



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

SISTEMA DE POSICIONAMIENTO EN INTERIORES CON TECNOLOGÍA UWB

Alumno: Lourdes Mercado Calero.

Tutor: Dr. D. Javier Medina Quero.
Dpto: Informática.

Julio, 2019

Alumno: Lourdes Mercado Calero.

Sistema de posicionamiento en interiores con tecnología UWB

Lourdes Mercado Calero

Julio, 2019



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don Javier Medina Quero , tutor del Proyecto Fin de Carrera titulado: Sistema de posicionamiento en interiores con tecnología UWB, que presenta Lourdes Mercado Calero, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Julio de 2019

El alumno:

Lourdes Mercado Calero

Los tutores:

Javier Medina Quero

Me gustaría expresar mi agradecimiento a todas aquellas personas que me han apoyado y que han aportado su experiencia ayudándome a llegar hasta aquí.

En primer lugar, me gustaría expresar mi agradecimiento a mi tutor, Javier Medina Quero, y a mis profesores los cuales han contribuido a mi formación y aportado su experiencia a lo largo de este periodo universitario.

En segundo lugar, me gustaría agradecer a mis compañeros de clase todos aquellos momentos compartidos y la gran ayuda aportada durante todo este tiempo.

Índice general.

1. Introducción.....	14
1.1 Motivación.	14
1.2 Propuesta.	16
1.3 Objetivos.	17
1.4 Descripción de la arquitectura.	17
1.5 Planificación temporal.	18
1.5.1 Estimación de tiempos.	18
1.4.2 Diagrama de Gantt.....	19
1.5 Estructura de la memoria.	21
2. Sistemas de posicionamiento en interiores.	23
2.1 Internet de las cosas.	24
2.2 Tecnologías para IPS.	27
2.3 Aplicaciones.	30
2.4 Descripción de la propuesta realizada.....	32
2.5 Protocolo de mensajería publicación/suscripción MQTT.....	32
2.5.1 Ventajas.....	34
2.5.2 Inconvenientes.....	35
2.5.3 Conclusión.	35
2.6 Laboratorio utilizado.	35
2.7 Tecnología UWB empleada.....	38
2.7.1 Suscripción a la localización del usuario en tiempo real desde un dispositivo móvil.	41
3. Proceso de ingeniería del software.	42
3.1 Fases de la ingeniería del software.	42
3.2 Análisis de requerimientos.	43

3.2.1	Requerimientos funcionales.....	43
3.2.2	Requerimientos no funcionales.....	44
3.2.2.1	Requerimientos físicos.	44
3.2.2.2	Requerimientos software.....	45
3.2.2.3	Requerimientos de la interfaz.....	45
3.3	Casos de uso	46
3.4	Diseño y arquitectura.....	52
3.4.1	Diseño de clases.....	52
3.4.2	Arquitectura.....	55
3.5	Implementación	56
3.6	Pruebas.	58
4.	Conclusiones y líneas de trabajos futuros.	71
5.	Anexo.	73
5.1	Manual de usuario.	73
5.2.1	Configuración de los dispositivos hardware UWB.	73
5.2.2	Recolección y estudio de los datos.....	76
5.2.3	Monitorización del recorrido.....	81
6	Bibliografía.	83

Índice de imágenes.

Figura 1.1: Esquema básico de posicionamiento en interiores.....	15
Figura 1.2: Comparativas de transmisión de radio.....	16
Figura 1.3: Estructura de trabajo.....	18
Figura 1.4: Diagrama de Gantt.....	20
Figura 2.1: Esquema básico IoT.....	24
Figura 2.2: Ejemplo posicionamiento BLE.....	29
Figura 2.3: Topología de estrella MQTT.....	33
Figura 2.4: Ejemplo de funcionamiento MQTT.....	34
Figura 2.5: Mapa Laboratorio CEATIC.....	36
Figura 2.6: SmartLab.....	37
Figura 2.7: Ejemplo real de una de las anclas utilizadas.....	38
Figura 2.8: Ventana principal DecaRange.....	40
Figura 2.9: Log obtenido con DecaRange.....	40
Figura 2.10: Monitorización del usuario.....	41
Figura 3.1: Diagrama de Caso de uso 1.....	47
Figura 3.2: Caso de uso 2.....	48
Figura 3.3: Caso de uso LogSender.....	50
Figura 3.4: Caso de uso LocationServer.....	51
Figura 3.5: Diagrama de clases.....	54
Figura 3.6: Esquema de la arquitectura del sistema.....	55
Figura 3.7: Ejemplo de un usuario realizando el recorrido con la tag encima...58	
Figura 3.8: Mapa de la distribución de las marcas.....	59
Figura 3.9: Gráfica primer recorrido.....	62
Figura 3.10: Gráfica segundo recorrido.....	63
Figura 3.11: Datos obtenidos con LogProcessor.....	66
Figura 3.11: Recorrido del usuario.....	67
Figura 3.12: App móvil.....	68
Figura 3.13: Coordenadas obtenidas con MapLocation.....	69
Figura 5.1: Antena UWB.....	74
Figura 5.2: Función de cada pin.....	74
Figura 5.3: Configuración DIP para ancla fija.....	75

Figura 5.4: Configuración del ancla del sistema	75
Figura 5.5: Configuración DIP para ancla móvil (tag).....	75
Figura 5.6: Configuración de la tag del sistema.....	76
Figura 5.7: Esquema de conexiones.....	77
Figura 5.8: Interfaz de usuario DecaRange.....	78
Figura 5.9: Panel de ajuste.....	78
Figura 5.10: Interfaz aplicación móvil.....	80
Figura 5.11: Ejemplo de monitorización del usuario.....	82

Índice de tablas.

Tabla 1.1: Estimación de tiempos.....	19
Tabla 3.1: Caso de uso 1.....	47
Tabla 3.2: Caso de uso 2.....	49
Tabla 3.3: Caso de uso LogSender.....	50
Tabla 3.4: Caso de uso LocationServer.....	52
Tabla 3.5: Símbolos del diagrama de clases.....	55
Tabla 3.6: Primer recorrido del usuario.....	65
Tabla 3.7: Segundo recorrido del usuario.....	66

1. Introducción

1.1 Motivación.

Actualmente los sistemas de posicionamiento o localización están muy extendidos y la sociedad se ha familiarizado bastante con ellos debido a que lo utilizan muy a menudo en su día a día, un claro ejemplo es el GPS o sistema de posicionamiento global del que disponen los smartphones. Esta tecnología nos ofrece una gran variedad de posibilidades, desde poder trazar la ruta más corta para realizar un viaje hasta poder obtener la localización de una persona en caso de emergencia.

La tecnología GPS se centra principalmente en posicionamiento en exteriores ya que, no dispone de una buena precisión para ser utilizado en interiores. Por ello, es necesario disponer de otro tipo de tecnología que nos permita obtener el posicionamiento en el interior con la mayor precisión posible.

Esta tecnología de posicionamiento en interiores se basa en la utilización de unos dispositivos fijos cuya posición es conocida que permiten obtener con una mayor precisión la localización de objetos o personas en el interior, en lugar de utilizar una red de satélites como hace la tecnología GPS.

En la Figura 1.1 se puede ver de forma general como se distribuye en el espacio los distintos componentes utilizados en los sistemas de posicionamiento en interiores. Entre estos componentes tenemos anchor (anclas), este término hace referencia en este trabajo fin de grado a un dispositivo del kit TREK1000 cuyo funcionamiento consiste en determinar la posición de la tag desde una posición estática. Por otro lado, tenemos la tag (etiqueta) este término hace referencia a otro dispositivo del kit TREK1000 a diferencia del anchor la tag es un dispositivo móvil que el usuario llevará encima para permitir su localización en todo momento. Ambos dispositivos hacen uso de la tecnología UWB para establecer la comunicación que permitirá determinar la localización del usuario en todo momento.

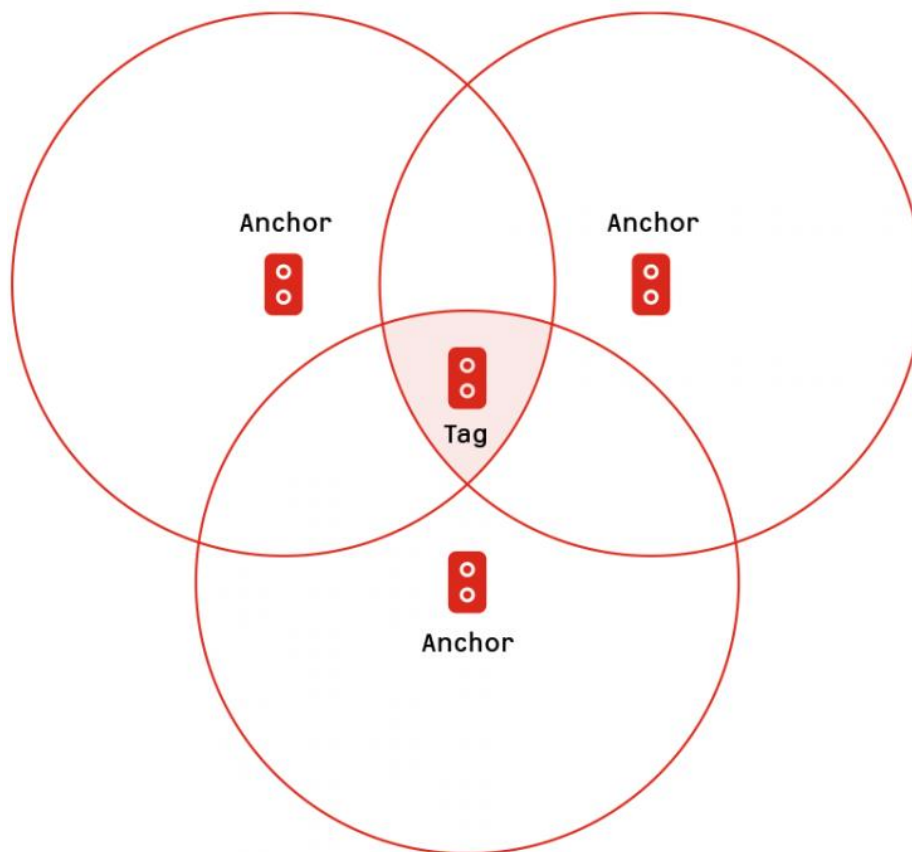


Figura 1.1: esquema básico de posicionamiento en interiores

Además, para calibrar los sistemas de posicionamiento en interiores (IPS) es importante disponer de una herramienta para etiquetar la posición real del usuario en la fase de pruebas y compararla con la estimación del sistema ya que, esta tecnología es sensible a obstáculos que distorsionan la señal provocando que puedan surgir errores en los resultados.

En la actualidad existen muchas tecnologías para los sistemas de posicionamiento en interiores, siendo las más conocidas las tecnologías Wifi [1] y Bluetooth [2]. Aunque, teniendo en cuenta que en estos sistemas es fundamental la precisión hay que destacar la tecnología UWB [3] (ultra wide-band).

- **Descripción de la tecnología UWB.**

UWB [4] es una tecnología de bandas de radio (RF) capaz de ofrecer transmisión de datos de alta velocidad (hasta 400-500 Mbit/s) y requiere un bajo consumo para su funcionamiento.

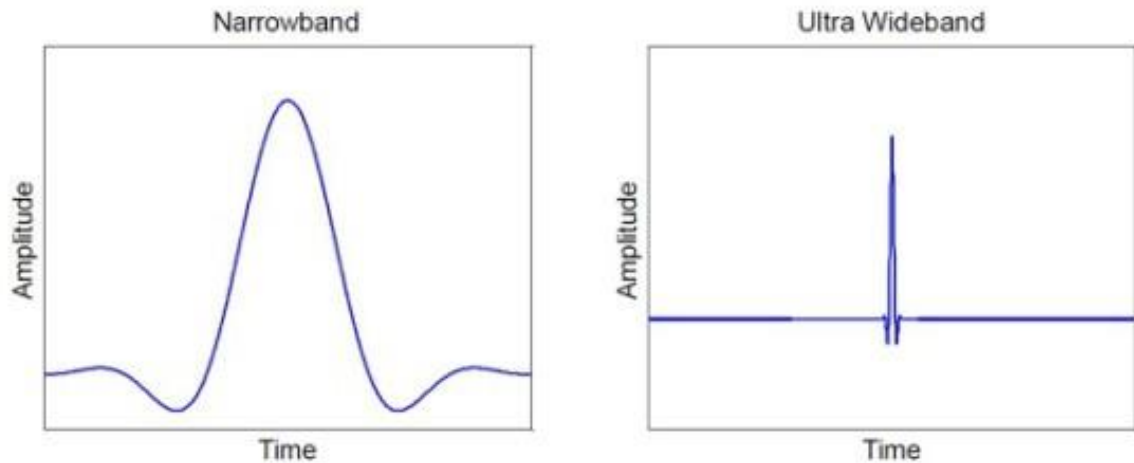


Figura 1.2: Comparativas de transmisiones de radio.

La Figura 1.2 muestra la diferencia entre la transmisión de radio típica y la transmisión de radio producida por la tecnología UWB. La transmisión realizada por UWB permite una mayor precisión incluso en entornos con ruido y causa menos interferencias con los elementos del entorno en el que se está haciendo uso de esta.

1.2 Propuesta.

La propuesta de este trabajo de fin de grado es desarrollar un prototipo de aplicación para posicionamiento en interiores con tecnología UWB en un ámbito específico, junto a la elaboración de los manuales necesarios para el correcto funcionamiento de este prototipo.

1.3 Objetivos.

Los objetivos previstos a lograr para este trabajo fin de grado son los siguientes:

- Llevar a cabo una revisión detallada de la bibliografía de la tecnología UWB.
- Definir el ámbito de trabajo en el que se va a desarrollar la aplicación con la tecnología UWB.
- Realizar el análisis y diseño de la aplicación con tecnología UWB para posicionamiento en interiores.
- Desarrollar un prototipo de aplicación con tecnología UWB.
- Elaboración del manual de usuario asociado al prototipo de aplicación con tecnología UWB.
- Redactar una memoria en la que se recoja todo el trabajo realizado incluyendo en la misma el manual de usuario del prototipo de aplicación desarrollada.

1.4 Descripción de la arquitectura.

En esta sección se va a realizar una breve descripción de la arquitectura del sistema de posicionamiento en interiores. Como ya se mencionará más adelante, la arquitectura de este sistema es una arquitectura cliente-servidor en la cual intervienen principalmente los dispositivos con tecnología UWB. Además, también formará parte de esta arquitectura el protocolo de envío de mensajes MQTT, el cual establece una conexión haciendo uso del protocolo TCP/IP. MQTT basa su funcionamiento en un broker encargado de establecer las diferentes conexiones entre los clientes y en topics a través de los cuales el broker enviará mensajes a los clientes que se suscriban a un determinado topic. En la sección 2.5 de este trabajo fin de grado se realizará una descripción más detallada del funcionamiento de este protocolo y se mencionarán las ventajas e inconvenientes de su uso.

1.5 Planificación temporal.

La Figura 1.3 se corresponde con la estructura jerarquizada de las diversas tareas realizadas para llevar a cabo la elaboración de este trabajo fin de grado.

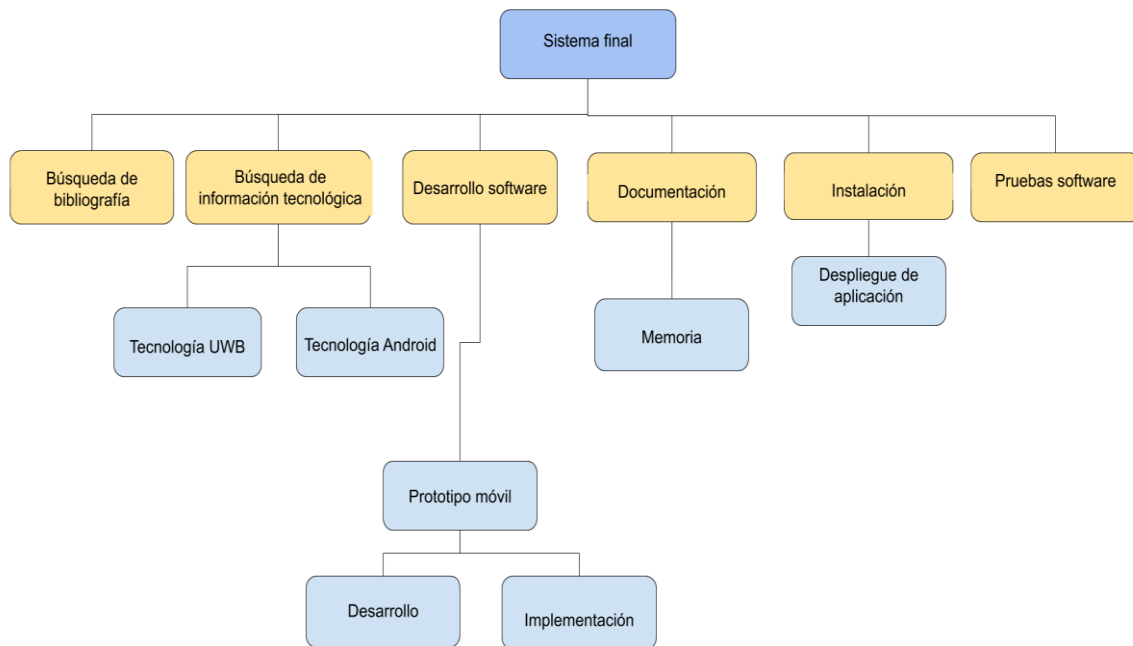


Figura 1.3: Estructura de trabajo

1.5.1 Estimación de tiempos.

La estimación de tiempos realizada se ha llevado a cabo desde un punto de vista optimista del periodo de tiempo necesario para completar las diversas tareas que han permitido completar este trabajo fin de grado. En la tabla 1.1 que se muestra a continuación aparece un desglose del periodo de tiempo dedicado para cada una de las tareas en días, obteniendo un total de 127 días para la realización de este trabajo fin de grado.

Tarea	Tiempo (días)
Búsqueda de bibliografía	19
Búsqueda de información tecnológica	
Tecnología UWB	18
Tecnología Android	9
Desarrollo software	
Prototipo móvil	
Desarrollo	11
Implementación	14
Documentación	
Memoria	30
Instalación	
Despliegue de aplicación	7
Pruebas software	19
Total	127

Tabla 1.1: Estimación de tiempos.

1.4.2 Diagrama de Gantt.

El diagrama de Gantt es una herramienta gráfica, la cual nos va a permitir exponer de forma más visual el tiempo y dedicación que se ha previsto para las diferentes tareas en las que se ha dividido el proyecto. Además, este diagrama, permite mostrar la duración total para cada una de las diferentes tareas y las fechas establecidas para cada una de estas tal y como se muestra en la Figura 1.4.

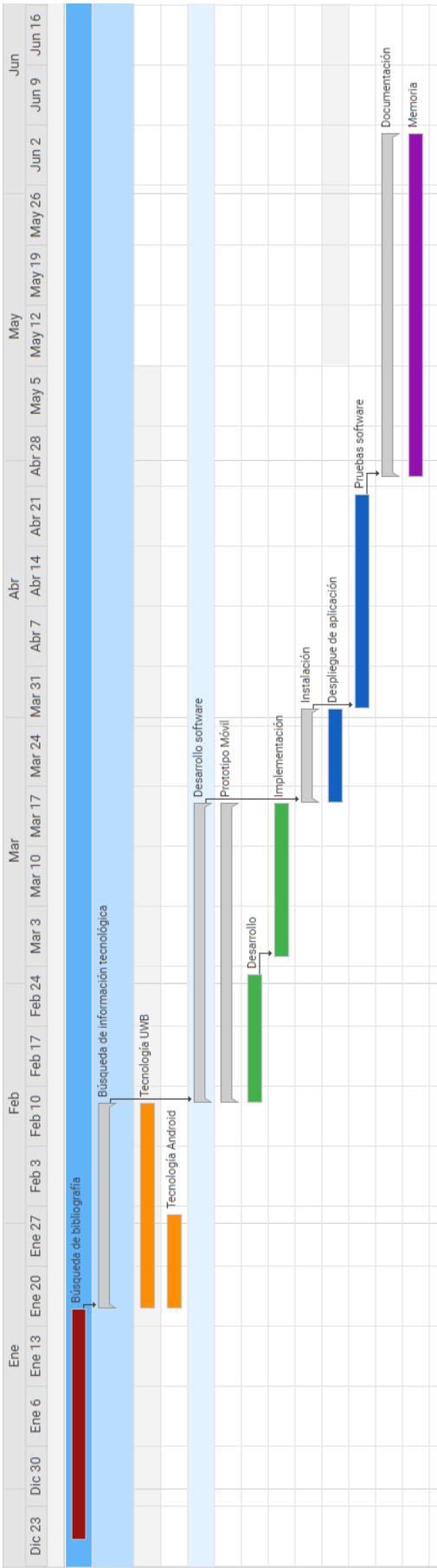


Figura 1.4: Diagrama de Gantt.

1.5 Estructura de la memoria.

A continuación, se va a proceder a realizar la descripción del contenido de este trabajo fin de grado.

- **Capítulo 1.** Durante este primer capítulo se ha procedido a explicar la motivación que ha llevado a la realización de este trabajo fin de grado, junto a los objetivos que se pretenden alcanzar en el mismo. Por último, se ha incluido la planificación temporal de las diversas tareas que han sido necesarias realizar para concluir este trabajo fin de grado.
- **Capítulo 2.** En este capítulo se comenzará explicando que es un sistema de posicionamiento en interiores, enumerando y explicando las diversas tecnologías que existen para estos sistemas e indicando sobre cuál de ellos nos centraremos en este trabajo fin de grado. A continuación, explicaré con ejemplos las diversas aplicaciones de esta tecnología. Posteriormente se pasará a describir el lugar utilizado para poner en marcha el sistema de posicionamiento en interiores con tecnología UWB, el laboratorio del Centro de Estudios Avanzados en Tecnologías de la Información y Comunicación (CEATIC) ya que, este ha sido el laboratorio que se ha utilizado para trabajar con la tecnología UWB durante la realización de este trabajo fin de grado. Para finalizar este capítulo, se va a proceder a realizar una explicación en detalle de la tecnología UWB y del estudio realizado para obtener y procesar los datos obtenidos por las mediciones realizadas por el sistema empleado.
- **Capítulo 3.** Durante la realización de este tercer capítulo va a proceder a describir los procesos de la ingeniería del software que se han llevado a cabo para la realización de este trabajo fin de grado, análisis de requerimientos, diseño y arquitectura, programación y pruebas. Inicialmente se comienza describiendo las fases de la ingeniería del software para posteriormente ir indagando en cada una de ellas, comentando el caso específico del software desarrollado para este caso.

- **Capítulo 4.** En este cuarto capítulo se comentan las conclusiones a las que se han llegado durante la realización de todo el proceso que ha conllevado este trabajo fin de grado, y además también se comentarán las posibles líneas de trabajo futuras.
- **Anexo:** En esta parte se incluirá el manual de usuario correspondiente para el correcto uso del sistema de posicionamiento en interiores con tecnología UWB por parte del usuario.
- **Bibliografía:** Una vez finalizada la estructura principal seguida por este trabajo fin de grado se procederá a incluir en esta sección toda la documentación que ha sido consultada para la realización de este trabajo.

2. Sistemas de posicionamiento en interiores.

El posicionamiento es un proceso el cual consiste en determinar la posición de una persona u objeto en un entorno determinado. Un sistema de posicionamiento en interiores o también conocido como Indoor Positioning System [5] (IPS), consiste en un conjunto de dispositivos situados en dicho entorno, que permiten realizar continuamente la localización de un objeto o persona en un entorno determinado en tiempo real. En este trabajo fin de grado nos centraremos en dispositivos de recepción inalámbrica, denominados anclajes, y dispositivos de emisión que transmiten una señal a los anclajes, permitiendo conocer su posición y que permiten localizar personas u objetos en el interior de los edificios.

Con el avance tecnológico la posibilidad de calcular la posición de una persona u objeto con precisión ha supuesto un gran paso para estos sistemas. Sin embargo, estos sistemas aún tienen algunas dificultades para realizar un cálculo de posicionamiento preciso ya que, los edificios cuentan con paredes, objetos y otras peculiaridades que pueden llegar afectar negativamente los resultados obtenidos al realizar el cálculo del posicionamiento en interiores.

Tal y como se va a mencionar a continuación, una de las tecnologías para el IPS es UWB, esta tecnología ha sido utilizada para llevar a cabo el sistema de posicionamiento en interiores de este trabajo fin de grado. Dicha tecnología hace uso del estándar de red para la comunicación conocido como Personal Area Network (PAN) [6] para establecer la comunicación entre los diferentes dispositivos del sistema de posicionamiento en interiores basado en la tecnología UWB.

Esta tecnología cuenta con numerosas ventajas [7] entre las cuales se encuentran las siguientes:

- Dispone de una alta precisión en el posicionamiento, incluso con la presencia de diversos obstáculos.

- No causa interferencias con los sistemas de radio frecuencia existentes, siempre y cuando el sistema basado en UWB haya sido correctamente diseñado.
- Dispone de un gran potencial para futuros desarrollos ya que, ofrece un bajo consumo y una alta productividad

2.1 Internet de las cosas.

Internet de las cosas (IoT – Internet Of Things) [8] hace referencia a un conjunto de dispositivos que se interrelacionan entre sí sin requerir interacción externa y que disponen de la capacidad de transferir datos a través de una red. Estos dispositivos pueden ser un automóvil con sensores incorporados, una persona o un animal con un chip, etc. Los diferentes dispositivos conectados a la red disponen de una dirección IP asociada a los mismos. En la Figura 2.1 se muestra un esquema básico de las conexiones que se establecen en el IoT.

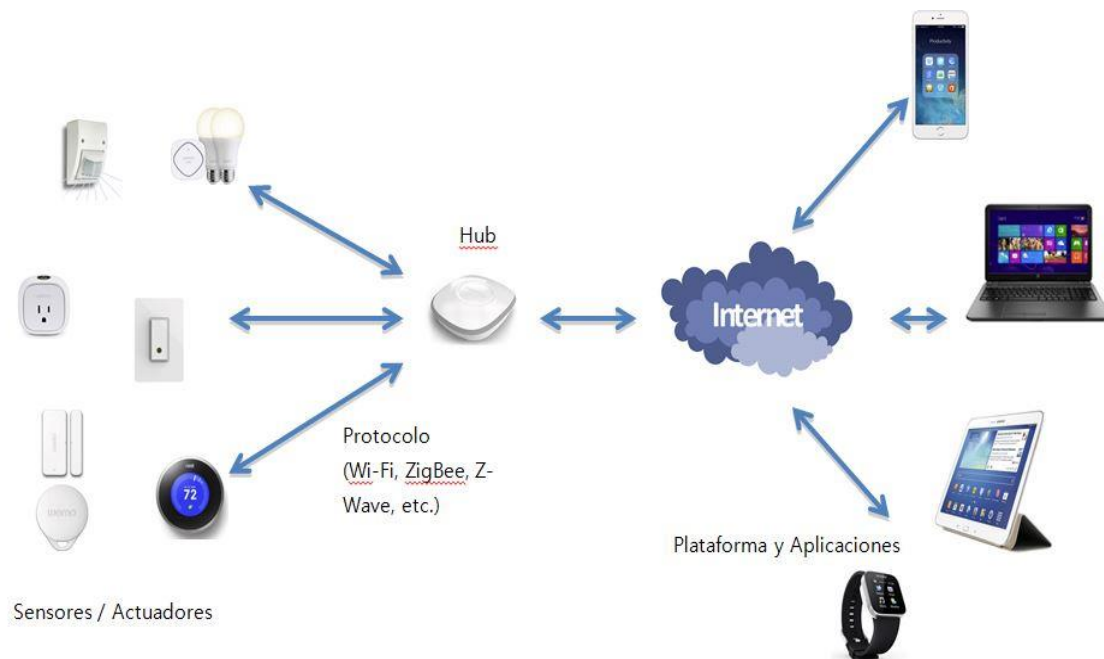


Figura 2.1: Esquema básico IoT.

Este concepto surge debido a la evolución de internet que permite dotar de conexión a la red a una gran cantidad de dispositivos abriendo una gran variedad de posibilidades para el IoT. Esto hace que tenga una gran cantidad de aplicaciones de entre las cuales estas son algunas de ellas [9]:

- **Medicina y salud:** Los dispositivos en este caso pueden variar desde monitores que permitan controlar las pulsaciones o la presión sanguínea hasta dispositivos que permitan realizar un seguimiento algunos implantes realizados a pacientes como pueden ser un marcapasos o un audífono.
- **Transporte:** Muy útil para monitorizar el tráfico en grandes ciudades. Un claro ejemplo de esta aplicación es Waze que proporciona a los conductores información en tiempo real o Google Maps.
- **Gestión de almacenes de logística:** Es imprescindible poder gestionar correctamente los almacenes de logística para que un consumidor final pueda recibir sus productos. Samrt Warehousing es una empresa dedicada a desarrollar software para trabajar en la gestión de almacenes de logística haciendo uso de IoT.
- **Domótica:** Para esta aplicación están surgiendo diferentes productos que permiten saber si, por ejemplo, una puerta está cerrada o abierta. También existen otros productos que permiten analizar piscinas para controlar la calidad del agua, un ejemplo de esto blue desarrollado por Riio.

Por otro lado, IoT dispone de una gran variedad de protocolos de comunicación que permiten cumplir requisitos esenciales para los diferentes dispositivos utilizados, como son: la escalabilidad, mantener el acoplamiento entre los dispositivos, la interoperabilidad, gran cantidad de comunicaciones simultaneas, seguridad y dispositivos fácilmente accesibles. Estos son algunos de los protocolos de comunicación M2M (machine to machine) disponibles para IoT [10] [11]:

- **MQTT (Message Queuing Telemetry Transport):** es un protocolo de mensajería basado en el estándar de publicación-suscripción que funciona sobre el protocolo TCP/IP. Hace uso de un broker para establecer las conexiones entre los diferentes dispositivos. Este protocolo resulta adecuado para dispositivos de baja potencia. Este protocolo no dispone de tolerancia a fallos debido a que la comunicación se realiza a través de un punto único (broker), que es el nodo central y si se produce un fallo en la comunicación con este nodo será imposible establecer una conexión entre los diferentes clientes.
- **AMQP (Advanced Messages Queuing Protocol):** es un protocolo basado en el estándar publicación-suscripción que asegura la confidencialidad y la interoperabilidad. Este protocolo no es recomendado para dispositivos con pocos recursos ya que requiere de un mayor rendimiento.
- **XMPP (Extensible Messaging and Presence Protocol):** es un protocolo abierto basado en XML inicialmente creado para mensajería instantánea. Este protocolo se caracteriza por su sencillez y adaptabilidad.
- **CoAP (Constrained Application Protocol):** protocolo cliente-servidor similar a HTTP cuya comunicación se realiza a través de UDP. Para establecer la seguridad de la comunicación utiliza DTLS (Datagram Transport Layer Security). Este protocolo se caracteriza por tener una buena tolerancia a fallos debido a que es descentralizado.
- **DDS (Data Distribution Service):** es un protocolo basado en el estándar publicación-suscripción para ser utilizado en sistemas que requieren respuesta en tiempo real. La comunicación se realiza a través de UDP y se caracteriza por su descentralización. Para establecer la seguridad de la comunicación hace uso de TLS, DTLS, DDS.

- **OPC UA (OPC Unified Architecture):** es un protocolo de comunicación con equipos y sistemas para la recolección y control de datos, hace uso de una arquitectura cliente-servidor. Este protocolo se caracteriza por tener un alto rendimiento y una fácil configuración y mantenimiento.

2.2 Tecnologías para IPS.

Existen diversas tecnologías para la creación de un sistema de posicionamiento en interiores, la elección de una tecnología u otra depende de diversos factores como son la precisión requerida, alcance o escalabilidad. Entre las tecnologías existentes que podemos encontrar se pueden clasificar en dos grupos, estos son tecnologías inalámbricas y tecnologías sin radio. Las tecnologías inalámbricas aprovechan la infraestructura inalámbrica existente en el interior de un edificio para realizar el posicionamiento, mientras que las tecnologías sin radio pueden utilizarse sin hacer uso de esta infraestructura.

A continuación, se explican algunas de las tecnologías que se pueden aplicar en sistemas de posicionamiento en interiores en la actualidad:

- **Wi-Fi positioning system (WPS):** pertenece al grupo de tecnologías inalámbricas y se basa en la utilización de puntos de acceso a red para detectar la posición de un dispositivo con conexión Wi-Fi, a pesar de esto no es necesario que el dispositivo esté conectado al acceso de red para que este pueda ser localizado. En caso de que se esté buscando que la detección de la posición sea precisa esta no es la mejor tecnología a utilizar ya que, WPS tiene problemas de oclusión y esto puede llegar afectar a la precisión, por ello, actualmente existen tecnologías que han tratado de solucionar este problema para mejorar su precisión. Un buen ejemplo [1] de la utilidad de esta tecnología es el sistema de posicionamiento en interiores desarrollado por un grupo de investigadores pertenecientes al MIT (Massachusetts Institute of Technology).

- **Ultra Wide Band (UWB)** [4]: pertenece al grupo de tecnologías inalámbricas, es una tecnología de banda de radio (RF) capaz de ofrecer transmisión de datos de alta velocidad (hasta 400-500 Mbit/s) con una baja potencia. Su uso permite obtener posicionamientos en interiores con precisión de centímetros, para ello utiliza sensores de movimiento como acelerómetros o giroscopios, magnetómetros y sensores de precisión. Su funcionamiento se basa en el envío de pulsos de baja potencia, donde unos emisores UWB envían impulsos a unos receptores. La realización de este trabajo fin de grado se centra en la utilización de esta tecnología para realizar un estudio de la misma.

- **Bluetooth Low Energy (BLE)** [12]: pertenece al grupo de tecnologías inalámbricas, este tipo de tecnología requiere que el usuario active el bluetooth de su dispositivo y permita explícitamente su localización. Se utiliza para determinar la proximidad a objetos, no la precisión exacta de un usuario. Esta tecnología realiza las mediciones comprobando la cercanía a otro dispositivo bluetooth, pero problemas para determinar con precisión la localización de una persona. A partir de Bluetooth 5.1 permite determinar la posición de una persona con una precisión de centímetros. Es una tecnología barata y se menos afectada por el efecto de oclusión de los objetos que la tecnología Wi-Fi. Este tipo de tecnología se utiliza principalmente para detectar la cercanía de una persona a un punto determinado, por ejemplo, para detectar si una persona está cerca de un cuadro en un museo. Un buen ejemplo de esto es el sistema de posicionamiento iBeacon de Apple Inc.

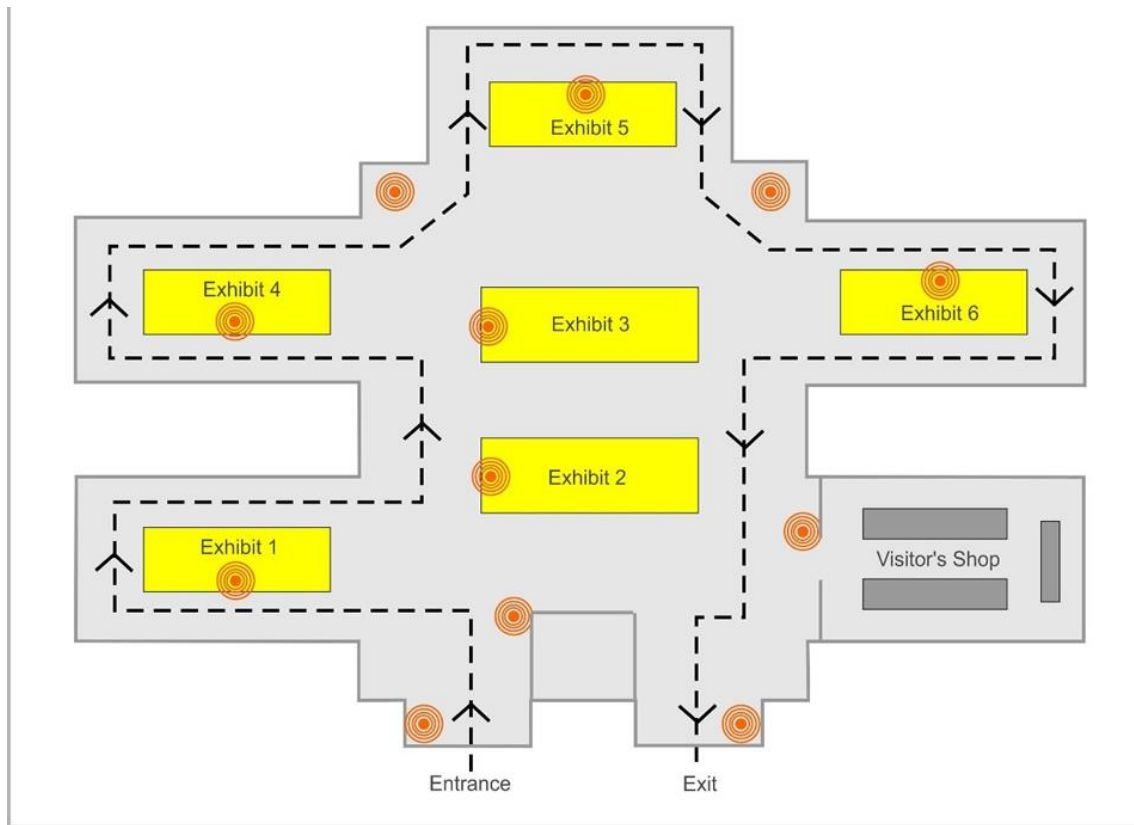


Figura 2.2: Ejemplo posicionamiento BLE.

En la Figurar 2.2 se muestra un ejemplo de un museo en el que cada pieza en exhibición dispone de un sensor con tecnología BLE que permite saber cuándo una persona está cerca de este sensor y utilizar esto para enviarle a su dispositivo móvil información sobre la pieza en exhibición a la que el usuario se ha acercado.

- **Posicionamiento magnético** [13]: pertenece al grupo de tecnologías sin radio, se basa en los datos que reciben sensores magnéticos de un smartphone. Esta tecnología se ve afectada por el campo magnético de los objetos metálicos existentes en un edificio. Ofrece precisión por un coste bajo siendo una tecnología rentable para los sistemas de posicionamiento en interiores.

2.3 Aplicaciones.

A continuación, se señalan algunas de las aplicaciones [1] [4] que pueden tener los sistemas de posicionamiento en interiores:

- **Seguimiento a tiempo real:** Mediante los sistemas de posicionamiento en interiores se puede localizar el lugar donde se encuentra una persona en todo momento, esto permite realizar un seguimiento a tiempo real de las personas que están dentro de un edificio. El sistema puede indicar en un mapa del lugar en el que se encuentra el usuario donde se está esta persona en todo momento. Esto puede ser usado, por ejemplo, para saber dónde se encuentra un determinado medico en un hospital, conocer donde se encuentra un trabajador de una fábrica cuando se le necesite, donde está un material o pedido de un usuario dentro de un almacén o incluso saber cuántas personas hay en la sala de espera de un hospital. También puede ser utilizado en una evacuación de un edificio para determinar cuántas personas quedan dentro del mismo y en que parte del edificio se encuentran. En este caso sería mejor hacer uso de la tecnología UWB debido a que se mantiene constantemente enviando la localización del usuario y dispone de una mayor precisión. Este trabajo fin de grado es un prototipo que pretende llevar a cabo esta aplicación.
- **Patrones de comportamiento:** Esta aplicación consiste en recoger las diversas posiciones que ha ocupado un usuario dentro de una zona determinada para posteriormente realizar un estudio de estas posiciones y detectar posibles patrones en los patrones de movimiento que tienen los usuarios en un edificio, por ejemplo, dentro de un museo o de un centro comercial determinar qué zonas son las que son las más y las menos visitadas por los clientes, también es posible detectar las zonas de los edificios que tienen mayor tránsito o las puertas, en caso de un centro comercial, que más se usan para salir o entrar.

- **Navegación** [14]: Los edificios son muy grandes y suelen contener diversos puntos de interés en función del tipo de usuario, con estos sistemas de posicionamiento en interiores los usuarios pueden llegar a conocer el mejor camino para llegar a un determinado lugar dentro de un edificio, por ejemplo, dentro de un edificio de una universidad se puede determinar donde se encuentran un aula específica, en una estación de tren pueden encontrar el lugar donde se encuentra un andén determinado o incluso conocer donde se encuentra el área de urgencias dentro de un hospital. A pesar de todo esto, esta funcionalidad no es la más común de estos sistemas, aunque como se ha mostrado en los ejemplos puede llegar a ser de gran utilidad, aunque está limitada al interior de un edificio.
- **Alerta por proximidad:** En estos casos se utilizan los sistemas de posicionamiento en interiores para determinar la cercanía a un lugar de tal forma que se establece un dispositivo con la tecnología correspondiente de posicionamiento en interiores en un lugar determinado y si detecta otro dispositivo determina que una persona que lleva este dispositivo está cerca, de esta forma no es necesario conocer las coordenadas en las que se encuentra el dispositivo que se detecta. Esto es muy útil, por ejemplo, para determinar en un museo que una persona está cerca de una obra de arte y enviarle al usuario un documento con información sobre esa obra. También puede ser utilizado en la construcción para que el personal de una grúa o de otra maquinaria que pueda resultar peligrosa detecte que algún obrero se acerca a la maquinaria y extirpe la precaución. Esta aplicación es más común para los sistemas Wi-Fi o los sistemas Bluetooth.

2.4 Descripción de la propuesta realizada.

Para llevar a cabo la propuesta realizada por este trabajo fin de grado se ha utilizado TREK 1000 kit evaluation, el cual es un kit que dispone de 4 antenas configurables para realizar el papel de ancla o tag. También ha sido necesario hacer uso de un terminal para ejecutar en él un software de código abierto que proporciona este kit. Por otro lado, ha sido necesario realizar la elaboración de diversos softwares para el procesamiento de los datos obtenidos de los recorridos del usuario y para poder realizar la monitorización del mismo.

Se desarrolló una aplicación móvil para el sistema operativo Android que interactúa para el envío y recepción de datos con el terminal haciendo uso del protocolo de mensajería MQTT. Este protocolo se explica detenidamente en la sección 2.4.

En primer lugar, se ha realizado toda la parte referente a la obtención y procesamiento de los datos recogidos por el sistema de posicionamiento con tecnología UWB y posteriormente se realizó el software para la monitorización del usuario. En la sección 3, en la parte de implementación se realiza la explicación del funcionamiento de los diferentes softwares desarrollados, incluyendo los lenguajes de programación y el IDE utilizado.

2.5 Protocolo de mensajería publicación/suscripción MQTT.

Para este trabajo fin de grado se ha desarrollado una aplicación móvil la cual permite realizar el seguimiento de un usuario dentro del SmartLab. Para ello la aplicación necesita obtener las coordenadas de la posición por la que se está moviendo el usuario.

Todo este proceso de obtención y envío de los datos necesarios para el funcionamiento de la aplicación se han realizado mediante el envío de mensajes utilizando el protocolo MQTT.

MQTT (Message Queue Telemetry Transport) es un protocolo de mensajería utilizado para la comunicación M2M (machine to machine) y con un consumo de ancho de banda muy bajo. Este protocolo utiliza una topología de

estrella como el que se muestra en la Figura 2.3 en la que el nodo central es denominado como broker, el cual es el encargado de gestionar toda la red y de la transmisión de los mensajes.

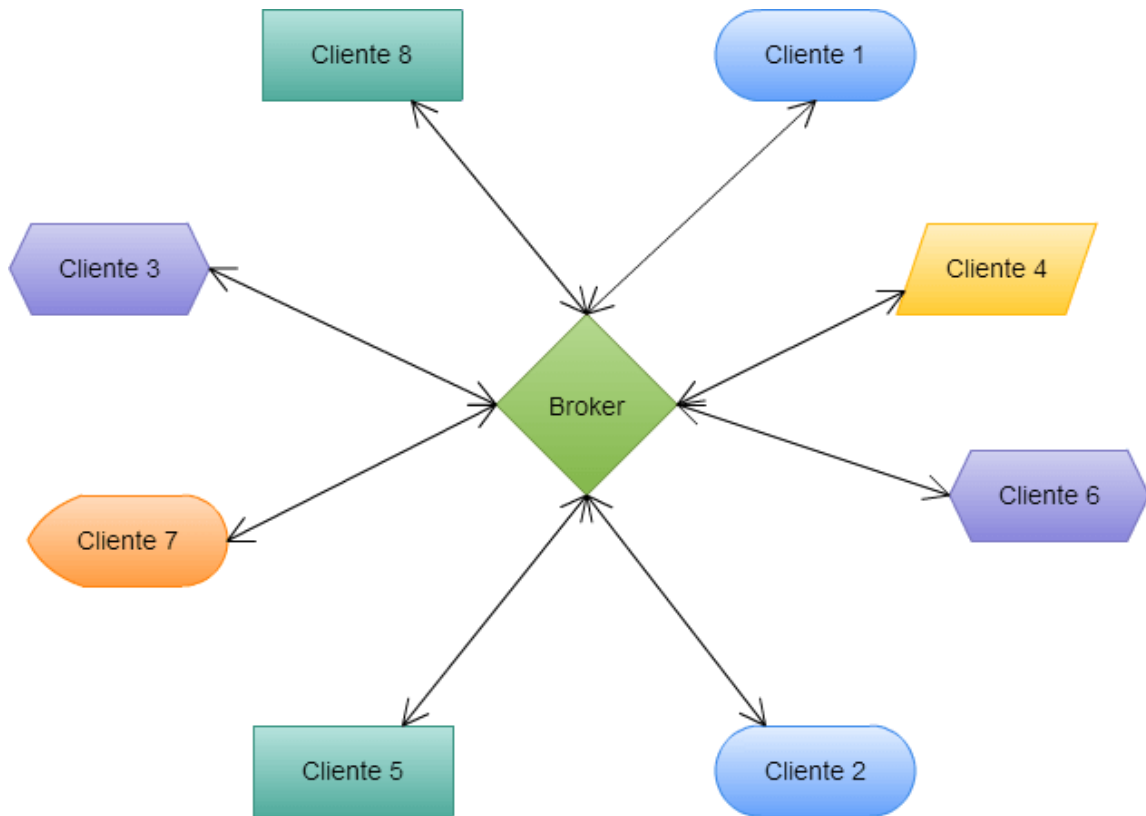


Figura 2.3: Topología de estrella MQTT

MQTT utiliza el concepto de “topics” o “temas” en español para establecer la comunicación entre emisor y receptor. Un cliente conectado al bróker publica un mensaje a través de un topic, y el resto de clientes conectados al mismo bróker han de suscribirse al mismo topic por el que se publicó el mensaje para así poder recibirlo. Además, los topics disponen de una estructura jerárquica con la que podemos establecer relaciones padre-hijo dándonos la posibilidad de, al suscribirnos a un topic padre, recibir también la información de sus topics hijos.

La Figura 2.4 muestra un ejemplo de las relaciones que se producen entre los dispositivos conectados al broker y del envío de mensajes a través del mismo.

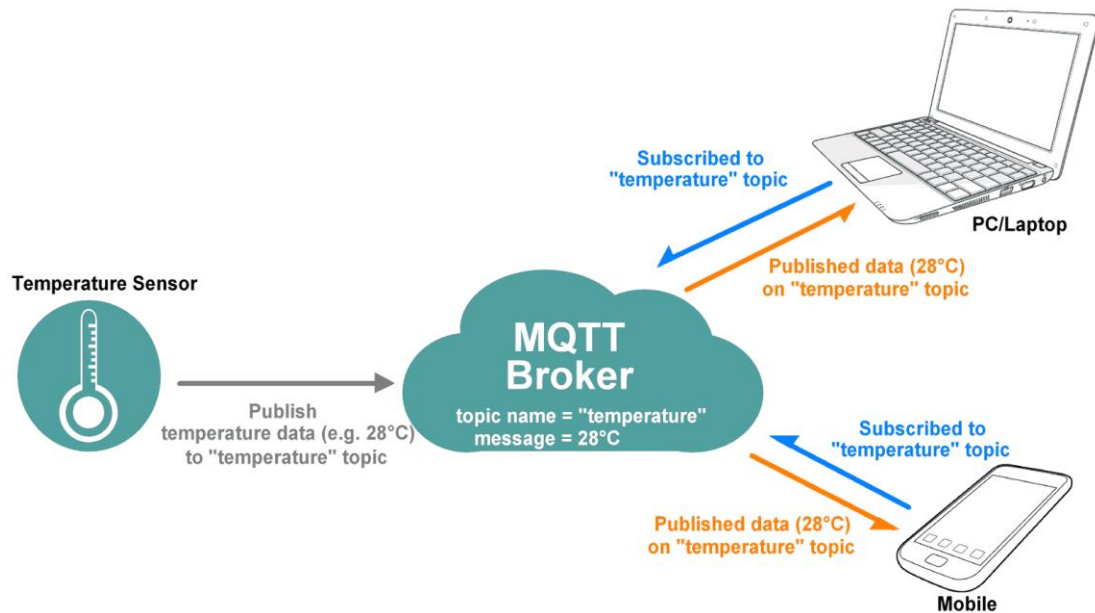


Figura 2.4: Ejemplo de funcionamiento MQTT.

2.5.1 Ventajas.

Estas son algunas de las ventajas que aporta utilizar este protocolo de mensajería:

- Facilidad de desarrollo de aplicaciones MQTT ya que, dispone de librerías para esto.
- Menor consumo de energía.
- Requiere un ancho de banda mínimo.
- Seguridad y calidad del servicio

- Aporta robustez al sistema.
- Útil para trabajar con aplicaciones IoT.
- Permite una gran cantidad de clientes conectados al broker.

2.5.2 Inconvenientes.

Estas son algunos de los inconvenientes que presenta la utilización del protocolo MQTT:

- Utiliza una topología de estrella, esto hace que la comunicación dependa únicamente del correcto funcionamiento de un punto específico.
- Los mensajes que envía este protocolo viajan sin cifrar en texto plano, esto hace que cualquier cliente que se conecte al broker y acceda al topic puede tener acceso a los mensajes enviados.

2.5.3 Conclusión.

En definitiva, el protocolo de mensajería empleado para la realización de este sistema de posicionamiento en interiores con tecnología UWB aporta al sistema una gran cantidad de ventajas. Principalmente estas ventajas han sido motivo de que MQTT haya sido elegido para el propósito de este sistema de posicionamiento en interiores.

2.6 Laboratorio utilizado.

El laboratorio utilizado para trabajar con la tecnología UWB ha sido el Centro de Estudios Avanzados en Tecnología de la Información y Comunicación (CEATIC), el cual se encuentra localizado el edificio C6 de la universidad de Jaén en la dependencia 109 del edificio C6.

La distribución del laboratorio se muestra en la Figura 2.5, y su superficie ocupa 25 metros cuadrados. Para trabajar la tecnología UWB en este lugar, se

ha considerado que el laboratorio este dividido en 4 secciones entrada, salón, habitación y cocina.



Figura 2.5: Mapa Laboratorio CEATIC.

Este laboratorio está totalmente equipado para simular un pequeño hogar, lo que permite trabajar con esta tecnología en un entorno lo más realista posible. El salón cuenta con un sofá, una televisión inteligente y además cuenta con una estación de trabajo la cual es la encargada de recoger información de los diversos sensores que se encuentran en este lugar, incluyendo la información que llegara a los receptores de la tecnología UWB colocada en el lugar.

Por otro lado, el dormitorio está equipado con una cama, un inodoro y un pequeño aseo. Y la cocina cuenta con un frigorífico, un microondas, una lavadora, un lavavajillas, etc.

Como he mencionado antes, la estación de trabajo es la que se encarga de gestionar los diversos sensores que hay por toda la sala, incluyendo la tecnología UWB. Para trabajar con esta tecnología en este entorno se han utilizado tres antenas equipadas con tecnología UWB y una tag también equipada con tecnología UWB. La tag va siempre con el usuario y se encarga de enviar información a las anclas que actúan como receptores, una de las anclas está conectada a la estación de trabajo, por lo que está situada muy cerca de la misma, la otra ancla está situada en el dormitorio cerca de la cama y la última ancla se encuentra localizada en la cocina cerca del microondas.

Para finalizar con esta sección se muestran algunas imágenes del interior del smartLab donde se puede apreciar con mayor claridad su distribución como son la Figura 2.6.



Figura 2.6: SmartLab.

2.7 Tecnología UWB empleada.

Como ya se ha comentado anteriormente la tecnología empleada para este sistema de posicionamiento ha sido UWB (ultra-wide-band), concretamente se ha utilizado el kit TREK 1000 el cual es utilizado para mantener un seguimiento en tiempo real. Este kit contiene principalmente cuatro placas EVB100 PCB y cuatro antenas que se colocan en dichas placas. Además, el kit incluye un software de código abierto denominado Decarange cuyo funcionamiento ha sido modificado en este trabajo fin de grado para obtener la localización computada en tiempo real. El precio [15] de este kit es de 899.00\$, 789.43€.

A continuación, en la Figura 2.7, se muestra un ejemplo de las antenas que proporciona este kit para trabajar.



Figura 2.7: Ejemplo real de una de las anclas utilizadas.

Las ventajas de utilizar esta tecnología son algunas de las mencionadas a continuación:

- Alta velocidad en la transferencia de datos.
- Bajo consumo de energía.
- Gran robustez ya que tolera las interferencias.

La principal desventaja de esta tecnología es su elevado precio, un buen ejemplo de esto es el precio del kit utilizado en este trabajo fin de grado. Sin embargo, existen otros dispositivos más económicos como por el ejemplo las placas de desarrollo DWM1001 en la cual cada placa cuesta [16] 35.99\$, 31.60€ y también sería necesario comprar una Raspberry Pi cuyo precio [17] sería de 33.73€. En total, para poder montar el mismo sistema que con el kit TREK 1000, el precio sería de 160€ aproximadamente, esto teniendo en cuenta sería necesario comprar 4 placas DWM1001 y una raspberry. Por tanto, tal y como se ha mencionado anteriormente el coste de esta última opción es bastante más económico que hacer uso de TREK1000.

- **Adaptación del software de localización DECAWAVE**

Este kit incluye un software de código abierto, denominado DecaRange, para la recepción de, entre otra información, de las coordenadas de la tag. DecaRange muestra una ventana de trabajo similar a la que se puede observar en la Figura 2.8, donde aparecen las distintas anclas fijas y los movimientos del ancla móvil, aunque el desplazamiento de esta última no es capaz de mostrarlo con buena precisión en todos los casos, apareciendo datos incompletos y anómalos en los logs. Este software permite grabar logs almacenando en una carpeta todos ellos con toda la información que reciben las diferentes antenas, la Figura 2.9 es un ejemplo de los registros que almacenan los logs creados por DecaRange.

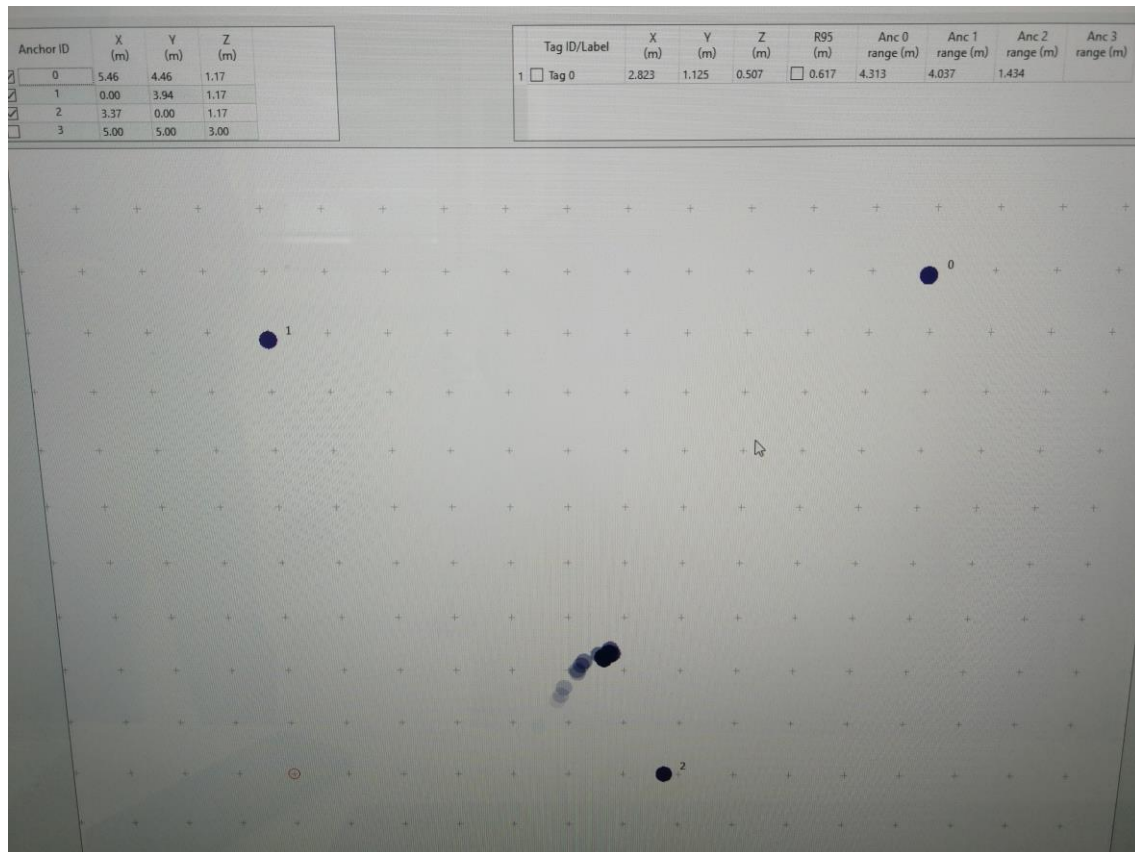


Figura 2.8: Ventana principal DecaRange.

```

T:095644135:DecaRangeRTLS:LogFile:Ver. 2.10 TREK:Conf:Anchor2:0:Chan2
T:095644170:AP:0:5.46:4.46:1.168
T:095644170:AP:1:0:3.94:1.168
T:095644170:AP:2:3.37:0:1.168
T:095644170:AP:3:5:5:3
T:095644170:RR:0:0:1952:1952:83:1013
T:095644170:RR:0:1:4435:4435:83:1013
T:095644170:RR:0:2:4477:4477:83:1013
T:095644170:LE:0:331:82:[4.1513,4.14259,-0.309351]:1999:4411:4467:-1
T:095644170:TS:0 avx:4.16552 avy:4.12501 avz:-0.328837 r95:0.0621623

```

Figura 2.9: Log obtenido con DecaRange.

2.7.1 Suscripción a la localización del usuario en tiempo real desde un dispositivo móvil.

Para la monitorización del usuario se hizo uso de la propia aplicación móvil ya desarrollada anteriormente para la obtención de las coordenadas reales del recorrido, y se le añadió una nueva funcionalidad que permitía a un usuario dado enviar, haciendo uso del protocolo MQTT y del log proporcionado por la modificación realizada sobre DecaRange, a dicha aplicación móvil las coordenadas del usuario para que se dibujara el recorrido del mismo por el mapa que contiene la aplicación móvil. El recorrido que realiza el usuario se puede ver en la Figura 2.10.

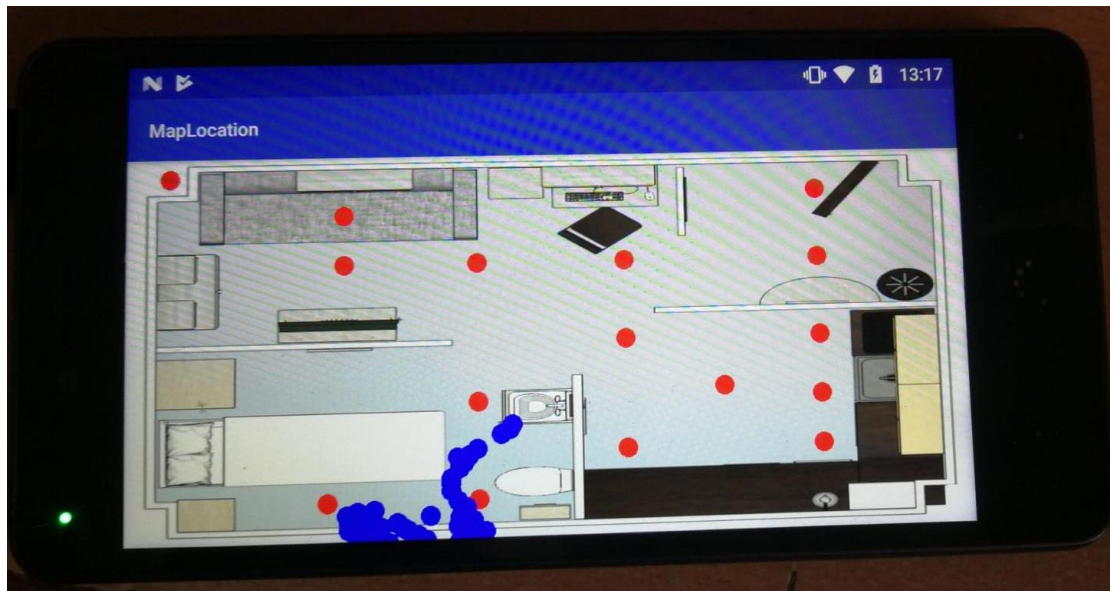


Figura 2.10: Monitorización del usuario

3. Proceso de ingeniería del software.

A lo largo de este capítulo se va a describir el proceso de ingeniería [18] del software llevado a cabo para el desarrollo del software empleado en este trabajo fin de grado.

Previamente a la descripción de este proceso se va a incluir las fases de la ingeniería del software que se van a tratar durante el desarrollo de este capítulo.

3.1 Fases de la ingeniería del software.

Las fases de las que consta el proceso de ingeniería del software son las siguientes:

- **Análisis de requerimientos:** En esta fase se especifica, verifica, refina y modela las peticiones del usuario. Es decir, se llevará a cabo la extracción de los requisitos del producto software y de sus funcionalidades.
- **Diseño y arquitectura:** Definición del funcionamiento general del software de forma que satisfaga los requerimientos previamente analizados.
- **Programación:** Desarrollo del software previamente diseñado de tal forma que satisfaga los requerimientos establecidos y permita obtener los resultados deseados.
- **Pruebas:** Comprobación de la correcta realización de las tareas por parte del software, cumpliendo con los requerimientos establecidos.

3.2 Análisis de requerimientos.

Para realizar el análisis de los requerimientos de forma clara y concisa se va a proceder a realizar una división en dos grupos de estos requerimientos, donde estos dos grupos serán requerimientos funcionales y requerimientos no funcionales.

- **Requerimientos funcionales:** son aquellos que describen los servicios que el sistema es capaz de llegar a realizar, indicando los parámetros de entrada y los casos de uso.
- **Requerimientos no funcionales:** son aquellos que especifican criterios que permiten estudiar y evaluar los diferentes servicios ofrecidos por un sistema determinado.

A continuación, se va a proceder a describir cada uno de los diferentes requerimientos de los que consta el sistema de posicionamiento en interiores desarrollado en este trabajo fin de grado.

3.2.1 Requerimientos funcionales.

Para este trabajo fin de grado han sido considerados los siguientes requerimientos funcionales:

- El sistema tiene que ser capaz de recolectar y almacenar todos los datos que las anclas generen sobre el posicionamiento del usuario haciendo únicamente uso de la tecnología proporcionada en el kit TREK 1000 de UWB.
- El sistema debe permitir al usuario la correcta gestión de las anclas activas tanto fijas como móviles (editar/visualizar).
- El sistema debe de permitir al usuario utilizar los ficheros logs que se generen durante un recorrido para el estudio y monitorización de los mismos.

- El sistema debe de mostrar el recorrido realizado por un usuario haciendo uso de la aplicación móvil desarrollada para esta finalizada y de los datos recogidos en log tras haber sido filtrado.
- El sistema tiene que facilitar al usuario la recolección de las coordenadas del mundo real mediante el uso de la aplicación desarrollada para así hacerle menos complejo el estudio de los datos obtenidos por el programa DecaRange.
- El sistema debe permitir al usuario interrumpir el envío de las coordenadas del mundo real mediante una marca que actúa como interruptor, situada en la esquina superior izquierda de la aplicación móvil, cuando la persona a localizar por el sistema finalice su recorrido en la zona indicada.

3.2.2 Requerimientos no funcionales.

Para realizar la descripción de los requerimientos no funcionales se va a realizar una división de estos en requerimientos físicos y requerimientos software.

3.2.2.1 Requerimientos físicos.

Debido a que el sistema hará uso de una arquitectura cliente-servidor, es necesario una estación de trabajo que permita al usuario hacer uso de los servicios del sistema.

Los requisitos físicos de los que deberá de disponer este terminal deberán de ser:

- Procesador: 2,6 GHz
- RAM: 2GB
- HDD: 10GB
- Conexión a internet.

Los requerimientos físicos de los que deberá de disponer el dispositivo móvil que ejecute la aplicación desarrollada para mejorar el estudio realizado de los datos y monitorizar al usuario deberán de ser:

- Procesador: 1,4 GHz
- RAM: 2GB
- Almacenamiento: 30 MB de almacenamiento disponible.
- Conexión a internet.
- Resolución de pantalla:1280x720

3.2.2.2 Requerimientos software.

Respecto a los requerimientos software que debe de tener el terminal son los siguientes:

- Software Mosquitto.
- Versión de Java 1.8 o superior
- Sistema operativo Windows 8/10 o Linux

Haciendo referencia a los requerimientos software del dispositivo móvil tenemos los siguientes:

- Sistema operativo Android versión 7.1.1

3.2.2.3 Requerimientos de la interfaz

Los requerimientos de la interfaz de la aplicación móvil son los siguientes:

- **Flexibilidad:** Debe ser capaz de ejecutarse en cualquier dispositivo que cumpla los requerimientos tanto físicos como software especificados anteriormente.
- **Robustez:** Debe ser capaz de soportar cualquier fallo producido por los usuarios que trabajen con ella.

- **Facilidad:** Ha de ser simple, intuitiva y fácil de utilizar para que cualquier usuario pueda trabajar con ella sin que esta le presente problemas, aunque se trate de la primera vez que hace uso de la misma.

3.3 Casos de uso

Los casos de uso permiten llevar a cabo la especificación del comportamiento y la comunicación de un sistema a través de la interacción con el usuario cuando ocurre un evento. Los casos de uso correspondientes a este sistema de posicionamiento en interiores son los siguientes:

- Casos de uso MapLocation.
 Dentro de la aplicación móvil se puede distinguir principalmente los siguientes casos de uso.
 - Caso de uso 1: Monitorización.
 En este caso de uso de la Figura 3.1 se muestra la interacción que lleva a cabo la aplicación móvil con el usuario al que le muestran los datos recogidos del recorrido de una persona determinada, la interacción con la persona a la cual se está monitorizando y finalmente la interacción con el broker que es esencial para que aplicación móvil reciba los datos a mostrar a través de este.

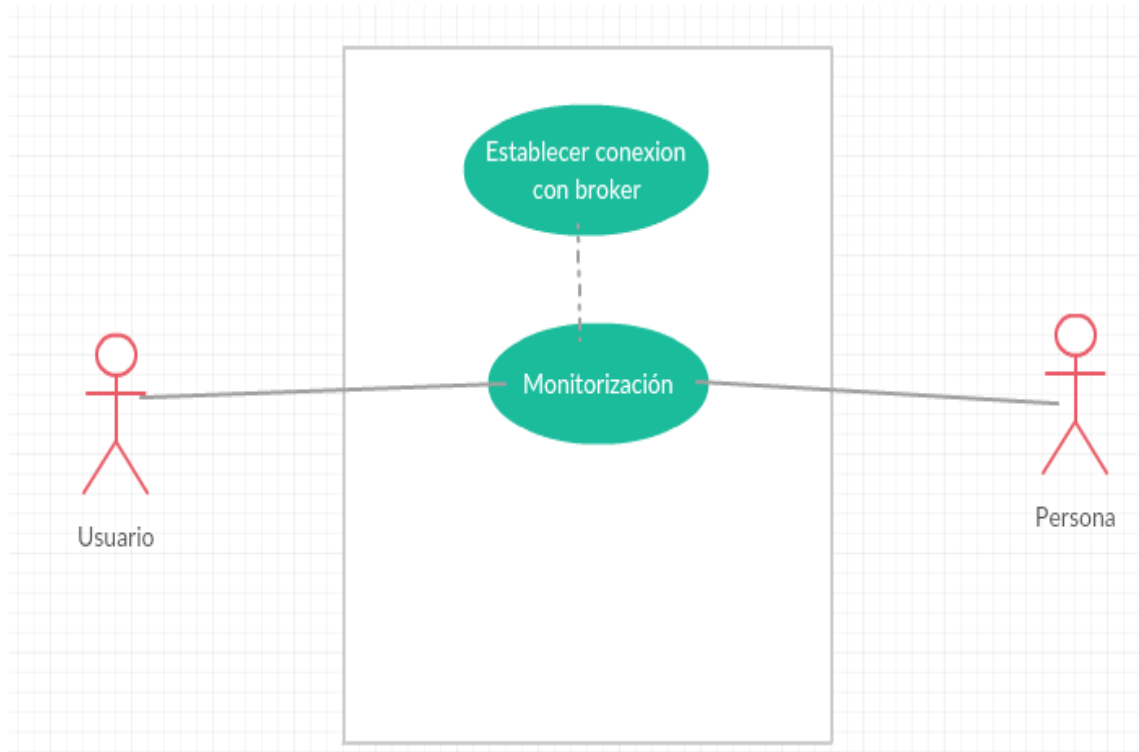


Figura 3.1: Diagrama Caso de uso 1.

Actores	Usuario, Persona
Condiciones de entrada	El mensaje recibido por MQTT debe contener al menos coordenadas x e y, y el id o mac de la tag.
Eventos	La aplicación muestra mediante círculos de color azul el recorrido realizado por la una persona. (E1)
Excepciones	E1: La ip a la que se conecta la aplicación no corresponde con la del broker.

Tabla 3.1: Caso de uso 1.

- Caso de uso 2: Recogida de datos reales.

En este caso de uso de la Figura 3.2 se muestra la interacción que lleva a cabo la aplicación móvil con el usuario para realizar la recolección de datos que muestra al usuario de los datos recogidos del recorrido realizado para llevar a cabo diversos estudios sobre el funcionamiento de esta tecnología UWB. Para ello la aplicación requiere del broker al que le enviará las coordenadas recogidas durante el recorrido.

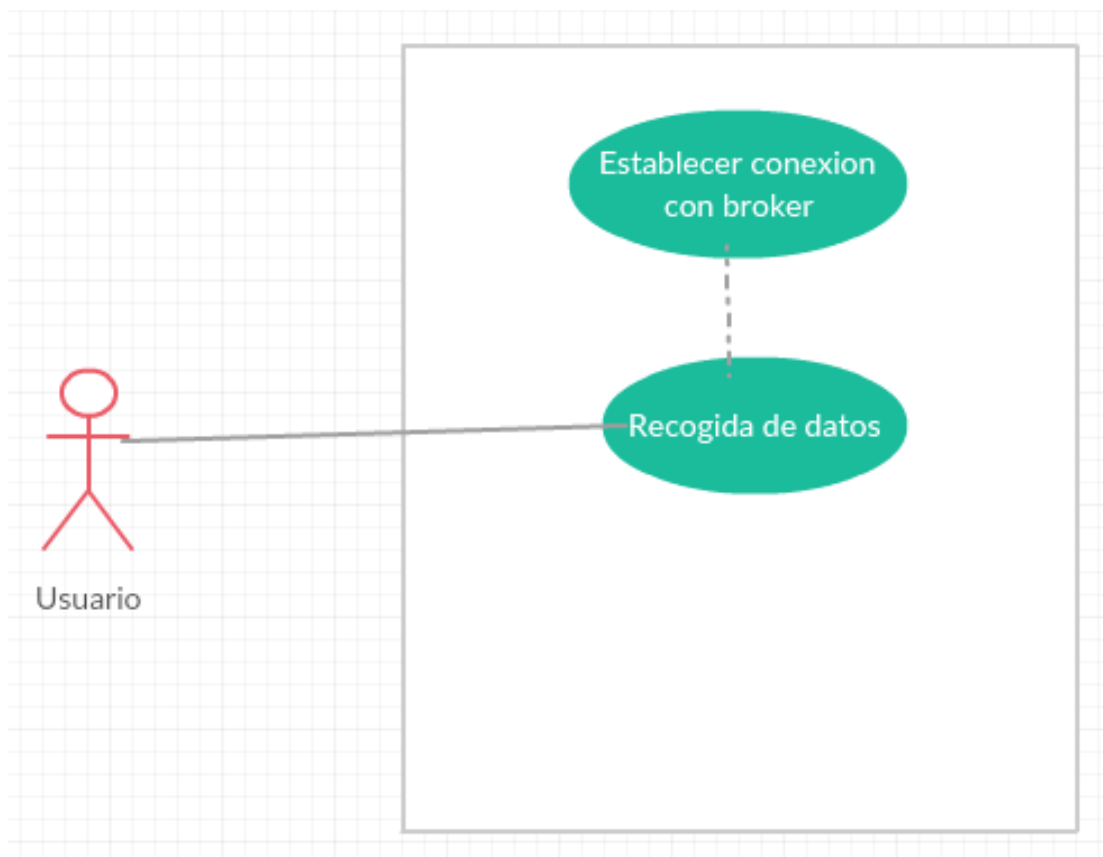


Figura 3.2: Caso de uso 2

Actores	Usuario
Condiciones de entrada	Es necesario iniciar primero la aplicación móvil y antes de pulsar cualquier marca se requiere que se inicie la actividad de LocationServer.
Eventos	La aplicación comienza el envío de las coordenadas indicadas a una estación de trabajo que está ejecutando el servidor MQTT. (E1)
Excepciones	E1: La ip a la que se conecta la aplicación no corresponde con la del broker.

Tabla 3.2: Caso de uso 2.

- Casos de uso LogSender.

En este caso de uso de la Figura 3.3 se muestra la interacción llevada a cabo por el software LogSender con el usuario para realizar el envío de las coordenadas para que la aplicación móvil pueda realizar la monitorización del usuario. Además, muestra la interacción con los diferentes logs creados para cada una de las diferentes coordenadas y la interacción con el broker que establece la conexión para el correcto funcionamiento del sistema de posicionamiento en interiores establecido.

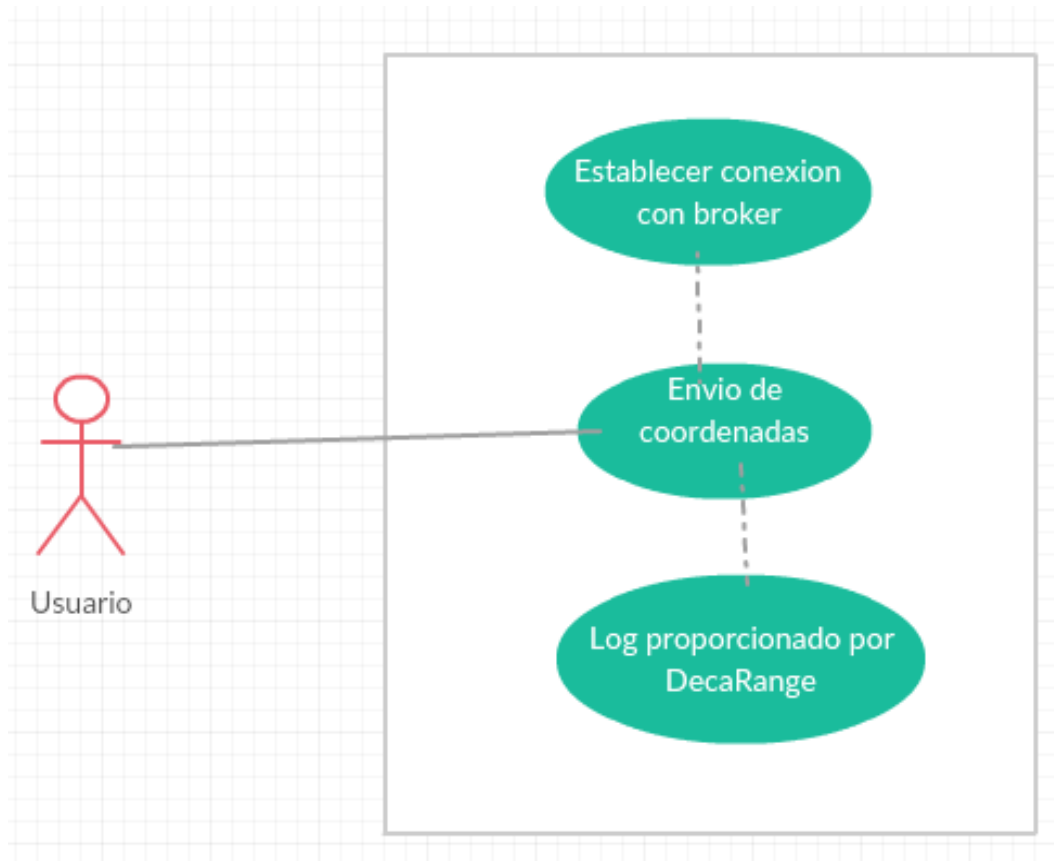


Figura 3.3: Caso de uso LogSender

Actores	Usuario
Condiciones de entrada	Requiere que la aplicación móvil esté en funcionamiento y que el log sea el proporcionado por DecaRange.
Eventos	Envía el log recibido de DecaRange tras ser modificado a través de MQTT a la aplicación móvil.(E1)
Excepciones	E1: La ip a la que se conecta la aplicación no corresponde con la del broker.

Tabla 3.3: Caso de uso LogSender.

- Casos de uso LocationServer.

En este caso de uso de la Figura 3.4 se muestra la interacción llevada a cabo por el software LocationServer con el usuario para realizar la recogida de las coordenadas emitidas por la aplicación móvil. Para ello el usuario establecerá al software en espera hasta que se lleve a cabo la recepción de las coordenadas. Cuando se reciben las coordenadas el sistema crea un log en el cual se almacenan dichas coordenadas. Además, es necesario que se establezca la correspondiente conexión con el broker para llevar a cabo todo el funcionamiento del sistema correctamente.

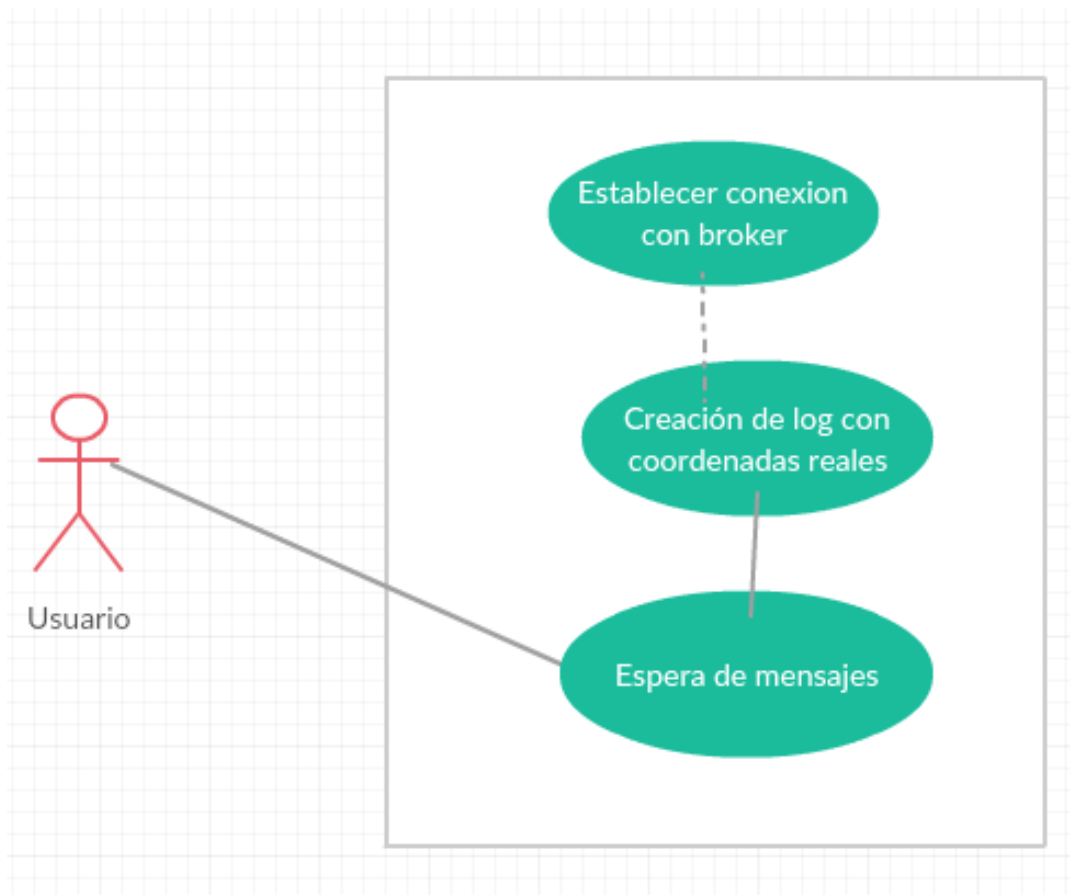


Figura 3.4: Caso de uso LocationServer.

Actores	Usuario
Condiciones de entrada	Es necesario que se conecte a la ip del broker y que cuando la aplicación móvil finalice este reciba la señal de parada.
Eventos	• Indica la espera del paso de mensajes (E1) • Crea un fichero con las coordenadas recibidas y las id de las marcas seleccionadas.(E1)
Excepciones	E1: La ip a la que se conecta la aplicación no corresponde con la del broker.

Tabla 3.4: Caso de uso LocationServer.

3.4 Diseño y arquitectura.

3.4.1 Diseño de clases.

En esta sesión se va a nombrar y describir las funciones que tienen las distintas clases que forman parte de la aplicación móvil desarrollada. Además, se ha incluido un diagrama de clases utilizando el lenguaje unificado de modelado (UML) con la Figura 3.5 que permite visualizar las relaciones entre las distintas clases implementadas. Las clases creadas han sido las siguientes:

- **Mark:** Esta clase ha sido definida para representar las marcas que aparecerán sobre el mapa utilizado en la aplicación móvil y contiene diferentes atributos que representan los datos que recibe o envía dicha aplicación.
- **CustomView:** Como se ha mencionado antes, será necesario dibujar las marcas que se almacenen para que el usuario las pueda visualizar, por tanto, esta clase permitirá dibujar las marcas recibidas en el mapa para mostrar el recorrido realizado por una persona y además es la

encargada de almacenar las marcas pulsadas por el usuario para el estudio de los datos.

- **AsyncSubMqtt:** Esta clase se encarga de recibir las coordenadas del recorrido que está siguiendo una persona conectándose al topic correspondiente por mqtt y de almacenarlas para que posteriormente la clase CustomView pueda hacer uso de estas para dibujarlas en el mapa.
- **MyAgents:** Como se ha mencionado anteriormente, la clase CustomView es la encargada de almacenar los datos de las marcas indicadas por el usuario durante el estudio de los datos. Esos datos de las marcas almacenados por CustomView son utilizados por la clase MyAgents para publicar en el topic correspondiente las marcas seleccionadas y enviarlas al servidor mqtt. Esta clase representa un agente asíncrono que se encarga de relizar el envío en una tarea sin bloquear el resto.
- **MainActivity:** Por último, esta clase será la encargada de crear las conexiones al broker, haciendo uso de una dirección mac y de un puerto para indicar a MyAgents y a AsyncSubMqtt el topic al que tienen que conectarse. Además, será la encargada de crear la aplicación con el mapa del smartLab inicialmente vacío, es decir, sin ninguna marca e ira añadiéndolas conforme se vaya avanzando en la ejecución de la aplicación y vaya haciendo llamadas al resto de clases. Tras finalizar la ejecución de la aplicación esta también será la encargada de destruirla.

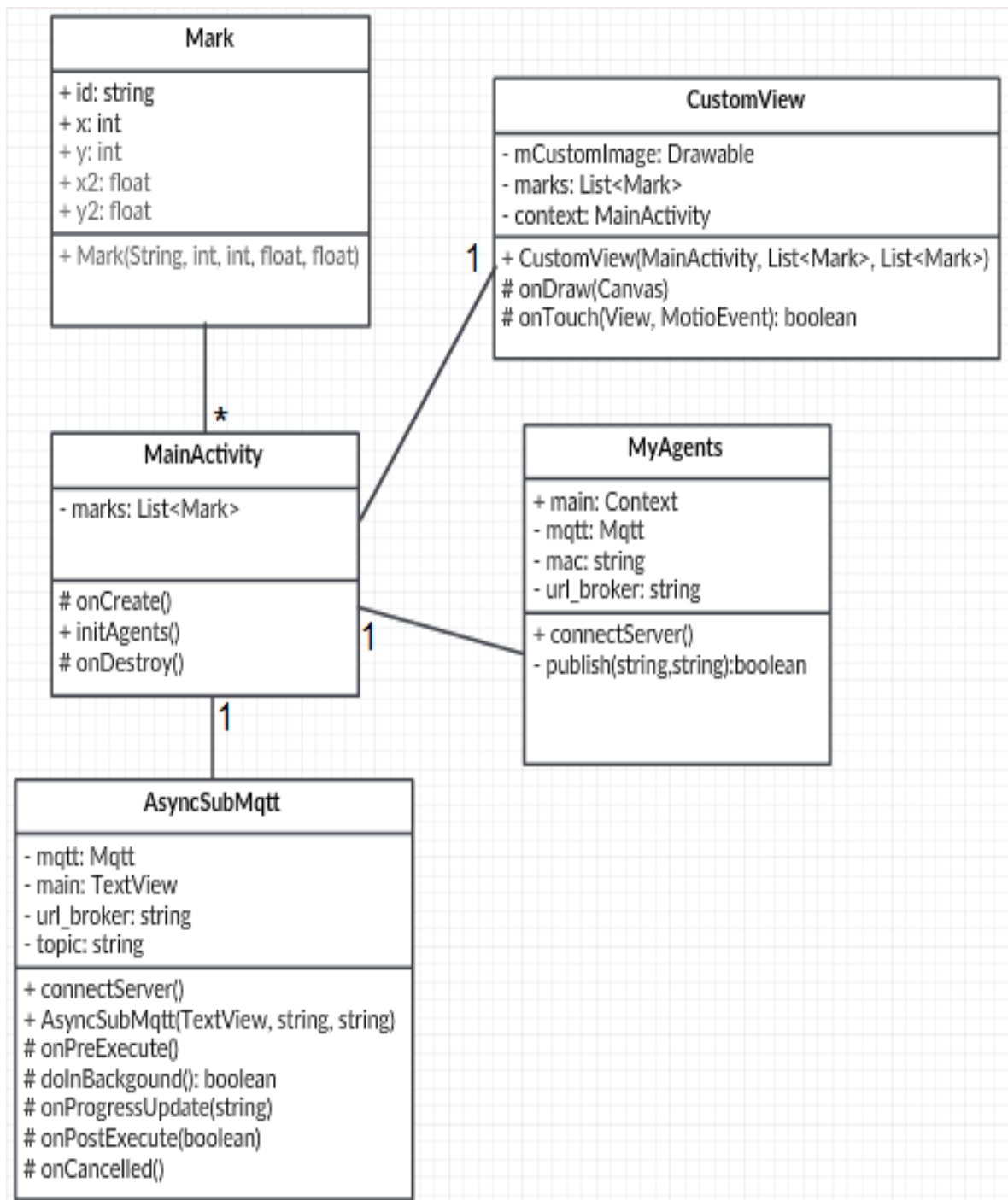


Figura 3.5: Diagrama de clases.

Símbolo	Descripción
+	Público
#	Protegido
-	Privado

Tabla 3.5: Símbolos del diagrama de clases.

3.4.2 Arquitectura.

Este sistema de posicionamiento en interiores utiliza un modelo de arquitectura cliente-servidor, donde interviene el protocolo MQTT explicado en la sección 2.3 de este trabajo fin de grado. El protocolo MQTT de utilizado en esta arquitectura funciona a su vez sobre el protocolo de red TCP/IP.

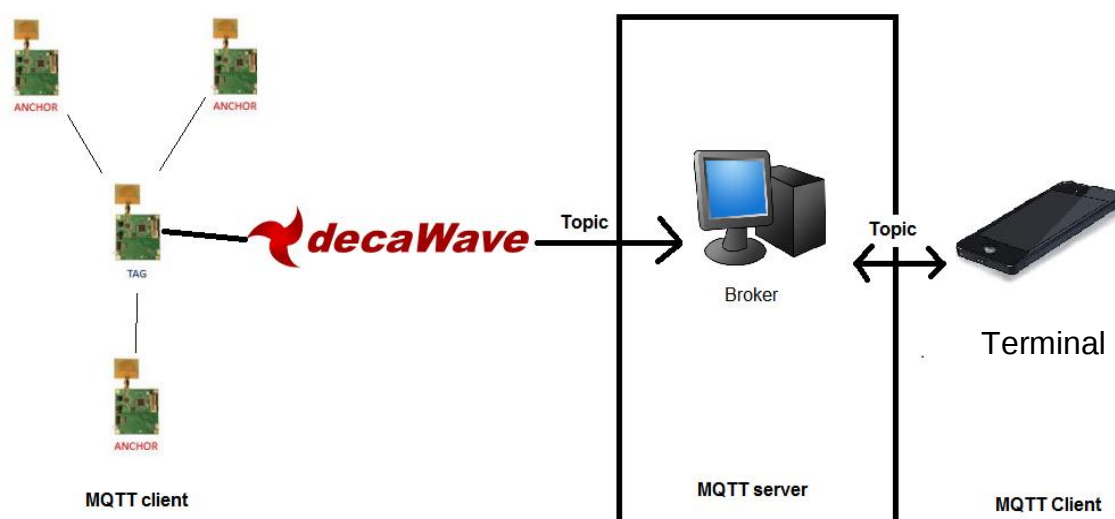


Figura 3.6: Esquema de la arquitectura del sistema

En la Figura 3.6 se muestra la arquitectura que se ha establecido haciendo uso de los diferentes elementos que forman parte del sistema de posicionamiento en interiores establecido en este trabajo fin de grado. El terminal actúa como publicador de MQTT publicando en un topic las coordenadas reales del recorrido realizado por un usuario determinado y también actúa como suscriptor de MQTT suscribiéndose en un topic para recoger las coordenadas que muestra en la aplicación móvil para la monitorización del usuario.

El broker se encargará de establecer la conexión entre el terminal móvil y el software decawave para que así el terminal pueda obtener las coordenadas del recorrido que realiza el usuario y mostrarlo en pantalla.

3.5 Implementación

La programación de la aplicación móvil se ha llevado a cabo en el entorno de desarrollo de Android Studio haciendo uso del lenguaje de programación Android java.

Por otro lado, la programación del software para procesar los datos de estudio para esta tecnología se ha realizado en el IDE de netbeans, en concreto en la versión 8.2, haciendo uso del lenguaje de programación java con la versión de java 1.8.

La programación del software utilizados para que la aplicación móvil envíe y reciba los datos por MQTT también se ha realizado en el IDE de Netbeans 8.2 haciendo uso del lenguaje de programación java, aunque en este caso el proyecto creado fue de tipo maven para tener acceso a la librería paho MQTT que permitiría el envío de mensajes utilizando este protocolo.

Por último, las modificaciones realizadas sobre el software de código abierto DecaRange se realizó en el entorno Qt y el lenguaje de programación que se utilizó fue C ya que, DecaRange estaba programado en este inicialmente y se trató únicamente de una pequeña modificación.

Los softwares desarrollados en esta etapa y sus funcionalidades son los siguientes:

- **LogProcessor:** Este software se encarga de procesar cada uno de los logs generados por DecaRange para filtrarlos y eliminar los datos que no son de utilidad incluyendo también las coordenadas que no se han recibido de forma correcta desde la tag y que por lo tanto no se han podido almacenar y se han guardado como nan (Not A Number).

- **MapLocation:** Esta es la aplicación desarrollada para trabajar con los datos obtenidos, es decir, dispone de dos funcionalidades. La primera funcionalidad es monitorizar al usuario mostrando por la pantalla del dispositivo móvil el recorrido que está realizando este por un lugar en concreto. La segunda funcionalidad consiste en mostrar todas las marcas que se establecieron en el suelo del smartLab para que cuando el usuario pase por esta se indique a la aplicación que la envíe al servidor donde se almacenara y se utilizará para posteriormente realizar el estudio de los datos obtenidos. Para enviar y recibir estos datos la aplicación hace uso del protocolo de mensajería MQTT.

- **LocationApp:** Es una aplicación móvil desarrollada para el sistema operativo Android la cual hace uso del protocolo MQTT para recibir los datos de las coordenadas indicadas por el usuario en la aplicación móvil cuando la persona pasa por encima de una de las marcas. Cada marca que recibe es almacenada en un fichero de texto para posteriormente poder realizar un estudio adecuado de los datos recogidos por el software DecaRange.

- **LogServer:** Hace uso del protocolo MQTT para enviar a la aplicación los datos que esta deberá de permitir mostrar en la pantalla del dispositivo móvil para la monitorización correspondiente del usuario. Los datos enviados por este software son obtenidos de los ficheros de texto que genera DecaRange tras haber modificado su código para que genere un fichero para cada coordenada que las antenas perciben.

3.6 Pruebas.

Todas las pruebas realizadas han tenido lugar en el laboratorio del CEATIC descrito en la sección 2.2.

La realización de esta parte en la ingeniería del software es de suma importancia ya que, va a permitir evaluar la calidad del sistema de posicionamiento en interiores desarrollado y además permitirá comprobar si es necesario mejorar este sistema.

Previamente a la realización de las pruebas una persona, la cual será monitorizada por el sistema, deberá llevar encima la tag durante todo el recorrido que realice la misma tal y como se muestra en la Figura 3.7.



Figura 3.7: Ejemplo de un usuario realizando el recorrido con la tag encima

Para realizar la recolección de los datos necesarios para posteriormente realizar algunas pruebas se han colocado marcas en el suelo del SmartLab con el objetivo de comprobar lo precisas que eran las lecturas de las antenas con respecto a la realidad, para ello se han medido manualmente las coordenadas de las marcas con respecto a un punto inicial a medida que eran colocadas, este punto inicial se consideró que fuera la esquina inferior izquierda del laboratorio que se encuentra justo en la entrada del mismo. La disposición de las marcas en el suelo del SmartLab se muestra en la Figura 3.8 junto a la disposición de las tres anclas fijas, donde las marcas nombradas como AX representan dichas antenas. Esta figura es un boceto del SmartLab que permite tener una clara distribución de las marcas y las anclas utilizadas, utilizado como base para establecer la interfaz de la aplicación móvil que se muestra en la Figura 3.9.

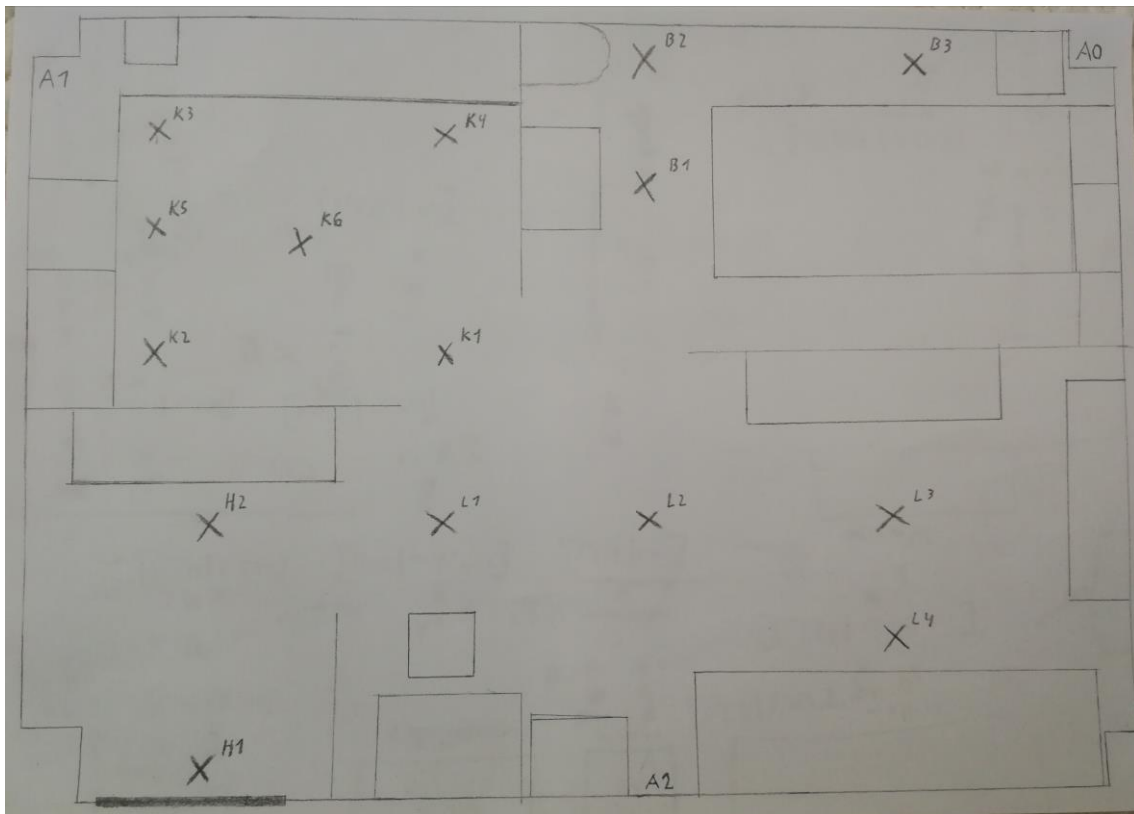


Figura 3.8: Mapa de la distribución de las marcas.

Para la recolección de los datos se utilizó DecaRange inicialmente y posteriormente fue necesario aplicarle a este una modificación en su código para así poder quedarnos únicamente con los datos de las coordenadas para que fuera más sencillo realizar la monitorización del usuario posteriormente.

Una vez desplegado el sistema de DecaRange era necesario realizar una calibración de estos para comprobar la fidelidad de los datos obtenidos por el sistema. El sistema de DecaRange usa trilateración [19] cuyo funcionamiento es determinar las posiciones relativas de personas u objetos haciendo uso de la geometría, es un método similar a la triangulación con la diferencia de que la trilateración hace uso de posiciones conocidas de unos puntos y la distancia del objeto a los puntos de referencia. En la trilateración es necesario que existan al menos 3 puntos de referencia. Además, este sistema es sensible al ruido, obstáculos, etc, sin embargo, la tecnología UWB empleada se caracteriza por su robustez y precisión frente al resto de tecnologías empleadas en los sistemas de posicionamiento en interiores.

Para esta comprobación la calibración hizo uso de las marcas establecidas anteriormente y de las mediciones que se recolectaron de las mismas a partir de un punto de referencia. De esta forma, el usuario puede anotar la posición de referencia real (ground truth) para poder compararla con diferentes modelos de localización.

La calibración de los datos se realizó inicialmente de forma manual utilizando las grabaciones de video realizadas sobre las escenas para comprobar en que instante de tiempo el usuario pasaba por encima de una de las marcas, posteriormente este proceso se automatizo dando lugar al desarrollo de la aplicación móvil la cual consistía en el mapa del smartLab con las mismas marcas que se muestran en la Figura 3.8, las cuales se crearon con círculos de color rojo. Esta aplicación móvil publica por MQTT los datos a un servidor que almacena en un fichero los datos recibidos.

Haciendo uso del log obtenido por DecaRange y de las coordenadas de las marcas proporcionadas por la aplicación móvil se contrastaron los datos

obtenidos por ambos. Los resultados obtenidos se pueden observar en las Figura 3.9 y Figura 3.10.

Ambas Figuras contienen dos gráficas donde la primera hace referencia al recorrido real seguido por el usuario y la segunda hace referencia al recorrido recogido por el software de calibración. Por ello, la segunda gráfica contiene una mayor cantidad de puntos que marcan el recorrido seguido por el usuario.

La Figura 3.9 se corresponde con el recorrido que se especifica en la Tabla 3.6 donde el recorrido se inicia en la puerta del SmartLab, el usuario se desplaza siguiendo las marcas establecidas en la Figura 3.8 hasta el sofá, donde permanece sentado unos segundos y se levanta para continuar el recorrido, posteriormente el usuario se desplaza hasta la cocina donde simula el proceso de prepararse la comida y comer, a continuación, el usuario se desplaza hasta el dormitorio donde finaliza el recorrido en el lado izquierdo de la cama.

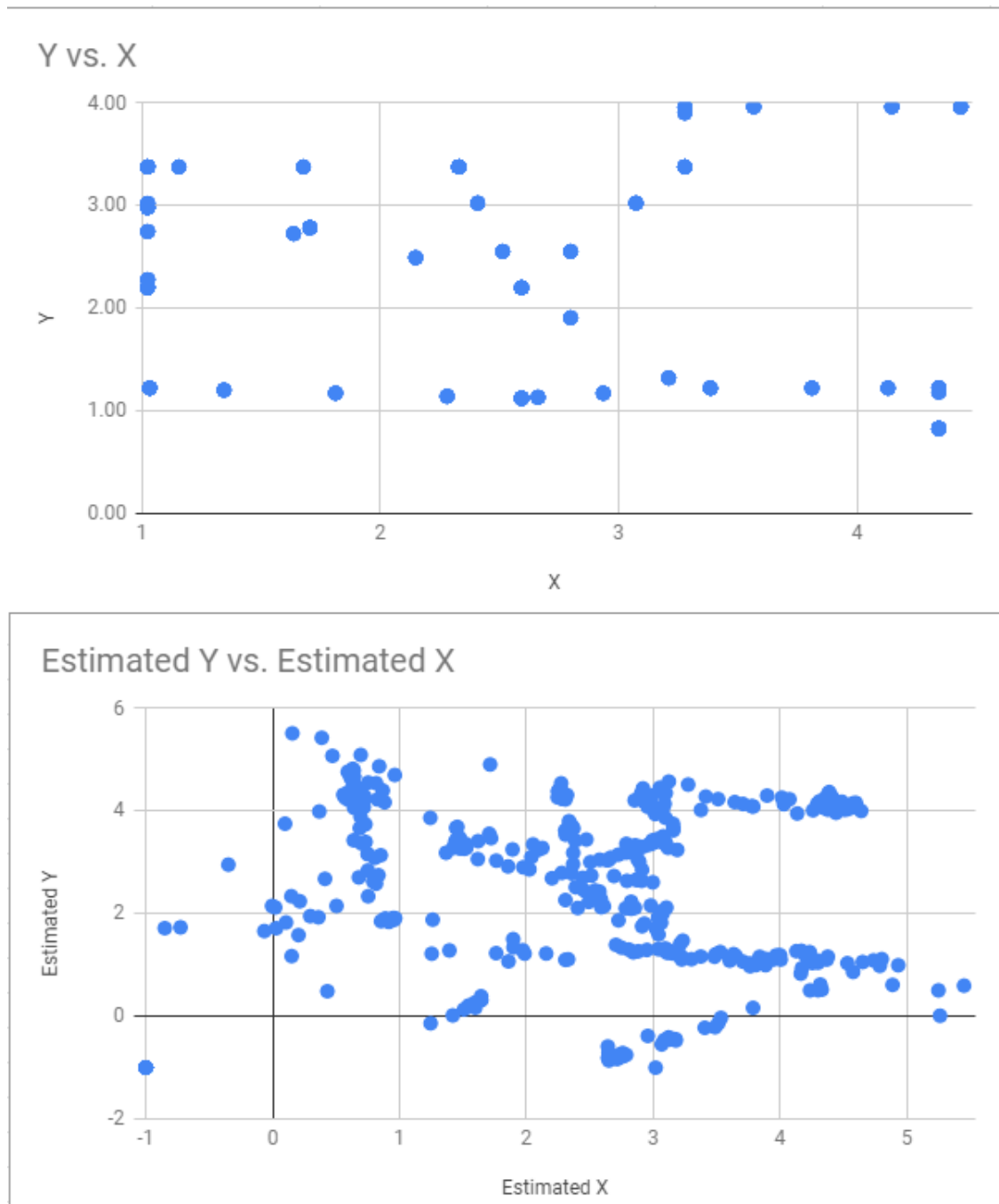


Figura 3.9: Gráfica primer recorrido

La Figura 3.10 se corresponde con el recorrido especificado en la Tabla 3.7, donde el recorrido se inicial en el lado izquierdo de la cama simulando que el usuario acaba de empezar el día, a continuación, el usuario se desplaza hasta la cocina donde simula prepararse el desayuno y desayunar, posteriormente el usuario se desplaza hasta el sofá donde permanece sentado unos segundos y finaliza el recorrido en la puerta del SmartLab.

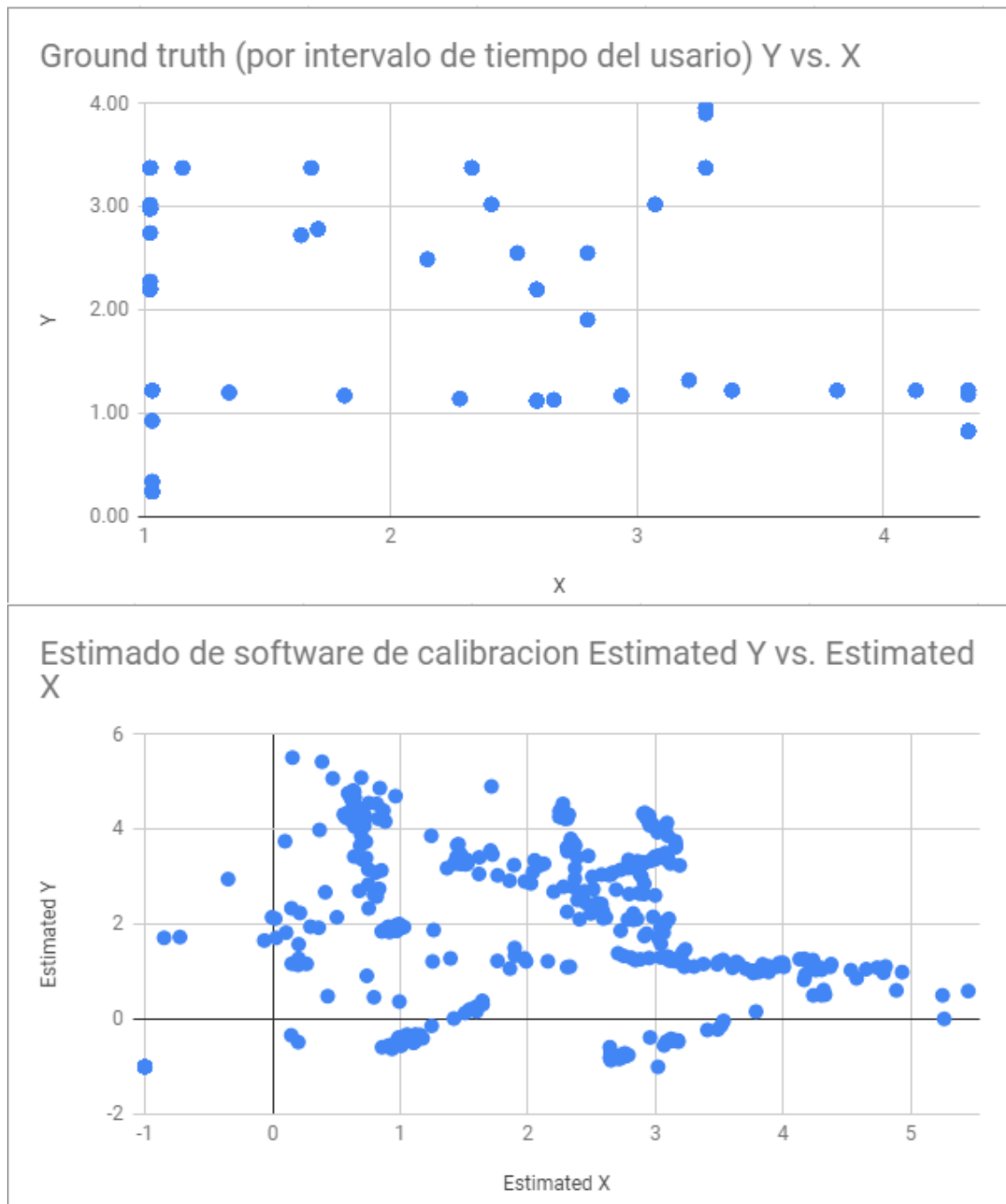


Figura 3.10: Gráfica segundo recorrido.

A continuación, se va a proceder a explicar de forma más detallada todo este proceso:

Se establecieron recorridos determinados que un usuario debía realizar mientras llevaba encima la tag. Durante una de las pruebas el usuario comenzó su recorrido en la puerta del smartLab, desde allí se desplazó por las marcas indicadas en la Tabla 3.6 por orden de aparición.

Marcas recorridas
H1
H2
L1
L2
L3
L4
L3
L2
K1
K2
K5
K3
K4
K6
K1
B1
B2
B3

Tabla 3.6: Primer recorrido del usuario.

Marcas recorridas
H1
H2
L1
K1
K6
K2
K5
K3
K4
K1
L1
L2
L3
L4
L3
L2
B1
B2
B3

Tabla 3.7: Segundo recorrido del usuario.

A continuación, se han realizado grabaciones en video de diferentes rutas por el smartLab mientras que a su vez se realizaba, con DecaRange, la grabación del log con los datos que iba recogiendo las antenas.

Como se puede observar en la Figura 2.6, el log proporcionado por DecaRange contiene mucha información que para este trabajo no es relevante ya que, este se centra principalmente en realizar un seguimiento de una persona por un espacio específico. Por tanto, fue necesario crear un software que dado un log extraído de DecaRange permitiera filtrar toda la información de este para así poder obtener únicamente los datos que eran de especial

relevancia para este estudio. Este software se denomina LogProcessor el cual devolvía ficheros de log con la estructura que se muestra en la Figura 3.9.

Timestamp	Range1	cRange1	Range2	cRange2	Range3	cRange3	Estimated X	Estimated Y	Estimated Z
100737113	1560	1560	4519	4519	4486	4486	4.34015	4.20902	-0.0616632
100737396	1574	1574	4514	4514	4472	4472	4.37415	4.2324	0.0713157
100737676	1597	1597	4514	4514	4453	4453	4.3675	4.21653	0.0613619
100737955	1583	1583	4538	4538	4486	4486	4.36322	4.19135	0.0387022
100738232	1602	1602	4538	4538	4482	4482	4.38452	4.21943	0.0316216
100738513	1593	1593	4524	4524	4486	4486	4.3798	4.21084	0.0115049
100738795	1607	1607	4538	4538	4416	4416	4.36967	4.22283	0.0310844
100739076	1485	1485	4528	4528	4528	4528	4.385	4.14076	0.0169518

Figura 3.9: Datos obtenidos con LogProcessor

Una vez procesados los datos obtenidos de los logs proporcionados por DecaRange es necesario calibrar estos datos para compararlos con las coordenadas del mundo real. Para ello se hizo uso de los videos grabados sobre el recorrido del usuario estimando donde se encontraba el usuario en cada instante de tiempo conociendo las coordenadas de las marcas que habían sido medidas previamente. En un principio esta estimación se realizó de forma manual y posteriormente se desarrolló una aplicación móvil que permitiese automatizar todo este proceso.

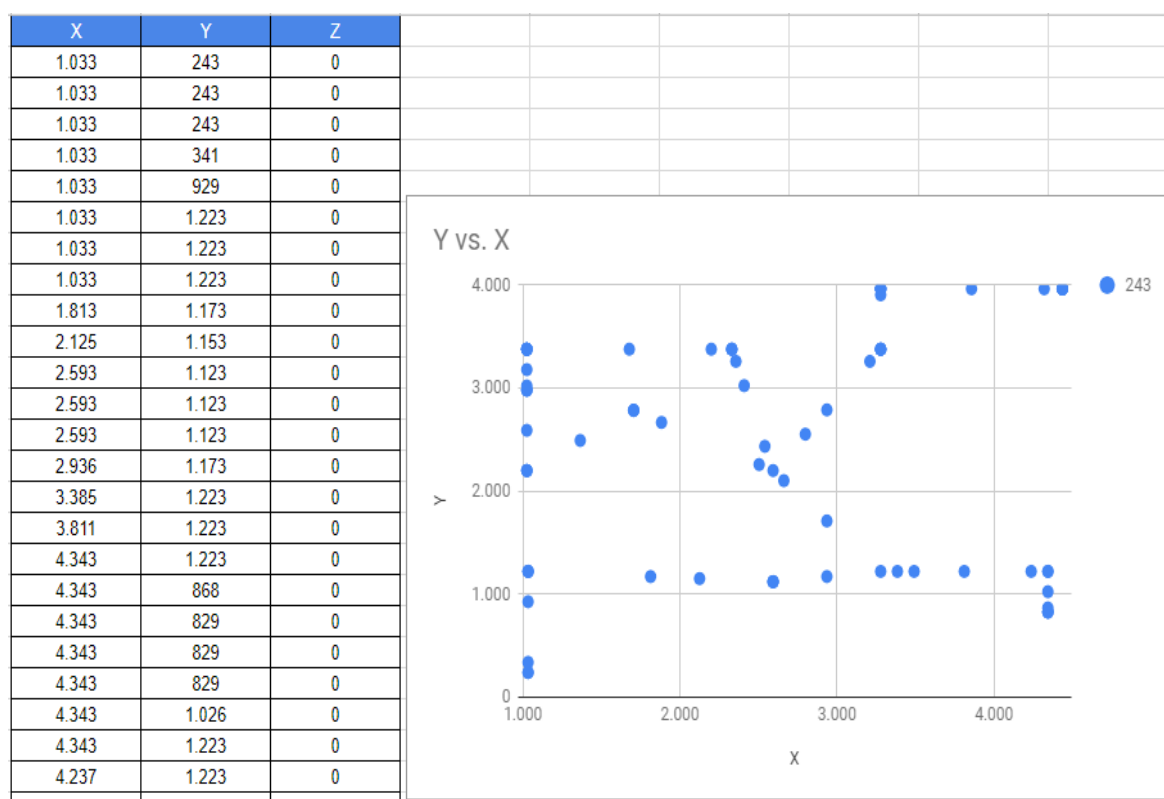


Figura 3.10: Recorrido del usuario

Una vez obtenidos los datos reales y los datos procesados por el programa LogProcessor estos se utilizarán para continuar con la investigación. Es necesario crear escenas de manera que los datos generados se puedan utilizar para realizar una predicción del camino que seguiría el usuario en base a la información que estaba en la escena. Para ello se desarrolló otro software denominado SignalSorter cuyo trabajo es combinar la información real y la obtenida por el programa de medición una vez procesada para crear un directorio “Escena” formado por los archivos necesarios para el entrenamiento.

Para recoger los datos de forma más precisa se desarrolló una aplicación móvil cliente tal y como se muestra en la Figura 3.11 basada en el protocolo de mensajería MQTT, esta aplicación móvil dependía de un servidor que se ejecutaba en un ordenador el recibía los datos que le enviaba la aplicación móvil. Ambos estaban suscritos al topic “marks” para que la aplicación móvil pudiese enviar las coordenadas de las marcas y el timestamp de las mismas al servidor, permitiendo de esta manera automatizar la recogida

de las coordenadas reales y facilitando el procesamiento de los datos. La aplicación móvil se denominó MapLocation



Figura 3.11: App móvil.

Los círculos de color rojo que se muestran en la Figura 3.8 se corresponden a las marcas que se colocaron en el suelo del smartLab las cuales nos permiten obtener las coordenadas del mundo real, la marca superior izquierda sirve de interruptor para detener el envío de mensajes entre la aplicación móvil y el servidor ya que, mientras no se indique una nueva marca la aplicación móvil continúa enviando al servidor la última marca que se indicó.

Una vez finalizada esta parte de la aplicación, ya no era necesario recoger manualmente las coordenadas reales de la escena. Simplemente es necesario tocar las marcas en la aplicación a medida que el usuario pasa por encima de ella en su recorrido y la información se recoge de manera automática en un fichero. Puesto que la idea en un futuro es extender este sistema para poder trabajar en un entorno con multiocupación el software identifica las coordenadas que recibe en función de un identificador único, este identificador es la dirección mac del dispositivo que las manda.

1541408989734	B3	4.435	3.964
1541408989734	B3	4.435	3.964
1541409004175	B2	3.278	3.964
1541409004175	B2	3.278	3.964
1541409004175	B2	3.278	3.964

Figura 3.12: Coordenadas obtenidas con MapLocation.

Puesto que ya se puede obtener las coordenadas de manera automatizada haciendo uso de MapLocation, se desarrolló un software final uniendo varias funcionalidades de los softwares independientes creados anteriormente y añadiendo varias mejoras. El software LocationApp recibe como entrada un fichero de log procesado de DecaRange y un fichero de coordenadas reales de la aplicación móvil MapLocation. La idea de este software es comparar la localización real (ground truth) y la estimada. Para ello, se aplica un preprocesamiento de los ficheros para sincronizar el inicio de sus datos en tiempo 0. Es decir, para cada fichero su primer dato es el instante 0 y el resto se calcula como la diferencia del tiempo en cualquier instante y el instante inicial. Posteriormente construye la escena de manera similar a como ya venía haciendo antes con la principal diferencia de que se ha mejorado la fórmula de variación por incrementos para que a partir de las posiciones fijadas por el usuario se simulan posiciones intermedias entre marcas que van gradualmente de una a otra, para dar coherencia de escena continuada de forma similar a como lo realiza el software DecaRange. Para ello se observa el tiempo entre marcas y se pondera la posición intermedia entre la marca anterior y la siguiente.

Para finalizar se realizó de nuevo el recorrido establecido en la Tabla 3.3 haciendo uso esta vez del programa DecaRange modificado, se puso en ejecución el broker y se comenzó a mandar por MQTT las coordenadas que recibía DecaRange de la tag para que la aplicación móvil mostrara el recorrido del usuario por el mapa de la aplicación. De esta forma se comprobó que se había completado el objetivo de monitorizar al usuario durante los recorridos

que realizará añadiendo una funcionalidad más a esta aplicación móvil mediante MQTT.

También se realizó una comprobación, en la que el usuario salió de la zona donde se encontraban las anclas fijas hacia el pasillo. El resultado obtenido es que como no era detectado por todas las anclas del smartlab, tan solo era detectado por una de ellas, el sistema no era capaz de localizarlo. Por tanto, se comprobó que para poder realizar una buena localización del usuario se necesita que el usuario se encuentre dentro del rango detectable por todas las anclas.

4. Conclusiones y líneas de trabajos futuros.

En este trabajo fin de grado se ha llevado a cabo el estudio sobre la tecnología UWB, el procesamiento de los datos proporcionados por esta tecnología y la elaboración de un sistema de posicionamiento en interiores haciendo uso de la tecnología UWB, donde se ha desarrollado principalmente un prototipo de aplicación móvil para la monitorización y seguimiento de un usuario.

Tras haber realizado todo este trabajo se puede concluir que se ha llegado a cumplir los objetivos establecidos para este trabajo fin de grado.

Como se ha podido mostrar a lo largo de todo este trabajo los sistemas de posicionamiento en interiores disponen de una gran variedad de aplicaciones y pueden llegar a ser de gran ayuda en situaciones críticas para las personas como es la evacuación de un edificio. Aunque también se ha visto que hay diversas tecnologías que se pueden utilizar, cada una con sus peculiaridades, por tanto, en función del trabajo para el cual se vaya a desarrollar el sistema de posicionamiento en interiores, es necesario realizar un estudio y elegir la tecnología que mejor se adapte a los requisitos que deba cumplir el sistema en un entorno específico.

Tras haber estudiado estos sistemas, sus aplicaciones y tecnologías entre otras cosas, creo que estos sistemas pronto estarán disponibles en lugares como museos, hospitales, construcciones, etc.

Por otro lado, como líneas de trabajo futuras es importante mencionar que estos dispositivos permiten la multiocupación, es decir, es posible establecer un sistema de posicionamiento en interiores utilizando la tecnología UWB para realizar el seguimiento de varios usuarios añadiendo tantas tag o ancla móvil, como usuarios se encuentren en la zona de estudio, al sistema actual de estudio y recoger diversos datos de las coordenadas de los desplazamientos de estos usuarios para posteriormente trabajar con ellos haciendo uso de este prototipo de aplicación.

Los mensajes enviados por MQTT al prototipo de aplicación móvil incluyen un identificador único para cada usuario, para poder realizar esto se llevó a cabo una modificación sobre el código de DecaRange acerca de cómo guarda los datos el programa de código abierto que incluye el kit TREK 1000 de UWB, para añadir este identificador único. Es decir, utilizando esto y realizando la adaptación correspondiente a la aplicación móvil se podría añadir la monitorización de diversos usuarios de una forma no muy compleja.

Otra posible línea de trabajo futuro consistiría en dar soporte a otros sistemas operativos móviles como por ejemplo iOS ya que, esta aplicación móvil y su funcionalidad completa ha sido desarrollada únicamente para el sistema operativo móvil Android.

5. Anexo.

En este anexo se incluye como parte de la documentación de este sistema de posicionamiento en interiores con tecnología UWB un manual de usuario en el que se describe detenidamente cada una de las peculiaridades de este sistema y como debe ser utilizado tanto para realizar estudios de los datos obtenidos como para monitorizar a las personas que se desplazan por el entorno determinado. Para el correcto funcionamiento del sistema se debe de tener en cuenta cada una de las partes que se mencionan a continuación como son: la configuración de los dispositivos hardware UWB, la recolección y estudio de los datos y, para finalizar, la monitorización del recorrido realizado por el usuario.

5.1 Manual de usuario.

5.2.1 Configuración de los dispositivos hardware UWB.

Es fundamental realizar la correcta configuración de cada una de las placas que proporciona TREK1000 evaluation kit para que actúe como tag (antena móvil) o como anchor (antena fija), estas disponen de una pantalla que muestra diferente información en caso de ser tag o anchor tal y como se puede observar en la Figura 5.1. Si el dispositivo fue configurado como tag (Figura 5.3) en la pantalla muestra la distancia en metros del anchor más cercado junto con el id del anchor correspondiente y si está configurada como anchor (Figura 5.4) la pantalla muestra el id de la misma y la distancia en metros a la que se encuentra la tag.

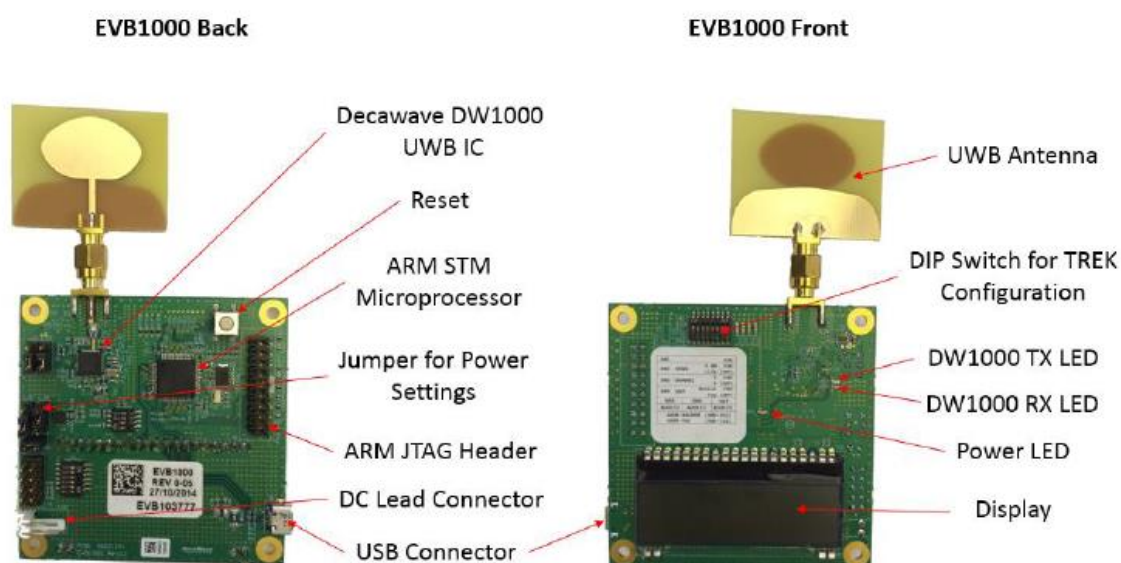


Figura 5.1: Antena UWB.

En la figura 5.2 se muestra la funcionalidad que tiene asignada cada uno de los ocho pines de los que dispone cada dispositivo, el primer pin ha de estar activado para que el dispositivo funcione. Para activarlo se desplazará hacia arriba en caso de que no esté en esta posición. El segundo pin permite cambiar entre dos velocidades distintas para cambiar la velocidad a la que se realiza la comunicación entre los diferentes dispositivos del sistema.

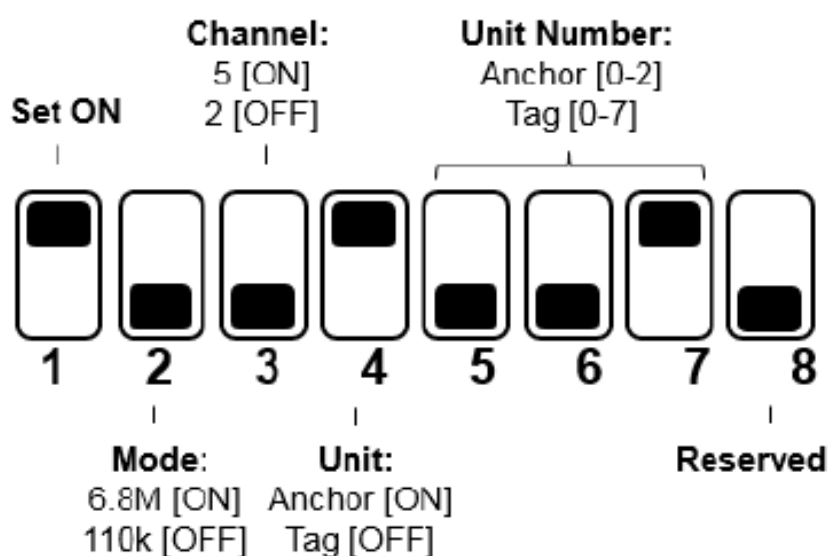


Figura 5.2: Función de cada pin.

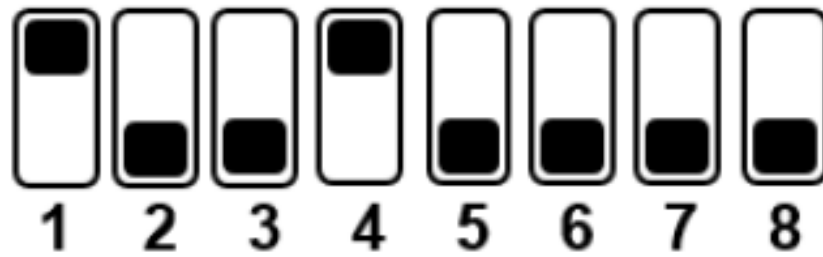


Figura 5.3: Configuración DIP para ancla fija

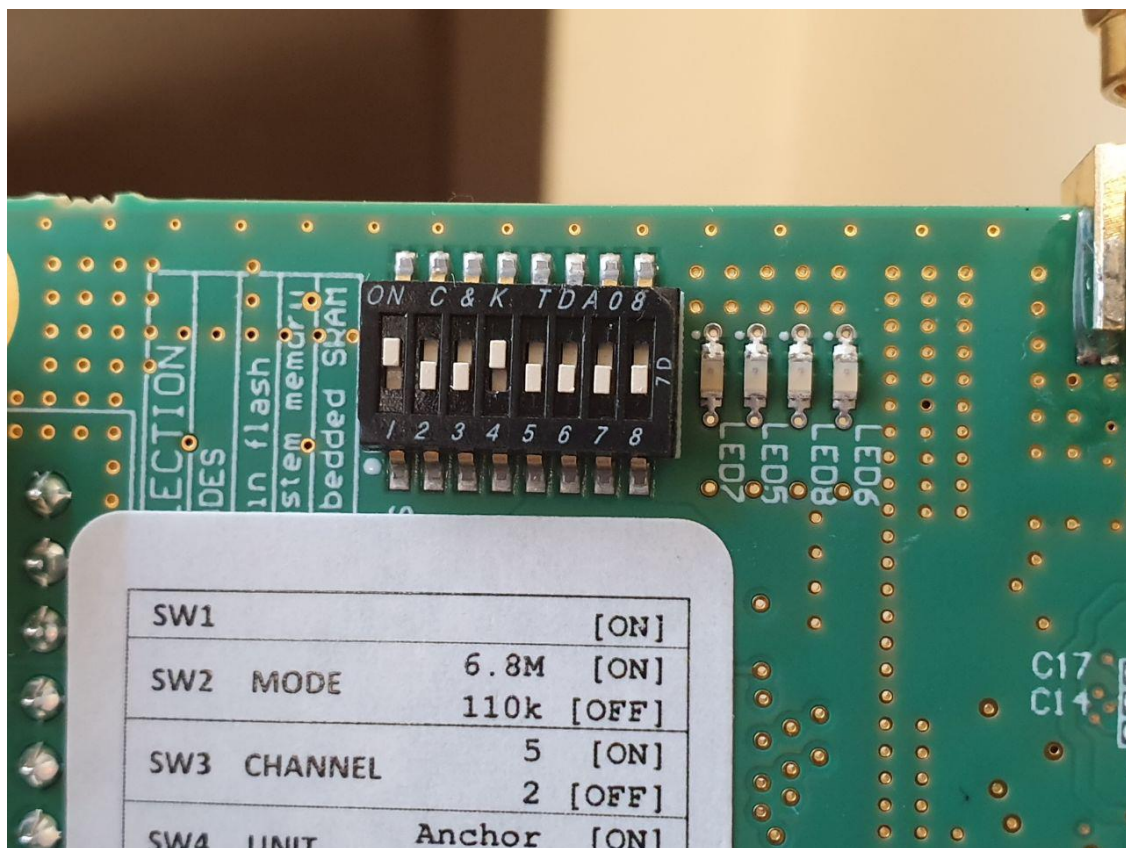


Figura 5.4: Configuración del ancla del sistema

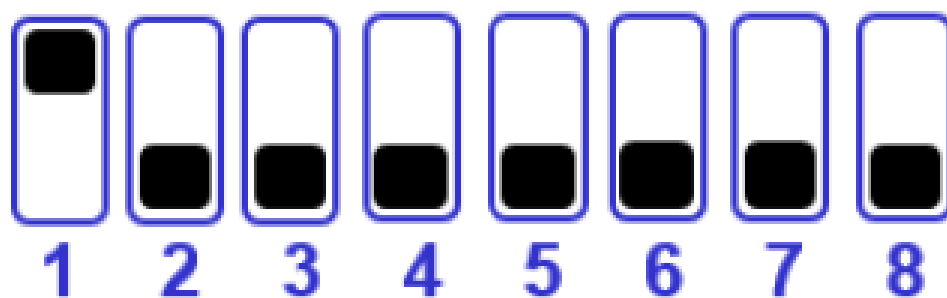


Figura 5.5: Configuración DIP para ancla móvil (tag)

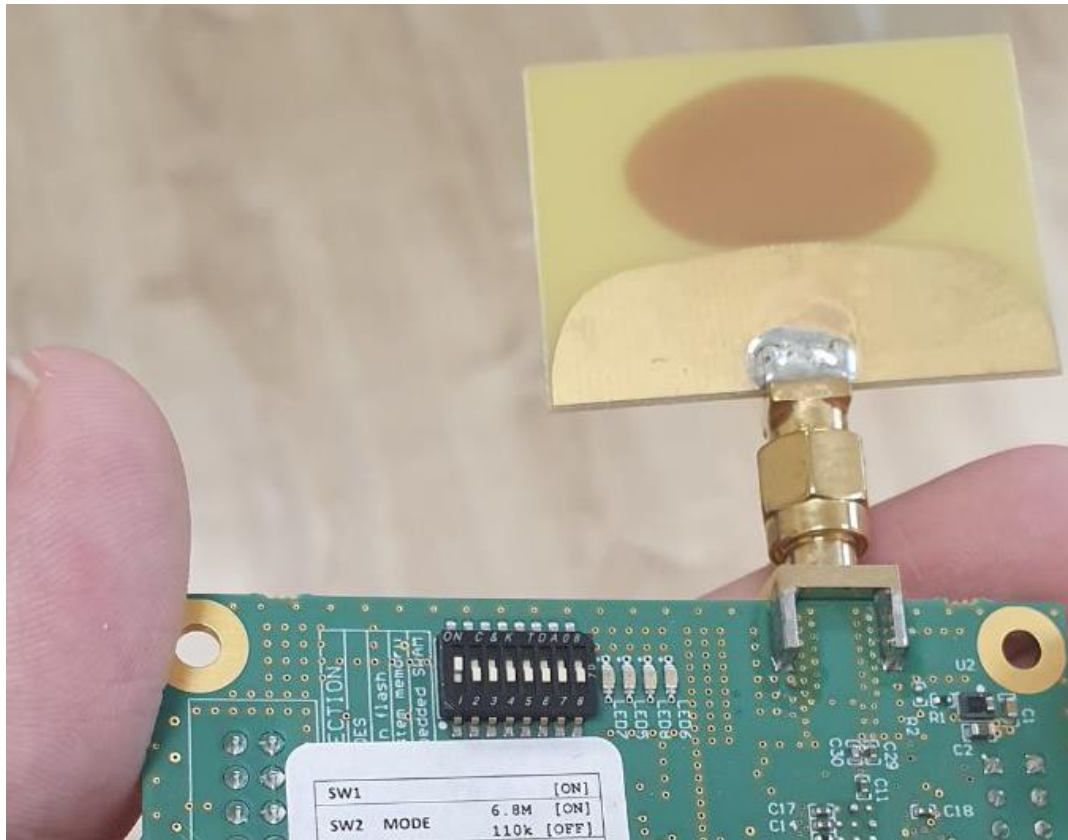


Figura 5.4: Configuración de la tag del sistema.

Para cambiar la configuración de un ancla fija o móvil únicamente es necesario modificar la posición de la patilla número cuatro, si esta patilla esta hacia arriba, como se muestra en la Figura 5.3, el ancla estará configurada como fija y si la patilla esta hacia abajo, como se muestra en la Figura 5.5, el ancla está configurada como móvil (tag). Las Figuras 5.4 y 5.6 muestran concretamente la configuración de las anclas y de la tag a partir de las Figuras 5.3 y 5.5.

5.2.2 Recolección y estudio de los datos.

Tras haber finalizado la correcta configuración de los dispositivos UWB necesarios para poner el funcionamiento del sistema de posicionamiento en interiores, se procede a recolectar los datos del recorrido seguido por un usuario.

Para que el sistema puede comenzar a detectar la posición del usuario es necesario que todas las anclas tanto las fijas como las anclas móviles de las que disponga el sistema este conectadas a una fuente de alimentación que les permita recibir corriente. Por ello, es fundamental haber configurado previamente los dispositivos tal y como se indica en la sección 5.2.2.

La recolección y almacenamiento de los datos mediante el software de código abierto DecaRange requiere que una de las anclas fijas esté conectada a la estación de trabajo en la que se está ejecutando este software. Esto es necesario debido a que de esta forma es posible utilizar el terminal para almacenar los diferentes datos recogidos en todo momento por el sistema. En la figura 5.5 se muestra de forma general las conexiones que se deben establecer entre los diferentes dispositivos que intervienen en el sistema.

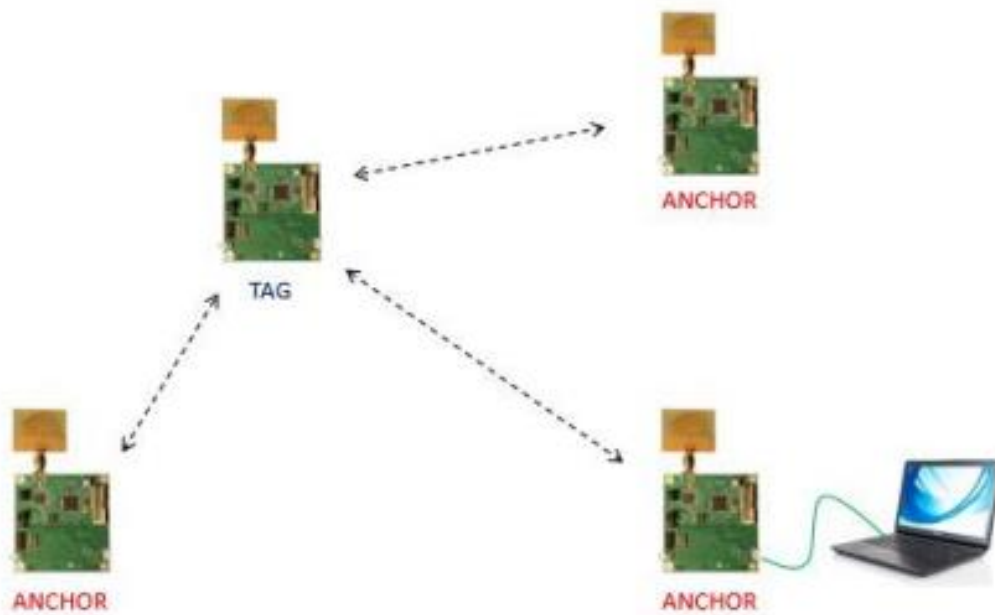


Figura 5.5: Esquema de conexiones.

Al ejecutar el software DecaRange se puede ver la siguiente ventana que se muestra en la figura 5.6. En esta ventana, en la esquina superior izquierda aparece una tabla con las anclas que está utilizando el actual sistema de posicionamiento en interiores, es necesario medir la localización en el

mundo real de estas anclas para introducirlas en esta tabla que proporciona el software tal y como se muestra en la Figura 5.6 ya que, este hará uso de esos valores para calcular el posicionamiento del usuario.

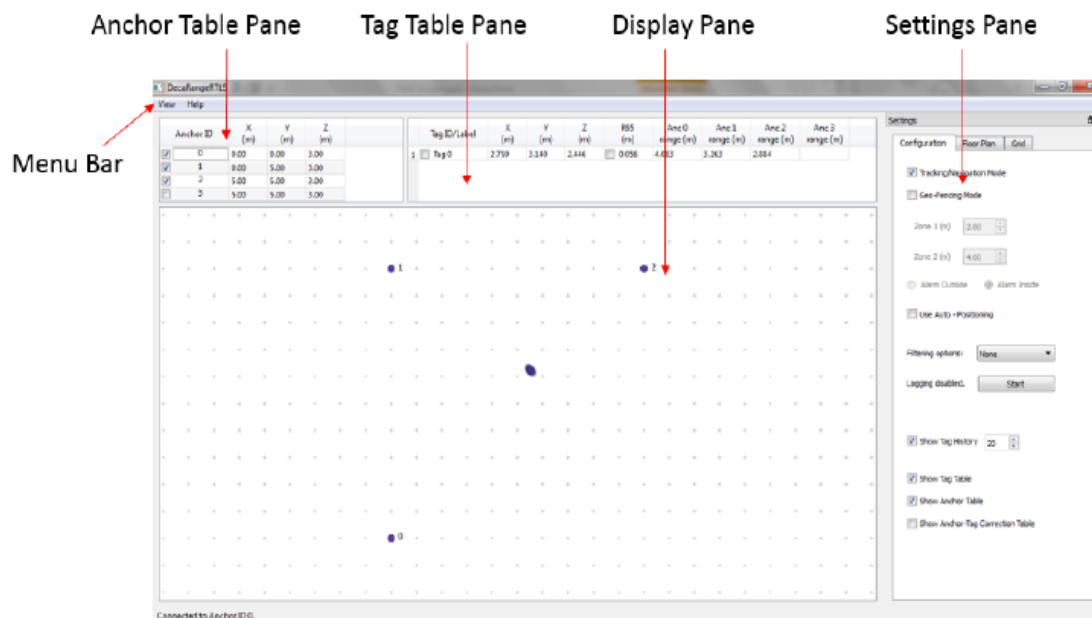


Figura 5.6: Interfaz de usuario DecaRange.

	Anchor ID	X (m)	Y (m)	Z (m)	
☒	0	0.00	0.00	3.00	
☒	1	6.00	0.00	3.00	
☒	2	0.00	4.00	3.00	
☐	3	5.00	5.00	3.00	

Figura 5.7: Panel de ajuste.

Para iniciar la grabación del log donde se almacenarán los datos del recorrido, es necesario pulsar el botón start que se muestra en la Figura 5.7 en el panel de ajustes y cuando se termine de obtener los datos necesarios para un log pulsar el botón stop que aparecerá en el mismo lugar donde se encuentra start. En el panel de ajuste Figura 5.6 se introducen las coordenadas

x, y, z correspondientes a cada una de las anclas fijas que intervienen en el sistema establecido.

Todos los logs creados por DecaRange se almacenan la ruta TREK1000\STM2.10_CooCox_PC3.6\TREK1000\DecaRangeRTLS-PC\Logs

Es recomendable estableces unas marcas en el suelo del lugar donde se vaya a realizar el estudio y medir con respecto a un punto de referencia las coordenadas de las mismas ya que, esto permitirá contrastar los datos obtenido por DecaRange con dichas marcas y saber si en un determinado tiempo el usuario se encontraba en la dicha marca tal y como no indicaba DecaRange en su log.

Una vez obtenido el log es necesario introducir este en la carpeta del programa logProcessor y ejecutar dicho programa, tras finalizar la ejecución en la misma carpeta donde se introdujo el log inicial aparecerá un nuevo archivo log cuyo nombre finalizara en “procesado” el cual es el log tras ser filtrado.

Al mismo tiempo se realiza la grabación del log es necesario que se ejecute en un terminal el servidor mqtt para la recepción de mensajes llamado LocationServer. Es posible que se deba modificar la dirección ip del broker al que se conecta este servidor debido a que la ip en la que se ejecute el broker no sea la misma que la indicada actualmente. Tras iniciar la ejecución de LocationServer se iniciará la ejecución de la aplicación, esta también necesita que se modifique la ip del broker si esta cambia, y cada vez que la persona que está realizando el recorrido pase por una marca se pulsara esa marca, cuando esta persona finalice el recorrido se detendrá el envío de mensajes pulsando la marca que se encuentra en la esquina superior izquierda de la pantalla del teléfono móvil. Es importante que el inicio de la grabación del log con DecaRange y la primera marca que se pulse en la aplicación móvil, LocationApp, se realicen lo más cercanas en el tiempo posible, lo mismo ocurre al detener la aplicación, se deben detener ambos a la vez si es posible. Esto evitará que el programa obtenga datos incorrectos.

Todas las marcas que se muestran dentro del mapa del laboratorio SmartLab indican la posición de las marcas realizadas en el suelo del laboratorio para medir con exactitud cuando el usuario pasa por una de ella, para ellos únicamente es necesario pulsar la marca cuando el usuario llega a esta. Cuando el usuario finalice todo el recorrido es necesario pulsar la marca que se encuentra en la esquina superior izquierda de la pantalla, de esta forma se creará un log con los datos reales del recorrido (ground thrut). La interfaz de usuario de esta aplicación móvil se muestra en la Figura 5.8. Esta interfaz de usuario es muy simple e intuitiva, ya que no requiere de una alta complejidad para iniciar el funcionamiento de la aplicación móvil.

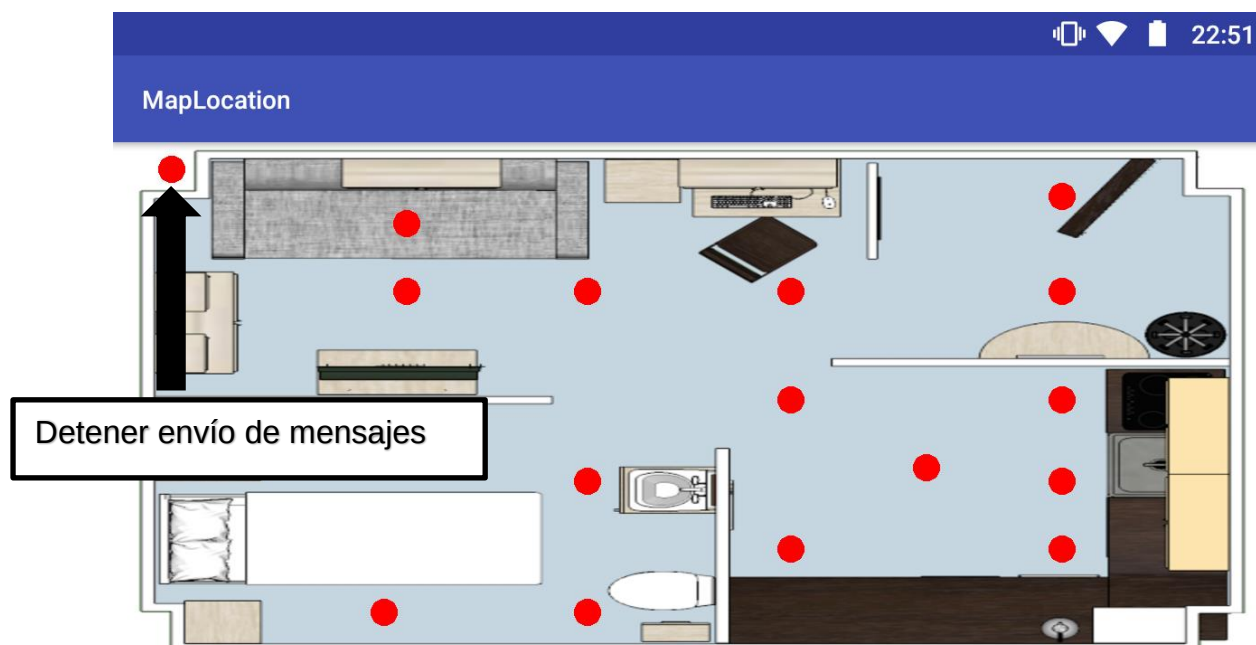


Figura 5.8: Interfaz aplicación móvil.

Tras obtener ambos archivos se puede proceder a realizar diferentes estudios con los datos de estos archivos, para intentar predecir el recorrido que realizara un usuario en un entorno determinado.

5.2.3 Monitorización del recorrido.

Para realizar la monitorización del recorrido seguido por un usuario determinado se utilizará DecaRange, pero la versión que se modificó ya que esta permite obtener en otra carpeta un log muy simple específico para este proceso con únicamente las coordenadas y el id de la tag que lleva el usuario. Este nuevo log se guardará en una carpeta denominada “TempLog” la cual se encuentra en la siguiente ruta indicada a continuación: TREK1000\STM2.25_CubeMX_PC3.8\TREK1000\DecaRangeRTLS Source Code\build-TREKdisplay-Desktop_Qt_5_2_1_MinGW_32bit-Debug\TempLog.

Se deberá iniciar el broker de mqtt y el software LogSender el cual se encargará de enviar todas las coordenadas que recoge de DecaRange para que la aplicación móvil dibuje sobre el mapa el recorrido que está siguiendo el usuario permitiendo de esta forma su monitorización. Al igual que en el punto anterior, hay que tener en cuenta que es posible que la ip del terminal donde se está ejecutando el broker no sea la misma que la ip actual a la que se conectan los softwares, por tanto, puede ser necesario modificar la ip para el correcto funcionamiento de los softwares.

Al establecer la conexión con el broker y ejecutar el software LogSender no es necesario realizar ningún proceso más ya que, el recorrido que ha seguido el usuario se ira marcando con círculos de color azul sobre el mapa del SmartLab tal y como se puede ver en la Figura 5.9.

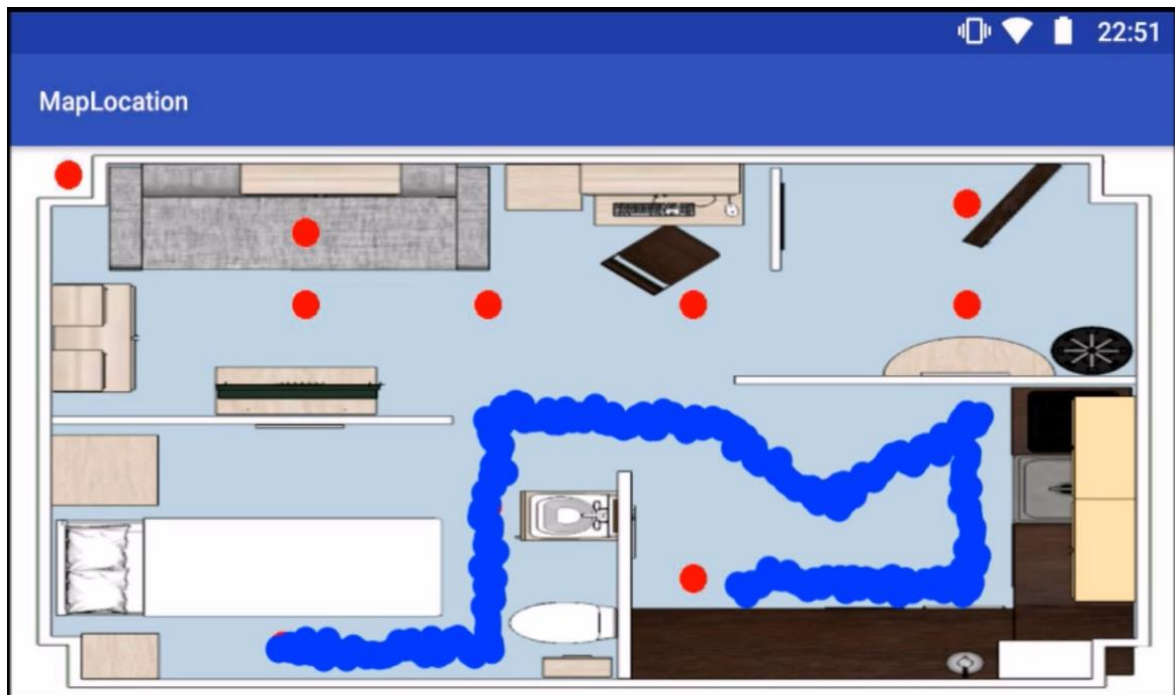


Figura 5.9: Ejemplo de monitorización del usuario.

6 Bibliografía.

- [1] G. Bejerano, P. (2016, 7 abril). El MIT convierte una señal WiFi en un GPS para interiores. Recuperado 25 julio, 2019, de <https://blogthinkbig.com/el-mit-convierte-una-senal-wifi-en-un-gps-para-interiores>
- [2] Barba, D., Moretón-Fernández, A., del Campo-Pando, H., García-Villanueva, S., March, J., & Llanos, D. R. Uso de Bluetooth de Baja Energía en Aplicaciones de Localización Indoor.
- [3] Sahinoglu, Z. (2008). *Ultra-wideband positioning systems*. Cambridge university press.
- [4] Siwiak, K. (2001, May). Ultra-wide band radio: introducing a new technology. In *IEEE VTS 53rd Vehicular Technology Conference, Spring 2001. Proceedings (Cat. No. 01CH37202)* (Vol. 2, pp. 1088-1093). IEEE.
- [5] Tyrberg, A. (s.f.). What is Indoor Positioning Systems?. Recuperado 25 julio, 2019, de <https://senion.com/indoor-positioning-system/>
- [6] Yong, S. K., Xia, P., & Valdes-Garcia, A. (2011). *60GHz Technology for Gbps WLAN and WPAN: from Theory to Practice*. John Wiley & Sons.
- [7] Alarifi, A., Al-Salman, A., Alsaleh, M., Alnafessah, A., Al-Hadhrami, S., Al-Ammar, M., & Al-Khalifa, H. (2016). Ultra wideband indoor positioning technologies: Analysis and recent advances. *Sensors*, 16(5), 707.
- [8] Evans, D. (2011). Internet de las cosas. *Cómo la próxima evolución de Internet lo cambia todo*. Cisco Internet Bussiness Solutions Group-IBSG, 11(1), 4-11.
- [9] Aplicaciones del IoT usos prácticos en el mundo real. (2019, 17 enero). Recuperado 25 julio, 2019, de <https://programarfacil.com/podcast/aplicaciones-del-iot-reales/>
- [10] IoT: protocolos de comunicación, ataques y recomendaciones. (2019, 20 mayo). Recuperado 25 julio, 2019, de <https://www.incibe-cert.es/blog/iot-protocolos-comunicacion-ataques-y-recomendaciones>
- [11] Protocolos de comunicación para IoT. (2019, 20 febrero). Recuperado 25 julio, 2019, de <https://www.luisllamas.es/protocolos-de-comunicacion-para-iot/>
- [12] Zaim, D., & Bellafkih, M. (2016, October). Bluetooth Low Energy (BLE) based geomarketing system. In *2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA)* (pp. 1-6). IEEE.

- [13] Descamps-Vila, L., Pérez Navarro, A., & Conesa Caralt, J. (2013). Integración de un sistema de posicionamiento indoor en aplicaciones SIG para dispositivo móvil.
- [14] Tapia, D. I., Cueli, J. R., García, Ó., Corchado, J. M., Bajo, J., & Saavedra, A. (2007). Identificación por radiofrecuencia: fundamentos y aplicaciones. *Proceedings de las primeras Jornadas Científicas sobre RFID. Ciudad Real, Spain*, 1-5.
- [15] TREK1000 - DW1000 Ultra-Wideband Two-Way-Ranging RTLS IC Evaluation Kit. (s.f.). Recuperado 25 julio, 2019, de <https://www.semiconductorstore.com/cart/pc/viewPrd.asp?idproduct=50557>
- [16] DWM1001-DEV - Development Board for Decawave DWM1001 Module. (s.f.). Recuperado 25 julio, 2019, de <https://www.semiconductorstore.com/cart/pc/viewPrd.asp?idproduct=70927>
- [17] Raspberry Pi 3 Modelo B+. (s.f.). Recuperado 25 julio, 2019, de https://www.amazon.es/Raspberry-Pi-3-Model-3-Modelo/dp/B07BDR5PDW/ref=asc_df_B07BDR5PDW/?tag=googshopes-21
- [18] Sommerville, I. (2005). *Ingeniería del software*. Pearson educación. 7ª. edición
- [19] Murphy, W., & Hereman, W. (1995). Determination of a position in three dimensions using trilateration and approximate distances. *Department of Mathematical and Computer Sciences, Colorado School of Mines, Golden, Colorado, MCS-95*, 7, 19.
- [20] Decawave Ltd. (2016). TREK 1000 user manual. Recuperado de https://www.decawave.com/wp-content/uploads/2018/09/trek1000_user_manual.pdf
- [21] González-Jorge, H., Roca, D., Torres, S., Armesto, J., & Puente, I. (2014). Una experiencia de Aprendizaje Basado en Proyectos en el ámbito tecnológico: Diseño de un sistema de navegación indoor de bajo coste. *Revista de Formación e Innovación Educativa Universitaria*, 7(1), 8-19.
- [22] Diaz, J. (2003). Tecnología Ultra-Wideband (UWB): la revolución a corto alcance. *Buran*, (19), 52-56.
- [23] Alvarado, M. A. C., & Delgado, J. C. S. (2017). Sistemas y tecnologías que facilitan el posicionamiento Indoor. *Pensamiento Actual*, 17(29), 132-144.

- [24]** De Angelis, A., Nilsson, J., Skog, I., Händel, P., & Carbone, P. (2010). Indoor positioning by ultrawide band radio aided inertial navigation. *Metrology and Measurement Systems*, 17(3), 447-460.
- [25]** García Pardo, C., Cano, M., Victoria, M., García-Pardo, M., María, J., Rodríguez Rodríguez, J. V., ... & Juan Llacer, L. (2010). Sistemas de comunicaciones UWB.