. <https://es.wikipedia.org/wiki/Sigfox>



- [21] Hernández, L. D. V. (2020, 25 mayo). *GPS Arduino localiza tus proyectos en todo el mundo. Programar fácil con Arduino*. [Consulta: 22 de marzo de 2021]. <https://programarfácil.com/podcast/gps-arduino-geolocalizacion>
- [22] Moes, T. (2014). *¿Qué es el Bluetooth y para qué sirve?*. *SoftwareLab*. [Consulta: 22 de marzo de 2021]. <https://softwarelab.org/es/bluetooth/>
- [23] Colaboradores de Wikipedia. (2021, mayo 25). *Bluetooth de baja energía*. *Wikipedia, la enciclopedia libre*. [Consulta: 22 de marzo de 2021]. [https://es.wikipedia.org/wiki/Bluetooth\\_de\\_baja\\_energ%C3%ADa](https://es.wikipedia.org/wiki/Bluetooth_de_baja_energ%C3%ADa)
- [24] M. Hazas, J. Scott y J. Krumm, «*Location-aware computing comes of age*, » [Consulta: 22 de marzo de 2021]. (*Computer*, vol. 37, n<sup>o</sup> 2, pp. 95-97, 2004)
- [25] Ruiz, P. (2021, 6 julio). *Aprende Bluetooth Low Energy – Pt. 2 - Siete Segmentos. Siete Segmentos - Blog de Tecnología en español*. [Consulta: 22 de marzo de 2021]. [https://sietesegmentos.com/2021/07/03/ble\\_2/](https://sietesegmentos.com/2021/07/03/ble_2/)
- [26] R. (2013, 10 junio). *¿Demasiados Cocineros en la Cocina de 2,4 GHz? | Revista Española de Electrónica*. *Revista Española de Electrónica | Todas las noticias de electrónica actualizadas a diario*. [Consulta: 29 de marzo de 2021]. <https://www.redeweb.com/articulos/instrumentacion/demasiados-cocineros-en-la-cocina-de-2-4-ghz/>
- [27] Afaneh, M. (2020, 21 Abril). *How Bluetooth Low Energy Works: Advertisements (Part 1)*. *Novel Bits*. [Consulta: 29 de marzo de 2021]. <https://www.novelbits.io/bluetooth-low-energy-advertisements-part-1/>
- [28] Pérez, E. (2019, 12 octubre). *Qué ventajas tiene la tecnología de banda ultraancha (UWB): una vieja alternativa al Bluetooth que Apple*. *Xataka*. [Consulta: 29 de marzo de 2021]. <https://www.xataka.com/servicios/que-ventajas-tiene-tecnologia-banda-ultrancha-uwv-vieja-alternativa-al-bluetooth-que-apple-samsung-estan-revitalizando>
- [29] Colaboradores de Wikipedia. (2021, julio 22). *Wifi*. *Wikipedia, la enciclopedia libre*. [Consulta: 29 de marzo de 2021]. <https://es.wikipedia.org/wiki/Wifi>
- [30] Colaboradores de Wikipedia. (2021, enero 2). *Zigbee*. *Wikipedia, la enciclopedia libre*. [Consulta: 29 de marzo de 2021]. <https://es.wikipedia.org/wiki/Zigbee>

- [31] Colaboradores de Wikipedia. (2021, mayo 27). *Ultrasonido*. Wikipedia, la enciclopedia libre. [Consulta: 26 de Julio de 2021]. <https://es.wikipedia.org/wiki/Ultrasonido>
- [32] MokoSmart. (2021, 14 Julio). *A Comprehensive Guide about iBeacon Technology*. MOKOSmart. [Consulta: 05 de abril de 2021]. <https://www.mokosmart.com/es/ibeacon-technology/>
- [33] Wikipedia contributors. (2021, Julio 21). *Eddystone (Google)*. Wikipedia. [Consulta: 05 de abril de 2021] [https://en.wikipedia.org/wiki/Eddystone\\_\(Google\)](https://en.wikipedia.org/wiki/Eddystone_(Google))
- [34] Blackstone, A. (2020). *Understanding the different types of BLE Beacons | Mbed*. ARM- Bed. [Consulta: 05 de abril de 2021]. <https://os.mbed.com/blog/entry/BLE-Beacons-URIBeacon-AltBeacons-iBeacon/>
- [35] Hernandez, D. Fernández-Caramés, T. J Escudero, C. Fraga-Lamas, P. (2017, December) *Design and Practical Evaluation of a Family of Lightweight Protocols for Heterogeneous Sensing through BLE Beacons in IoT Telemetry Applications*. [Consulta: 05 de abril de 2021]. [https://www.researchgate.net/publication/322092785\\_Design\\_and\\_Practical\\_Evaluation\\_of\\_a\\_Family\\_of\\_Lightweight\\_Protocols\\_for\\_Heterogeneous\\_Sensing\\_through\\_BLE\\_Beacons\\_in\\_IoT\\_Telemetry\\_Applications](https://www.researchgate.net/publication/322092785_Design_and_Practical_Evaluation_of_a_Family_of_Lightweight_Protocols_for_Heterogeneous_Sensing_through_BLE_Beacons_in_IoT_Telemetry_Applications)
- [36] Gracia, L. (2016, 4 agosto). *Un poco de Beacons: iBeacon vs Eddystone*. Un poco de Java. [Consulta: 12 de abril de 2021]. <https://unpocodejava.com/2016/08/04/un-poco-de-beacons-ibeacon-vs-eddystone/>
- [37] Particle. (2020). *Particle Argon*. [Consulta: 12 de abril de 2021]. <https://docs.particle.io/argon/>
- [38] Espressif. (2020). *ESP32-DevKitC V4 Getting Started Guide - ESP32 - — ESP-IDF Programming Guide latest documentation*. Espressif Official Website. [Consulta: 12 de abril de 2021]. <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-devkitc.html>
- [39] Adafruit Industries. (2020). *Adafruit Feather MO Bluefruit LE*. Adafruit. [Consulta: 12 de abril de 2021]. <https://www.adafruit.com/product/2995>
- [40] Particle. (2020). *Particle Boron*. [Consulta: 12 de abril de 2021]. <https://docs.particle.io/boron/>

- [41] The Raspberry Pi Foundation. (2021). *Raspberry Pi 4 Model B*. [Consulta: 12 de abril de 2021]. <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/>
- [42] Accent Systems. (2021, 8 Julio). *iBKS 105*. [Consulta: 12 de abril de 2021]. <https://accent-systems.com/es/producto/ibks-105/>
- [43] Accent Systems. (2021b, julio 20). *Accent Systems*. [Consulta: 12 de abril de 2021]. <https://accent-systems.com/es/>
- [44] Bowdentheo, B. (2020, November). *GitHub - bowdentheo/BLE-Beacon-Scanner: Scan for BLE Beacons (iBeacon and Eddystone) using Python. Designed to be used with a Raspberry Pi 3 B+*. *GitHub*. [Consulta: 12 de abril de 2021]. <https://github.com/bowdentheo/BLE-Beacon-Scanner>
- [45] Matplotlib. (2021, mayo). *Matplotlib: Python plotting — Matplotlib 3.4.2 documentation*. [Consulta: 12 de abril de 2021]. <https://matplotlib.org/>
- [46] Cursos abiertos universidad. (2020). *Gráficas en Python y Matplotlib*. [Consulta: 19 de abril de 2021]. [https://www.youtube.com/watch?v=8Pwh4zjI9cg&ab\\_channel=CursosAbiertosUniversidad](https://www.youtube.com/watch?v=8Pwh4zjI9cg&ab_channel=CursosAbiertosUniversidad)
- [47] Speecheck. (2020). *¿Qué es un RSSI?* [Consulta: 19 de abril de 2021]. <https://www.speedcheck.org/es/wiki/rssi/>
- [48] Dinomycle Blogspot. (2011, 20 Abril). *Difference between RSSI and RSS or RSS vs RSSI*. [Consulta: 19 de abril de 2021]. (<http://dinomycle.blogspot.com/2011/04/difference-between-rssi-and-rss.html>)
- [49] Wikipedia contributors. (2021, Julio 14). *DBm*. *Wikipedia*. [Consulta: 19 de abril de 2021]. <https://en.wikipedia.org/wiki/DBm>
- [50] Easyvocabforall. (2021, October 7). *How to Calculate Distance from the RSSI value of the BLE Beacon*. [Consulta: 19 de abril de 2021]. <https://iotandelectronics.wordpress.com/2016/10/07/how-to-calculate-distance-from-the-rssi-value-of-the-ble-beacon/>
- [51] Dzone article. (2013). *Formula to convert the rssi value of the blue ble bluetooth*. [Consulta: 19 de abril de 2021]. <https://dzone.com/articles/formula-to-convert-the-rssi-value-of-the-ble-bluet>

- [52] Chris Downey (2013), *Understanding wireless range calculations*. [Consulta: 19 de abril de 2021].  
<https://www.electronicdesign.com/technologies/communications/article/21796484/understanding-wireless-range-calculations>
- [53] Whelan, C. (2021, 21 febrero). *GitHub - pydata/bottleneck: Fast NumPy array functions written in C*. *GitHub*. [Consulta: 19 de abril de 2021]. <https://github.com/pydata/bottleneck>
- [54] Numpy. (2021). *NumPy*. *Numpy Official Website*. [Consulta: 19 de abril de 2021].  
<https://numpy.org>
- [55] Colaboradores de Wikipedia. (2021b, enero 11). *Media móvil*. *Wikipedia, la enciclopedia libre*. [Consulta: 19 de abril de 2021]. [https://es.wikipedia.org/wiki/Media\\_m%C3%B3vil](https://es.wikipedia.org/wiki/Media_m%C3%B3vil)
- [56] Narula, M. (2021, 22 junio). *Media móvil para NumPy Array en Python*. *Delft Stack*. [Consulta: 26 de abril de 2021]. <https://www.delftstack.com/es/howto/python/moving-average-python/>
- [57] Wikipedia contributors. (2021, Julio 19). *Kalman filter*. *Wikipedia*. [Consulta: 26 de abril de 2021]. [https://en.wikipedia.org/wiki/Kalman\\_filter](https://en.wikipedia.org/wiki/Kalman_filter)
- [58] Labbe, R. (2016). *KalmanFilter — FilterPy 1.4.4 documentation*. *FilterPy*. [Consulta: 26 de abril de 2021]. <https://filterpy.readthedocs.io/en/latest/kalman/KalmanFilter.html>
- [59] Angel, L. (2005, enero). *Control visual de robots paralelos: análisis, desarrollo y aplicación a la plataforma RoboTenis*. [Consulta: 26 de abril de 2021].  
[https://www.researchgate.net/publication/39425275\\_Control\\_visual\\_de\\_robots\\_paralelos\\_analisis\\_desarrollo\\_y\\_aplicacion\\_a\\_la\\_plataforma\\_RoboTenis](https://www.researchgate.net/publication/39425275_Control_visual_de_robots_paralelos_analisis_desarrollo_y_aplicacion_a_la_plataforma_RoboTenis)
- [60] Sala, A. (2021, 30 mayo). *Filtro de Kalman extendido: teoría*. *Universidad Politécnica de Valencia*. [Consulta: 26 de abril de 2021].  
<http://personales.upv.es/asala/DocenciaOnline/Video/ekfteo.html>
- [61] J.O. Road, A.R. Jiménez, F. Seco, C. Prieto, J.L. Ealo, F. Ramón, J. Guevara. (2007). *Distribución optimizada de balizas para sistemas de localización fundamentos en Trilateración Esférica e Hiperbólica* [Consulta: 26 de abril de 2021].  
[http://www.car.upmcsic.es/lopsi/static/publicaciones/Congreso%20Internacional/MAEB\\_Roa2007.pdf](http://www.car.upmcsic.es/lopsi/static/publicaciones/Congreso%20Internacional/MAEB_Roa2007.pdf)

- [62] J.O. Road, A.R. Jiménez, F. Seco, C. Prieto, J.L. Ealo, F. Ramón, J. Guevara. (2005). *Primeros resultados en la optimización de la ubicación de balizas para localización utilizando algoritmos genéticos*. [Consulta: 26 de abril de 2021]. <http://www.car.upmcsic.es/lopsi/static/publicaciones/Congreso%20Nacional/XXVIJAutomRoa2005.pdf>
- [63] Cejas, Agustín, Andrés, Chardon. (2019, abril). *Algoritmos avanzados de posicionamiento en interiores utilizando la combinación de distintos tipos de sensores*. [Consulta: 26 de abril de 2021]. <https://www.ridaa.unicen.edu.ar/xmlui/bitstream/handle/123456789/2046/Tesis%20Cejas-Chardon.pdf?sequence=1&isAllowed=y>

