



# UNIVERSIDAD DE GRANADA

UNIVERSIDAD DE GRANADA  
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA  
INFORMÁTICA Y DE TELECOMUNICACIONES

---

## Localización planar mediante protocolo Bluetooth

---

*Autores*

Andoni Pérez Segura  
José Miguel Pérez Navarrete

10 de febrero de 2021

# Índice general

<b>1. Descripción breve del proyecto</b>	<b>2</b>
1.1. Motivación del proyecto . . . . .	2
1.2. Objetivo del proyecto . . . . .	2
<b>2. Desarrollo del proyecto</b>	<b>3</b>
2.1. Estado de arte . . . . .	3
2.1.1. Tecnología Bluetooth Low Energy . . . . .	3
2.1.2. Filtro de Kalman . . . . .	4
2.1.3. Concepto de <i>Fingerprint</i> . . . . .	5
2.2. Estrategia del proyecto . . . . .	5
2.3. Especificaciones y observaciones del Hardware aplicado . . . . .	6
2.3.1. Periféricos . . . . .	6
2.3.2. Arduino . . . . .	7
2.3.3. Raspberry Pi . . . . .	7
2.3.4. Presupuesto del Hardware . . . . .	7
2.4. Desarrollo del Software del proyecto . . . . .	8
2.4.1. Monitorización de las balizas . . . . .	8
2.4.2. Flujo de datos entre las balizas con módulo central . . . . .	9
2.4.3. Funciones del módulo principal . . . . .	9
<b>3. Experimentación, resultados y conclusiones</b>	<b>11</b>
3.1. Lugar de pruebas . . . . .	11
3.2. Fingerprint . . . . .	12
3.3. Procesamiento de tramas . . . . .	12
3.4. Cálculo de la localización . . . . .	12
3.5. Conclusiones . . . . .	13
<b>4. Perspectiva al futuro</b>	<b>14</b>

# Capítulo 1

## Descripción breve del proyecto

### 1.1. Motivación del proyecto

Dada la situación actual ocasionada por la COVID-19, es una tarea esencial el rastreo de posibles brotes dentro de una organización empresa, o conferencias. Un rastreo a tiempo real del flujo de personas dentro de unas instalaciones supone una actuación rápida y prudente.

También, la evolución del flujo de personas dentro de una instalación facilita un estudio de las posibilidades de aglomeración, siendo más factible la planificación de otras rutas donde la aglomeración se reduzca.

Adicionalmente, la localización planar o 3D del personal en instalaciones privadas supone una mayor seguridad e incluso, la posibilidad de personalizar una serie de aplicaciones IoT según la zona en donde se sitúe.

### 1.2. Objetivo del proyecto

A partir de las motivaciones descritas anteriormente, se fijaron los siguientes objetivos principales del proyecto:

- Localización planar de un individuo dentro de unas instalaciones. A extensión de este objetivo, se pretende alcanzar una localización 3D.
- Creación de una lista de los sitios o puntos que ha recorrido, dibujando su trayectoria en un determinado rango de tiempo.
- Monitorización del grado de la aglomeración de cada zona de las instalaciones.

Como extensión de los objetivos principales, se propone una serie de objetivos secundarios, denominado como deseables, yendo más allá de los requisitos mínimos.

- Localización 3D de unas instalaciones.
- Creación de lista de contactos de cada individuo, adjunto a un tiempo estimado de contacto.

# Capítulo 2

## Desarrollo del proyecto

En este capítulo se especifica cada uno de los aspectos a tratar, abarcando tanto la investigación previa como el desarrollo de una solución frente a los objetivos descritos anteriormente. También se incluye la justificación de cada decisión tomada, analizando sus pros y contras.

### 2.1. Estado de arte

Este proyecto se fundamenta principalmente en una serie de conceptos, técnicas y tecnologías aplicadas a lo largo de su desarrollo. Dichos puntos son los siguientes.

#### 2.1.1. Tecnología Bluetooth Low Energy

*Bluetooth Low Energy* (BLE) es una de las versiones más nuevas y recientes del protocolo *Bluetooth* que, al igual que la versión clásica, opera en 2.4 GHz con Frequency-Hopping Spread Spectrum (FHSS) sobre sus canales de transmisión, que son 40 canales y pueden operar una velocidad de transmisión máxima de 2 MB/s con bajo consumo. *Bluetooth Classic* se diferencia de ello con 79 canales y con velocidades de transmisión de hasta 3 MB/s, con un consumo mayor que BLE.

Sin embargo, el mayor atractivo de Bluetooth Low Energy, lo que le hace totalmente distinguible, no es solo la posibilidad de obtener una mejores prestaciones mediante una mayor optimización del consumo, sino la posibilidad que con dispositivos de bajo consumo, de poder crear topologías de red en malla (Mesh) y de difusión (Broadcast). Esta característica principal es la que ha llegado a impulsar el sector IoT.

La estructura de Bluetooth Low Energy se centra principalmente en el clásico modelo de red "*Publish-and-Subscribe*" definiendose 2 tipos de roles: *Periférico* y *Central*. Un dispositivo de rol *Pérférico*, que actúa como un servidor, se centra en el uso de la sub-capa *Generic Attribute Profile* (GATT) para la definición de una serie de servicios diferenciados. Cada servicio está relacionado con una serie de características, que puede ser un parámetro específico del dispositivo, una lectura de un sensor o un dato calculado. Dichas características pueden llegar a actualizarse hasta notificar sobre el cambio. Por otro lado, un dispositivo de rol *Central*, que actúa como cliente, se centra en la solicitud de las características de un servicio de un periférico.

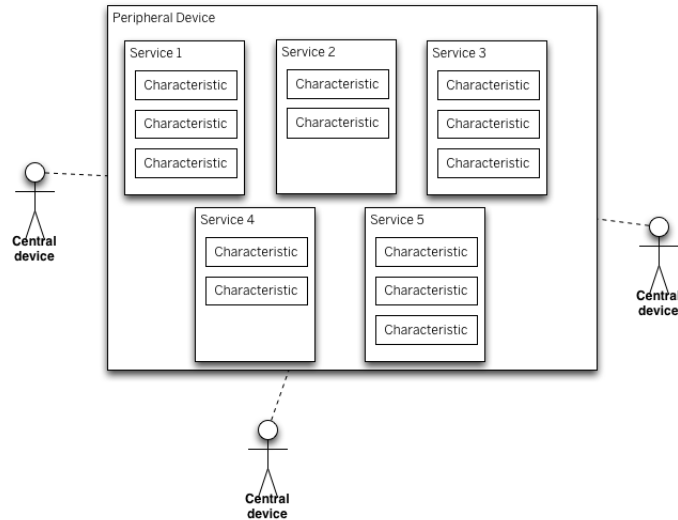


Figura 2.1: Estructura de red de *Bluetooth Low Energy*

Adicionalmente, para la conexión entre central y periférico es necesario la inicialización de dicha conexión mediante la subcapa *General Advertising Profile* (GAP) y posteriormente, se vinculan los dispositivos para operar con las mayores prestaciones posibles. Sin embargo, no es necesaria la vinculación, pudiendo operar solo a partir de GAP, con prestaciones más limitadas.

Un aspecto también a tratar es el rango de distancia que ofrece *Bluetooth Low Energy*. Existe 4 clases distintas, según la potencia de transmisión. Teóricamente, se obtienen los siguientes rangos, en exteriores, mostrados en la tabla 2.1.

	Distancia	Potencia
Clase 1	100 m	100 mW
Clase 2	20 m	2.5 mW
Clase 3	1 m	1 mW
Clase 4	0.5 m	0.5 mW

Cuadro 2.1: Clases de Bluetooth

Actualmente, en el tiempo de la redacción de esta memoria, la página web oficial de Bluetooth ofrece un estimador del rango de bluetooth según la sensibilidad de receptores, el entorno, la potencia de transmisión y las ganancias de la antena de transmisión y de recepción. En la bibliografía se encuentra una dirección *url* del estimador.

### 2.1.2. Filtro de Kalman

Como es habitual en comunicaciones inalámbricas, la potencia de la señal recibida está sujeta a desvanecimientos y ruido introducidos por el canal. Es por esto, que el valor de esta potencia puede sufrir ligeros cambios aún estando en la misma posición. En un sistema donde la precisión de adquisición de los datos de potencia, como el expuesto en este trabajo, es esencial para el correcto funcionamiento del mismo, se hace necesario utilizar filtros para el suavizado de estas muestras.

En este trabajo se ha decidido utilizar el filtro de Kalman. Con este tipo de filtro se consigue el suavizado de las muestras de entrada al calcular las muestras filtradas basándose en las de entrada y sus predicciones. Se podría decir que el filtro trabaja en un bucle continuo de predicción - corrección. Cada muestra filtrada va a depender tanto de la muestra de entrada como de la predicción realizada para la muestra anterior. Para aclarar la forma de trabajar del filtro se expondrá un ligero ejemplo. Si consideramos las muestras de entrada  $[-35 -40 -36 -38 -38 -34 -40 -50 -37 -35]$ , valores de potencia típicos, vemos como las ligeras desviaciones son corregidas por el filtro.

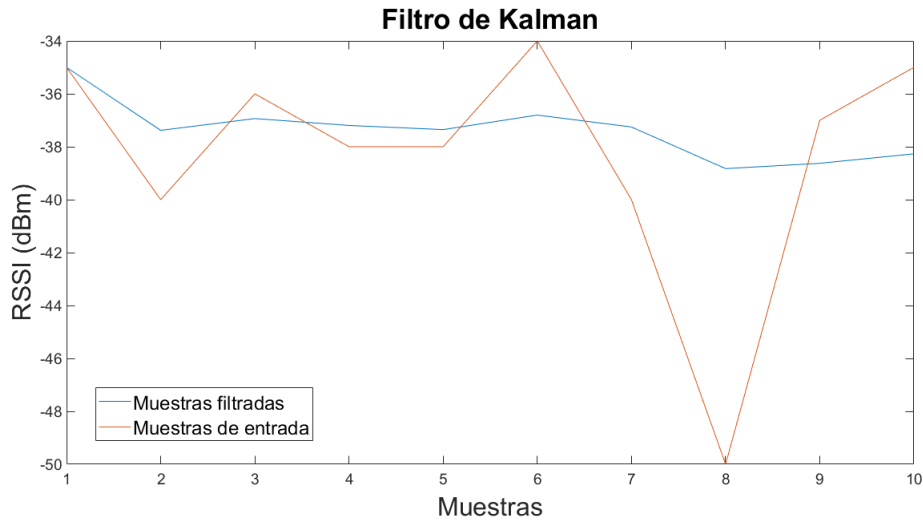


Figura 2.2: Ilustración del uso de filtro de Kalman

### 2.1.3. Concepto de *Fingerprint*

La técnica *Fingerprint* es una técnica de localización basada en la caracterización de posiciones a partir de una serie de parámetros, sean estos principalmente la potencia recibida (RSSI) de una serie de balizas colocado en el campo a monitorizar. Dicha serie de características actúa como vectores, permitiendo localizar al dispositivo según la comparación entre las lecturas actuales del dispositivo y el vector del *Fingerprint*. Si son lo mas parecidos, es decir, su distancia vectorial es mínima, el dispositivo se asocia al lugar correspondiente.

La exactitud que puede ofrecer el *Fingerprint* depende del número de balizas que se coloquen en la zona a monitorizar.

## 2.2. Estrategia del proyecto

Tras un largo proceso de investigación y estudio, se ha concluido plantear el desarrollo de la localización planar a partir del desarrollo de una infraestructura de red, donde se colocan al menos 3 balizas en un mismo plano. Dichas balizas son dispositivos encargados de hacer una lectura de la potencia media de señal recibido del dispositivo a localizar. Dicha medida se realiza en un intervalo determinado, aplicando un filtro de Kalman para suavizar los efectos de las desviaciones rápida originadas y del ruido.

La red esta formada por un módulo central al que estarán conectadas todas las balizas.

Este módulo central se encarga de la recopilación de la potencia de señal proveniente de las balizas, que es guardada en una base de datos; de la formación de tramas, asociando la potencia de señal de cada baliza a un tiempo determinado y a la Id de un usuario, y de la identificación del lugar en donde se encuentra en cada instante el dispositivo a monitorizar a partir de la técnica de *Fingerprints*. También aplica un filtro de Kalman a los valores de lectura de las balizas y un algoritmo de asociación de lugar inteligente, corrigiendo trayectorias que podría resultar imposible de realizar.

Una de las cuestiones importantes a tratar y que debemos de considerar en todo momento es la frecuencia de monitorización de los dispositivos a monitorizar (A lo largo de la memoria nos referiremos a estos como periféricos). Los periféricos se moverán a una velocidad relacionada a la del ser humano y determinada según una serie de factores que influya en el individuo. Por ejemplo, el individuo tenga prisa por alcanzar su sitio de destino; el mismo entorno del individuo, pues es menos probable que empiece a correr en interiores que en exteriores, que sea más propenso a andar, etc. También es necesario considerar el número de periféricos a monitorizar pues nos limitará la capacidad de monitorización.

Otra cuestión a considerar es el sincronismo de las balizas entre ellas para enviar al módulo central los valores de lectura asociado a un mismo periférico y que la conexión entre balizas y módulo central sea lo más fiable posible, que esté disponible siempre.

Y por último, es necesario determinar el grado de precisión que queremos del periférico en cuanto al lugar, como dividiremos el recinto a monitorizar en cuanto a la colocación de los *Fingerprints* y cuantos *Fingerprints* colocaremos. Cuantos más coloquemos, mayor será la precisión de la ubicación del periférico. Sin embargo, a mayor número de *Fingerprints*, mayor será la complejidad de cálculo y diferenciación entre cada uno de ellos. Se requerirán métodos más sofisticados.

## 2.3. Especificaciones y observaciones del Hardware aplicado

Para llevar a cabo el desarrollo del proyecto, es necesario realizar una buena elección del hardware, cuyas prestaciones se nos ajuste lo máximo posible para alcanzar unos resultados razonablemente buenos y que no suponga un alto coste.

### 2.3.1. Periféricos

Como mencionamos anteriormente en las motivaciones del proyecto, el hecho de querer rastrear el flujo de personas hace que nuestro periférico deba ser un dispositivo que se haga personal al usuario. Por lo que hace que la mejor decisión sea que los dispositivos móviles actúe con el perfil de periféricos.

Además, actualmente está extendida la incorporación de tecnología *Bluetooth* en dispositivos móviles, sobre todo la de versiones actuales. Por lo que refuerza más la decisión de escoger el dispositivo móvil como periférico.

Sin embargo, al tratarse de un dispositivo que suele ser personal, es preciso asegurar la seguridad de la identidad de los usuarios, en el caso de una aplicación profesional. Más adelante, en apartados posteriores se esbozará un poco sobre el tema mencionado.

### 2.3.2. Arduino

Para la elección del hardware de las balizas, hemos considerados oportunos el uso de *Arduino*, un placa electrónica de Hardware libre y programable, especializada en la lectura de valores en sensores con un nivel de procesamiento básico. Se caracterizan también por el uso del microcontroladores *ATmega* y de la modularidad basada en *Shield*, que aumentan las características de la misma placa.

El modelo escogido en este proyecto es el *Arduino Nano 33 BLE* ya que ofrece, al igual que las características de una *Arduino Nano*, la funcionalidad *Bluetooth* a un coste mas económico de 17.50 €, que con impuestos quedan en 25 €.

Se consideró la opción de usar la placa *Arduino Uno* o *Nano* junto con un *Shield* especializado en *Bluetooth Low Energy*. Sin embargo, el precio medio de una *Arduino Nano* y *Arduino Uno* ronda sobre los 20€ y sería necesaria para cada baliza y un *shield* promedio de BLE ronda de precio sobre los 15€. Por lo que se hace más caro de este modo.

Cabe mencionar que un aspecto a considerar es la sensibilidad y directividad de la antena Bluetooth. Según las especificaciones del documento, cuenta con una sensibilidad de -95 dBm para BLE de 1 Mbps y de -103 dBm para BLE de 125 Kbps. La potencia de transmisión es configurable entre los -20 dBm a +8 dBm y su antena no llega a ser totalmente omnidireccional.

### 2.3.3. Raspberry Pi

Para la elección del módulo central, hemos considerado el uso de modelos de la *Raspberry Pi*, un nano-ordenador de propósito general de bajo coste, que permite un procesamiento mayor que el hardware de las balizas. Su sistema operativo, *Debian*, esta centrado principalmente en el soporte del lenguaje *Python*, uno de los lenguajes de programación más extendidos y usados actualmente.

Específicamente, para el desarrollo de este proyecto, se ha usado el modelo *Raspberry Pi 4 Modelo B* de 4 Gb. Con un modelo menor, la *Raspberry Pi 3 Modelo B* de 1 GB, también se podría aplicar. Su precio actual ronda sobre los 40 €.

### 2.3.4. Presupuesto del Hardware

Descrito el hardware aplicado en el proyecto, se esboza en la siguiente tabla un presupuesto relacionado al coste del Hardware.

Hardware	Cantidad	Precio Unidad	Precio total
Arduino Nano BLE 33	3	25 €	75 €
Raspberry Pi 3 Modelo B	1	40 €	40 €
		Total	115 €

Cuadro 2.2: Presupuesto del Hardware



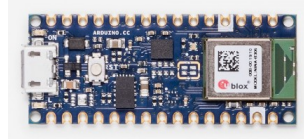


Figura 2.3: Arduino Nano 33 BLE



Figura 2.4: Raspberry Pi 3 Modelo B

## 2.4. Desarrollo del Software del proyecto

A continuación, se detalla el software desarrollado para alcanzar los objetivos de proyecto fijados anteriormente.

### 2.4.1. Monitorización de las balizas

Para la monitorización de los periféricos a partir de las balizas, se ha considerado, dentro de la programación de las balizas, una vez que estas balizas están conectadas con el modulo central, exista un periodo de búsqueda, donde la baliza escanea. Mediante la clase BLE Device (función de la librería *ArduinoBLE.h*, cuya propietaria es Arduino) y la función *BLE.available()*, se obtiene, en el caso de haber encontrado un periférico, información acerca del dispositivo. Dicha información principalmente es la dirección MAC de *Bluetooth* y el nombre local del dispositivo.

Si el dispositivo coincide con el nombre local de uno de los dispositivos que esta autorizados, la baliza se conectará con el periférico y mediante el la función *rssi()* obtendremos una estimación de la potencia de transmisión. Para ofrecer una mayor precisión, evitando los efectos del desvanecimiento rápido y del ruido, se procede a obtener en un determinado tiempo una cantidad considerable de valores (entre 15 y 20 valores), se aplica a dichos valores un filtro de Kalman y posteriormente se pone como característica de la transmisión a leer la media de dichos valores.

Una cuestión a tratar es la frecuencia de la monitorización, se ha supuesto que la persona puede llegar a una velocidad máxima de 6 Km/h, siendo de 1,67 m/s. En cuanto al criterio de precisión, se ha optado por considerar el periodo de monitorización como el tiempo que tarda el individuo en recorrer un metro, siendo esta la inversa de la velocidad fijada en metro partido segundo.

El periodo obtenido es de 600 milisegundos. Por lo que, se opta por garantizar que el tiempo entre muestra y muestra correspondiente a un periférico sea de 600 milisegundos. Cabe mencionar que en este proyecto se ha decidido implementar la conexión simultanea entre 5 periféricos, por lo que cada periférico debe garantizar un tiempo de computo

que sea menor que la quinta parte del tiempo de período de monitorización, es decir, en nuestro caso de 120 milisegundos.

Es necesario considerar que, para que la baliza pueda monitorizar a los periféricos, el dispositivo móvil tiene que activar un servicio determinado de *Bluetooth* donde muestre el nombre local y ofrezca leer la potencia de transmisión. No podemos monitorizar a partir de la dirección MAC de *Bluetooth* del dispositivo móvil ya que tanto para Android como Apple, cuentan con mecanismos de seguridad donde la dirección MAC del *Bluetooth* va cambiando, siendo la que muestra una dirección MAC virtual. Cabe mencionar que, para obtener el mayor rango posible, es de gran interés poder modificar la ganancia de transmisión del móvil.

Para este proyecto, se ha usado la aplicación Android *nRF Connect* para la personalización de servicios *Bluetooth*, simplificando más el proceso de la realización del proyecto.

### 2.4.2. Flujo de datos entre las balizas con módulo central

Para el flujo de datos entre las balizas y el módulo central, se ha abordado a partir de la conexión simultánea entre baliza y módulo central a partir. Dicha conexión simultánea es llevada a cabo, en la programación de la *Raspberry Pi*, mediante el uso de un gestor de subprocesos o hebras.

El módulo central actúa como cliente de la baliza, donde el módulo principal pedirá constantemente las características del servicio *Bluetooth*. Cada característica está identificada mediante el UUID (Identificador único universal, que se trata de un identificador de 128 bits o una serie de string) y predefinida anteriormente.

Se establece 4 características a enviar que son las siguientes:

- **RSSI.** Potencia media recibida del periférico.
- **Marca de sincronismo.** Número de contador de la baliza que identifica la trama a partir de un número entero. De esta manera, el módulo central puede identificar las tramas de lectura que obtiene, de forma que, si obtiene la misma marca de sincronismo que la obtenida anteriormente, la ignora pues se trata de la misma.
- **Tiempo de computo.** Esta característica nos permite saber cuanto ha tardado la baliza en medir el valor de la RSSI obtenido. Su propósito principal es poder comprobar con facilidad si la baliza ha comenzado a tener fallos de computo.
- **Id del periférico.**

### 2.4.3. Funciones del módulo principal

Como ya se ha comentado con anterioridad, el módulo principal de nuestro sistema de localización será la *Raspberry Pi*. Este módulo se encargará de la gran parte de procesamiento de datos del sistema.

La función básica de la *Raspberry Pi* es la de recoger la información que aportan las diferentes balizas sobre los diferentes periféricos, de la manera en la que se ha expuesto en el apartado anterior. Todo este flujo de datos se recoge en una gran base de datos conformada por las distintas tramas que recibe la *Raspberry Pi*. Dicha base de datos servirá como punto de partida para la conformación de una base de datos organizada, que

será la principal, de una manera más clara y orientada a un procesamiento más extenso de los datos recogidos. Esta base de datos primera será borrada cada 15 minutos, en nuestro caso, para el ahorro de memoria del módulo principal.

Esta base de datos principal recoge las tramas de seguimiento de cada periférico. Dichas tramas de seguimientos estarán formadas por los valores de RSSI de cada baliza en el mismo instante de tiempo, el tiempo de computo de cada baliza en obtener dicho valor, la marca de tiempo en el que el módulo principal lee los valores de la baliza, el lugar asociado, el id del periférico y la marca de tiempo de la lectura de la ultima baliza, de las 3 asociadas.

Es complejo que se tenga el valor de potencia de las tres balizas para un mismo instante de tiempo. Por ello, se agrupan valores de potencia de manera que no difieran en más de un segundo por si el periférico está en movimiento.

Una vez generadas todas las bases de datos necesarias para el correcto procesamiento de los datos, la *Raspberry Pi* tiene la función de conseguir la localización del periférico ayudándose de los datos logrados con *Fingerprints*. Todo este proceso se explicará con más detalle en el siguiente apartado. Para finalizar, la *Raspberry Pi* dibujará el plano del lugar en el que se están tomando las medidas y la trayectoria del periférico a rastrear.

# Capítulo 3

## Experimentación, resultados y conclusiones

En este apartado se dará una explicación de los pasos realizados durante la experimentación, los resultados obtenidos y las diferentes conclusiones a las que se han llegado tras numerosas pruebas.

### 3.1. Lugar de pruebas

El sistema de localización planar se ha puesto a prueba en un piso que tiene la siguiente disposición:

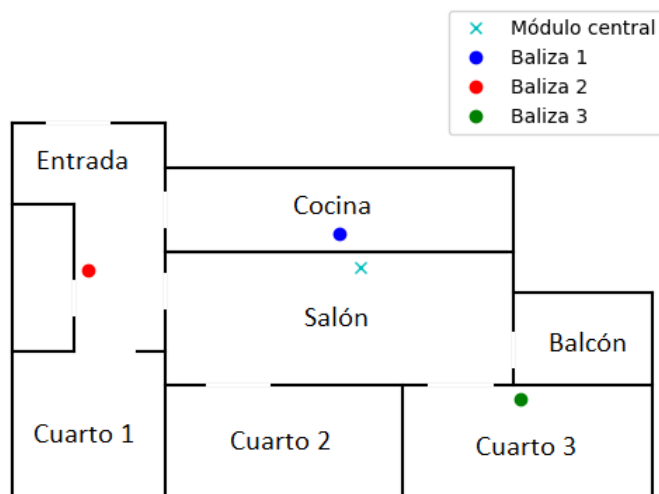


Figura 3.1: Plano del recinto utilizado

En la figura 3.1 se puede observar la localización de las balizas utilizadas y del módulo principal. Las balizas están colocadas de manera que todos los periféricos tengan la mayor parte del tiempo conexión con todas las balizas y se cubra la mayor parte del recinto. Por otro lado, el módulo principal ha sido colocado en una zona central para que tenga conexión en todo momento con las balizas. Los periféricos si pueden desconectarse de las

balizas pero una desconexión entre baliza y módulo principal es muy perjudicial para el sistema por la pérdida de tramas de información.

### 3.2. Fingerprint

Antes de poder calcular la posición de cada periférico se necesita realizar un estudio previo del recinto para conocer las medidas de potencia en cada punto. Este estudio se conoce como *Fingerprint*. Se dividirá cada sala del recinto en otras subsalas para poder dar una precisión mayor. En cada subsala se tomarán medidas durante un tiempo lo suficientemente extenso para que las medidas puedan ser fiables. Una vez tenemos estas medidas, se les aplica un filtro de Kalman para poder corregir posibles desviaciones, y para obtener solo un valor de potencia se aplica una media de todos los valores.

### 3.3. Procesamiento de tramas

Una vez se obtienen los datos de los *Fingerprints* se puede pasar a la etapa de procesar la base de datos resultante de las tramas recibidas por las balizas. Por lo general, para cada instante de tiempo están disponibles los valores de las tres balizas. Sin embargo, puede ocurrir que algún periférico se desconecte y llegue con un RSSI nulo. Este hecho puede introducir errores no deseados en el cálculo de las posiciones, por lo que se debe adoptar alguna solución al respecto. En este caso, se ha optado por realizar una interpolación lineal. Es decir, si tuviéramos muchas tramas seguidas con un valor nulo de potencia, estos nulos se sustituirían por una interpolación entre la última trama que aportó potencia no nula hasta la siguiente trama de potencia no nula. Por ejemplo, si tuviéramos la entrada [-37 -36 0 0 -39], los valores de potencia utilizados para el cálculo de posición serían [-37 -36 -37 -38 -39].

Antes de calcular la posición asociada a cada trama, dependiendo de los valores y el recinto en el que esté, se puede aplicar un filtro de Kalman a las muestras. Esta parte es opcional, ya que en el recinto utilizado para este caso no aporta una gran ventaja.

### 3.4. Cálculo de la localización

Para cada trama captada por el módulo central se calcula su posición asociada. El método utilizado en este sistema es el de vecinos más cercanos. Es decir, se calculará la distancia euclídea entre los valores de potencia de la trama respecto a los valores de potencia de las subsalas, conseguidas con *Fingerprint*. Por lo tanto, la subsala que otorgue una distancia euclídea menor será la asignada. Sin embargo, los datos de potencia pueden fluctuar mucho entre tramas, por lo que se ha optado por hacer uso de ponderaciones. La ecuación a utilizar será la siguiente:

$$Distancia = P * \sqrt{(T_{x1} - F_{p1})^2 + (T_{x2} - F_{p2})^2 + (T_{x3} - F_{p3})^2}$$

Donde  $P$  es el valor de ponderación asociada a la subasala anterior,  $T_x$  el valor de potencia de la baliza y  $F_p$  el valor de potencia del *Fingerprint* asociada a esa subsala.

El valor de ponderación se elige teniendo en cuenta la subsala asociada a la trama anterior. Dependiendo de la subsala anterior y la subsala con la que estamos calculando la distancia el valor de ponderación será distinto. Para subsalas con gran distancia entre ellas o entre las que no tengan acceso directo tendrán un valor de ponderación mayor que subsalas contiguas. En este caso, los valores de ponderación se han asignado dependiendo del número de subsalas entre ellos. El valor óptimo asociado a cada ponderación se ha obtenido experimentalmente tras múltiples ensayos. Para cada recinto se obtendrán valores de ponderación distintos.

Una vez tenemos todas las tramas asociadas a una localización el módulo central dibujará la trayectoria del periférico que se esté analizando. Un ejemplo sería el siguiente:

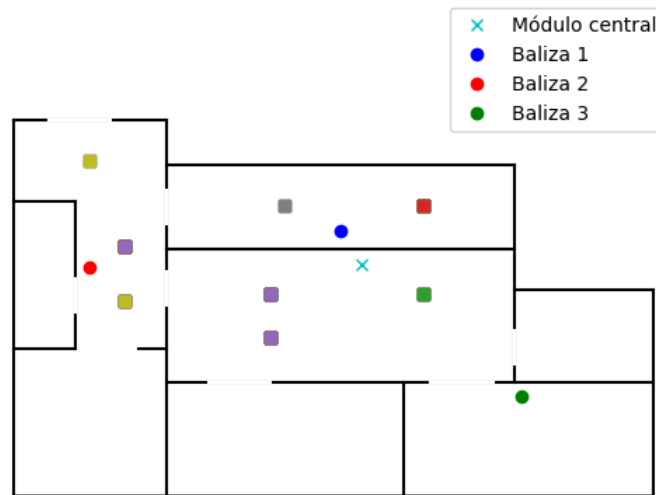


Figura 3.2: Ejemplo de una trayectoria

Se dibuja la trayectoria de un periférico que se mueve desde la parte superior derecha del salón hasta la parte derecha de la cocina. Se puede observar como, en líneas generales la trayectoria es la correcta. Se obtienen algunas desviaciones esporádicas como justo antes de salir del salón que la localización no es del todo exacta ya que baja y después justo antes de entrar a la cocina se desvía un poco hacia la entrada. Sin embargo, los resultados obtenidos son muy satisfactorios.

### 3.5. Conclusiones

Una vez finalizado el proyecto, las conclusiones obtenidas son muy positivas. El objetivo de localización planar se ha logrado con cierta exactitud. Cabe decir que si en vez de haber buscado una precisión tan alta, dividiendo las salas en subsalas, los resultados hubiesen sido mucho mejores. Sin embargo, no se ha podido realizar el sistema para más de un periférico simultáneamente ya que no se ha conseguido que estuviese conectado a las tres balizas en el mismo instante de tiempo. Es por esto, que nuestro objetivo de conseguir monitorizar los aforos de las salas no tendría sentido al disponer de un único periférico. Por ello, nos declinamos finalmente a un sistema más enfocado a una localización con mayor exactitud.

# Capítulo 4

## Perspectiva al futuro

En este apartado se propondrán algunos aspectos en los que mejorar nuestro sistema para futuras continuaciones del mismo.

Una de las posibles mejoras sería la optimización del código de Arduino para que los periféricos se puedan conectar a todas las balizas. El código ya programado está muy cerca de este objetivo pero faltaría tiempo para poder indagar un poco más en este problema. Si se pudieran conectar más de un periférico se podría abordar el tema del control de aforos por sala.

Por otro lado, en nuestro sistema los valores de RSSI nulos se consideran como desconexiones, no tenemos en cuenta el caso de que un periférico esté fuera del alcance de una baliza. Por lo tanto, otro tema a tratar sería el hecho poder conseguir un mayor rango de operabilidad teniendo en cuenta esta problemática, ya que podrían moverse los periféricos con mayor libertad.

Un aspecto importante sería el desarrollo de una aplicación móvil que optimice también los recursos de *Bluetooth* del periférico. En este proyecto se ha desarrollado a partir de móviles Android, pues la aplicación para móviles iPhone empiezan a fallar o no ofrece tanta personalización.

Por último, a la hora obtener información sobre el estudio a tratar se ha podido observar como una métrica alternativa a la hora de calcular distancias es la de utilizar redes neuronales. Este método sería mucho más complejo y daría para un trabajo nuevo entero.

# Bibliografía

- [1] **Bluetooth Radio Versions:** <https://www.bluetooth.com/learn-about-bluetooth/radio-versions/>
- [2] **Estimador de rango Bluetooth:** <https://www.bluetooth.com/learn-about-bluetooth/key-attributes/range/>
- [3] **Librería Arduino BLE** <https://www.arduino.cc/en/Reference/ArduinoBLE>
- [4] **Especificaciones de Arduino Nano BLE 33** [https://content.arduino.cc/assets/Nano\\_BLE\\_MCU-nRF52840\\_PS\\_v1.1.pdf](https://content.arduino.cc/assets/Nano_BLE_MCU-nRF52840_PS_v1.1.pdf)
- [5] **Sitio web de compra de la Arduino Nano 33 BLE** <https://store.arduino.cc/arduino-nano-33-ble>
- [6] **Información oficial acerca de la Raspberry Pi 3** <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>