

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA



Trabajo Fin de Grado

**Desarrollo de una herramienta para la
localización de dispositivos en entornos
inalámbricos**

**(Development of a tool for locating devices in
wireless environments)**

Para acceder al Título de

***Graduado en
Ingeniería de Tecnologías de Telecomunicación***

Autor: Juan Carrasco Alonso

Julio-2016



E.T.S. DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACION

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

CALIFICACIÓN DEL TRABAJO FIN DE GRADO

Realizado por: Juan Carrasco Alonso

Director del TFG: Luis Sánchez González

Título: “Desarrollo de una herramienta para la localización de dispositivos en entornos inalámbricos”

Title: “Development of a tool for locating devices in wireless environments”

Presentado a examen el día: 27 de Julio de 2016

para acceder al Título de

GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Composición del Tribunal:

Presidente (Apellidos, Nombre): García Arranz, Marta

Secretario (Apellidos, Nombre): Sánchez González, Luis

Vocal (Apellidos, Nombre): García Gutiérrez, Alberto Eloy

Este Tribunal ha resuelto otorgar la calificación de:

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del TFG
(sólo si es distinto del Secretario)

Vº Bº del Subdirector

Trabajo Fin de Grado Nº
(a asignar por Secretaría)

Índice general

1. Introducción	10
1.1. Motivación	10
1.2. Objetivos	11
1.3. Resumen ejecutivo	12
2. Estado del Arte	13
2.1. Tecnologías existentes para el posicionamiento de dispositivos mediante el uso de una interfaz inalámbrica IEEE 802.11	13
2.1.1. Protocolo IEEE 802.11	13
2.1.2. Parámetros relevantes para el cálculo de la distancia	17
2.1.3. Modelos de propagación	18
2.1.4. Algoritmos existentes para posicionamiento	20
2.1.5. Localización por Tags Zigbee	24
2.1.6. Huella Digital	24
2.2. Soluciones de posicionamiento en interiores existentes en el mer- cado	25
2.2.1. Ekahau	25
2.2.2. Servicios de Localización de Google	25
2.2.3. Accuware Indoor Navigation	26
3. Diseño del sistema de posicionamiento	27
3.1. Entorno de despliegue y sensores a desplegar	27
3.1.1. Elección del entorno de despliegue	27
3.1.2. Sensores desplegados	27
3.2. Arquitectura lógica del sistema	29
4. Implementación y despliegue del sistema	31
4.1. Descripción del despliegue	31

4.1.1. Instalación	31
4.1.2. Hardware instalado	32
4.2. Implementación de la plataforma software para la captación de datos	34
4.2.1. Tecnologías utilizadas	35
4.2.2. Descripción técnica de la implementación en el servidor	38
4.2.3. Descripción técnica de la implementación en los nodos .	41
5. Validación del sistema y resultados	45
5.1. Metodología para la validación del sistema	45
5.1.1. Pruebas de campo realizadas	45
5.1.2. Metodología utilizada para la obtención de resultados .	49
5.2. Resultados obtenidos	51
5.2.1. Dificultades encontradas tras el primer análisis de los datos	51
5.2.2. Resultados de la validación	52
5.2.3. Correlación entre número de dispositivos y temperatu- ra/humedad	57
6. Conclusiones y líneas futuras	62

Índice de figuras

1.1. Evolucion del numero de dispositivos [16].	10
2.1. Línea de tiempo con las versiones más utilizadas.	14
2.2. Reparto de canales IEEE 802.11 en la banda de 2.4GHz.	14
2.3. Formato de la trama IEEE 802.11.	15
2.4. Asociación a AP IEEE 802.11.	16
2.5. Ejemplo de sistema con Angle of Arrival.	18
2.6. Ejemplo de centroide con 8 puntos de referencia.	20
2.7. Ejemplo de trilateración.	22
2.8. Ejemplo de trilateración con el sistema propuesto.	24
3.1. Vista exterior del Mercado del Este.	28
3.2. Arquitectura lógica del sistema.	30
4.1. Plano del Mercado del Este con la posición de los nodos instalados.	32
4.2. Características técnicas de la Raspberry Pi 2 Model B.	33
4.3. Características técnicas del adaptador WiFi.	34
4.4. Características técnicas del sensor de temperatura DHT-22.	35
4.5. Ejemplo de ejecución del programa airodump-ng.	36
4.6. Diagrama de flujo que muestra el funcionamiento del servidor.	39
4.7. Diagrama de flujo que muestra el funcionamiento de la captación de IEEE 802.11.	41
4.8. Línea de tiempos mostrando que insertando cada 10 segundos no hay sincronización.	42
4.9. Línea de tiempos realizando las inserciones cuando pasan las decenas de segundo. Se demuestra que hay una pequeña sincronización entre los nodos.	43
5.1. Número real frente a número estimado.	53

5.2. Dispositivos calculados a lo largo de un día.	53
5.3. Dispositivos calculados a lo largo de una semana.	54
5.4. Histograma de los errores en las medidas con centroides usando 8 nodos medidores.	54
5.5. Puntos reales frente a puntos estimados con 8 nodos usando centroides.	55
5.6. Representación de un momento dado utilizando centroides . . .	55
5.7. Histograma de los errores en las medidas con centroides usando 4 nodos medidores.	56
5.8. Puntos reales frente a puntos estimados con 4 nodos usando centroides.	57
5.9. Histograma de los errores en las medidas con trilateración usan- do 8 nodos medidores.	57
5.10. Puntos reales frente a puntos estimados usando trilateración con 8 nodos.	58
5.11. Representación de un momento dado usando trilateración. . . .	58
5.12. Histograma de los errores en las medidas con trilateración usan- do 4 nodos medidores.	59
5.13. Puntos reales frente a puntos estimados usando trilateración con 4 nodos medidores.	59
5.14. Comparación entre número de dispositivos y temperatura. . . .	60
5.15. Comparación entre número de dispositivos y humedad.	60
5.16. Medidas de temperatura y humedad a lo largo de una semana. . .	61

Índice de tablas

5.1. Medidas de posicionamiento realizadas	46
5.2. Primer conjunto de datos relativos al conteo	47
5.3. Segundo conjunto de datos relativos al conteo	47
5.4. Medidas de cadencia de Probe Request	48

Resumen

Ante el creciente aumento del uso de dispositivos móviles inalámbricos, aumentan las posibilidades de aprovechar el uso de estos dispositivos para ofrecer un servicio añadido a los usuarios. Aparece la posibilidad de, dado un recinto comercial, estimar la localización de todas las personas que se encuentren dentro del mismo, con el fin de poder ofrecerles servicios como pueden ser ofertas o recomendaciones, además de ayudar a los comercios con el posicionamiento de sus productos.

El sistema que se ha desarrollado permite realizar una estimación de las tendencias en el aforo del recinto en el que se ha desplegado la herramienta, además de poder localizar las áreas del recinto a las que va más gente y, por tanto, son más populares. Además, los comercios tendrían la posibilidad de lanzar ofertas a los clientes que pasen a su lado. Por otro lado, se han desplegado también sensores ambientales, que permitirán llevar un control de la temperatura y humedad en el recinto.

Abstract

Watching the growing increase of the use of wireless mobile devices, the possibilities to make use of these devices to offer an added service to users increase. The possibility appears to, given a commercial enclosure, estimate the position of all the people inside it, in order to offer them services such as offers or recommendations, apart from helping the shops with their product placement.

The developed system allows to make an estimation of the tendencies in the capacity of the enclosure where the tool has been deployed, apart from being able to place the areas of the enclosure where more people go, and hence are more popular. Furthermore, the shops would have the possibility to send offers to the clients who pass next to them. On the other hand, environmental sensors have also been deployed, allowing to control the temperature and humidity in the enclosure.

Agradecimientos

En primer lugar, quiero agradecer la ayuda de Juan Ramón Santana, sin el cual este proyecto no habría podido llevarse a cabo. Muchas gracias, me has ayudado mucho.

También quiero agradecer la ayuda a Luis, que en los últimos días antes de terminar ha trabajado mucho para poder acabar a tiempo.

Muchas gracias también a familia, amigos y compañeros, que aunque a veces discutamos, al final sois lo importante que queda. Princesitas, muchas gracias, aguantáis más de lo necesario.

Por último agradecer a profesores y al Grupo de Ingeniería de Telemática, que me habéis ayudado siempre que me ha hecho falta.

Acrónimos

- AP: Access Point
- DiY: Do It Yourself
- GPIO: General Purpose Input/Output
- IEEE: Institute of Electrical and Electronics Engineers
- ITU: International Telecommunications Union
- MAC: Medium Access Control
- SD: Secure Digital
- TCP: Transport Control Protocol
- USB: Universal Serial Bus
- WiFi: Wireless Fidelity
- ZMQ: ZeroMQ

1.1. Motivación

Es evidente que en los últimos años el número de dispositivos móviles ha aumentado de forma exponencial. El conjunto de teléfonos móviles, ordenadores portátiles, pulseras inteligentes y demás dispositivos crean una red personal para cada usuario que, por ejemplo, teniendo cada dispositivo unas características concretas, podría ser usada para caracterizar al usuario de manera casi unívoca. Este concepto se conoce como Huella Digital [14] y puede ser útil para solucionar problemas de seguridad como el robo de contraseñas. A la vista de este ejemplo y de otros muchos análogos, queda claro que al mismo tiempo que los dispositivos digitales han pasado a formar parte de nuestra vida, los servicios que éstos ofrecen a los usuarios son capaces de mejorar su vida o cuanto menos, facilitársela.

Evolución de los números de smartphones vendidos en el mundo (en unidades)

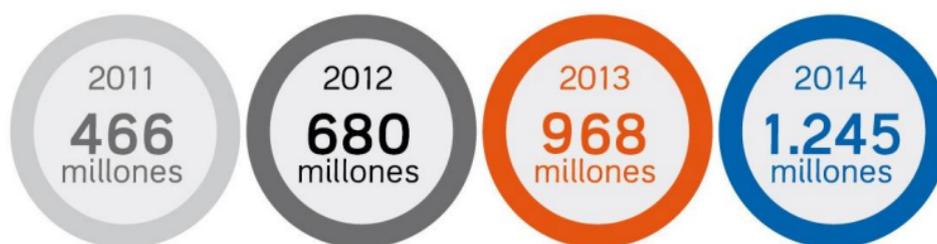


Figura 1.1: Evolucion del numero de dispositivos [16].

Por otro lado, otra tecnología que también ha proliferado en paralelo es la de las redes locales inalámbricas basadas en WiFi. Estas redes han alimentado y a la vez facilitado la provisión de los nuevos servicios que se ofrecen para nuestros dispositivos, principalmente vídeos en alta resolución o streaming de música. Estos nuevos modelos de servicio hacen que muchas veces no sea suficiente con la conexión de datos que el teléfono del usuario tiene a través

de las redes celulares de banda ancha y tenga que conectarse a una red WiFi, ya sea la de su casa, de un comercio o alguna de las múltiples que se han desplegado en espacios públicos. Para la mayoría de usuarios es habitual dejar la conexión WiFi activa incluso cuando no se está conectado a ninguna red, con el fin de evitar tener que operar con el teléfono más de lo necesario. Además, las mejoras de rendimiento que ha sufrido este tipo de conexión a lo largo de los últimos años hacen posible que el dejar la conexión activa no sea un problema para el consumo de batería del teléfono.

La motivación principal para el desarrollo del proyecto es el hecho de que no existen soluciones de posicionamiento en interiores tan populares como por ejemplo lo es GPS en entornos de exteriores. Por este motivo, se ha acordado presentar una herramienta que permita la localización en entornos de dichas características, que presente un funcionamiento correcto. Además, esta herramienta puede ser utilizada por experimentadores, permitiéndoles desarrollar sus propias aplicaciones para el sistema.

Otra de las motivaciones para este proyecto es su posibilidad de aportar servicios a los locales comerciales que se encuentran dentro del área de cobertura del sistema. Para éstos, o incluso los gerentes del espacio, es muy útil conocer los lugares de concentración de gente, o los lugares donde nunca pasa nadie. Con esta información, es posible tomar medidas en términos de colocación de productos, renovación de inventario o incluso en toma de medidas de seguridad, como puede ser la colocación de extintores o salidas de emergencia. Esto, además, es otra manera de ofrecer un servicio al usuario. Si un comerciante renueva las zonas de su local para cambiar los productos de sitio, es posible que la experiencia del usuario al comprar allí se vea mejorada.

1.2. Objetivos

El objetivo principal de este proyecto es la implementación y despliegue de un sistema de posicionamiento en interiores basado en tecnologías inalámbricas. Para el desarrollo de esta implementación se han impuesto una serie de objetivos a cumplir, que, unidos, nos permitirán tener un sistema completo de localización.

El primer objetivo que se ha planteado es el desarrollo de una herramienta que permita obtener la información que posteriormente se utilizará para la consecución de los siguientes objetivos. Esta parte consta de una planificación hardware, consistente en la selección de los componentes instalados, y una implementación software, que consta de la implementación de los programas que se utilizarán para el funcionamiento del sistema, que se detallarán más adelante.

Usando la herramienta anterior, el siguiente objetivo es el de implementar

un sistema de Conteo de Personas y Dispositivos. El resultado que se persigue es, teniendo las medidas de potencia de señal y detección de los nodos receptores, poder establecer una relación entre el número de dispositivos detectados y la cantidad de personas presentes en un lugar para así poder realizar una estimación del aforo en dicho recinto.

Una vez modelado el aforo de personas, se implementan dos algoritmos para realizar una estimación de la posición física que ocupa cada dispositivo dentro del recinto, haciendo así posible la realización de un seguimiento de las distintas zonas para poder llevar un control de las zonas más visitadas o los comercios más populares.

Por último, se realizará un análisis de la correlación entre las medidas anteriormente citadas con las medidas proporcionadas por otros tipos de sensores, como pueden ser aquellos de temperatura o dosímetros de campo electromagnético.

1.3. Resumen ejecutivo

Esta memoria se ha dividido en seis capítulos. Tras este capítulo en el que se introducen los puntos básicos del proyecto, se describe en el segundo capítulo el estado actual de las tecnologías utilizadas y se han explicado las bases tecnológicas fundamentales para la realización de la herramienta desarrollada. En el capítulo 3 se comentan las elecciones realizadas en cuanto a lugar de despliegue y la composición de los nodos de medida.

En el cuarto capítulo se presentan los aspectos técnicos de la realización del desarrollo, tanto en el lado hardware como el software. Se explicará en detalle la plataforma software creada y cómo está relacionada con el hardware elegido.

En el capítulo 5 se recogerán las distintas pruebas realizadas en el campo, así como los resultados arrojados por la herramienta y se compararán con los datos reales medidos.

Por último, en el sexto capítulo se presentarán las conclusiones sacadas tras la consecución del proyecto, así como una serie de líneas futuras que se proponen para la continuidad del proyecto.

2.1. Tecnologías existentes para el posicionamiento de dispositivos mediante el uso de una interfaz inalámbrica IEEE 802.11

Para el desarrollo del proyecto se han tenido en cuenta algunas soluciones ya existentes, que están posicionadas actualmente en el mercado pero que, por una razón o por otra no son del todo válidas para el servicio que se quiere ofrecer. También se han detallado las herramientas matemáticas que se van a utilizar para los cálculos realizados.

2.1.1. Protocolo IEEE 802.11

IEEE 802.11 [20], conocido comercialmente como WiFi, es un estándar que define el uso de las dos capas inferiores de la pila de protocolos OSI (Física y Enlace) con el fin de utilizarlas en una Red de Área Local Inalámbrica en las bandas de frecuencias de 2.4, 5 y 60 GHz. Se definen los conceptos básicos para el funcionamiento de estas redes, como por ejemplo las Estaciones Base, el formato de la trama o los requerimientos de potencia. A lo largo de los años ha ido evolucionando el estándar y se han añadido funcionalidades. Cada vez que aparece una nueva versión, que aporta características nuevas con respecto a las anteriores, se publica con el formato IEEE 802.11 más una letra que la distingue.

Lineas generales

La comunicación basada en IEEE 802.11 se realiza con dos tipos de equipos: el AP inalámbrico (que se ocupa de procesar las señales que le llegan, ya sea por su interfaz inalámbrica, o bien por su interfaz cableada que le conecta a otras redes, Internet en general) y el dispositivo móvil que quiere acceder a la red. El dispositivo envía tramas IEEE 802.11 con la información que desea

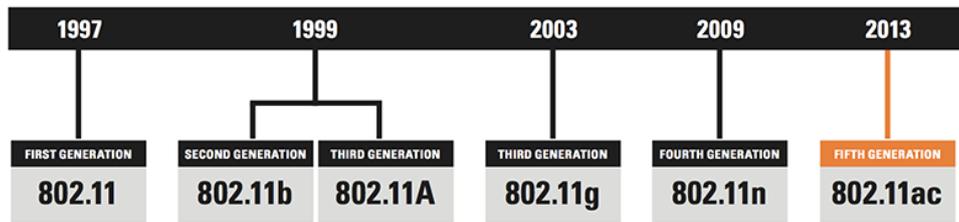


Figura 2.1: Línea de tiempo con las versiones más utilizadas.

intercambiar al AP, este las transporta hacia su otra interfaz (por ejemplo Ethernet) y las reenvía por su interfaz cableada. El proceso inverso es igualmente posible, es decir, un paquete proveniente de Internet tiene al dispositivo como destinatario. En este caso el AP insertará la información en una trama IEEE 802.11 con dirección destino la del dispositivo. Existe también otro modo de funcionamiento ad-hoc, en el que dos dispositivos se conectan directamente entre ellos, sin necesidad de dispositivos intermedios, pero este modo no es relevante para el proyecto y por lo tanto no se abordará en esta sección.

El funcionamiento más común de IEEE 802.11 actualmente se localiza en la banda de frecuencias de 2.4GHz. Cuando se trabaja en esta banda, se divide en 14 canales de 22MHz separados 5MHz entre si. Por lo tanto, existe una clara interferencia entre canales, por lo que se recomienda que solamente se utilicen los canales no interferentes 1, 6 y 11. El funcionamiento en la banda de 5GHz es algo más complicado de explicar, ya que es diferente en función de la legislación de cada país o región.

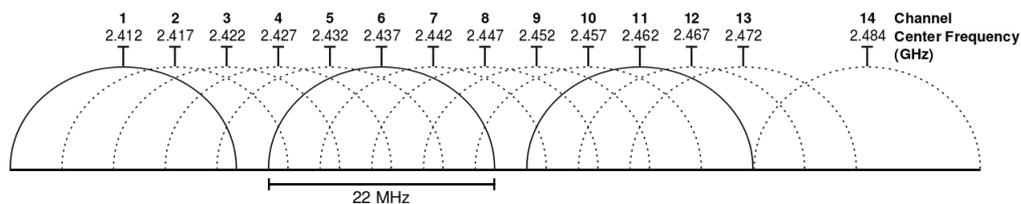


Figura 2.2: Reparto de canales IEEE 802.11 en la banda de 2.4GHz.

Formato de la trama IEEE 802.11

Los datos de las capas superiores se encapsulan en una trama IEEE 802.11. Los campos más relevantes que aparecen en la cabecera de estas tramas son:

- Type/Subtype: Especifica el tipo (Control, Administración o Datos) y subtipo de la trama.
- Address: A diferencia de la trama Ethernet, en el que solo hay direcciones de origen y destino, aparecen 4 direcciones. Esto es porque cabe la

posibilidad de realizar forwarding desde, hacia o a través del Sistema de Distribución.

- Sequence Control: Para manejar correctamente retransmisiones.
- Power Management: Indica si las funciones de control de potencia están o no activas.

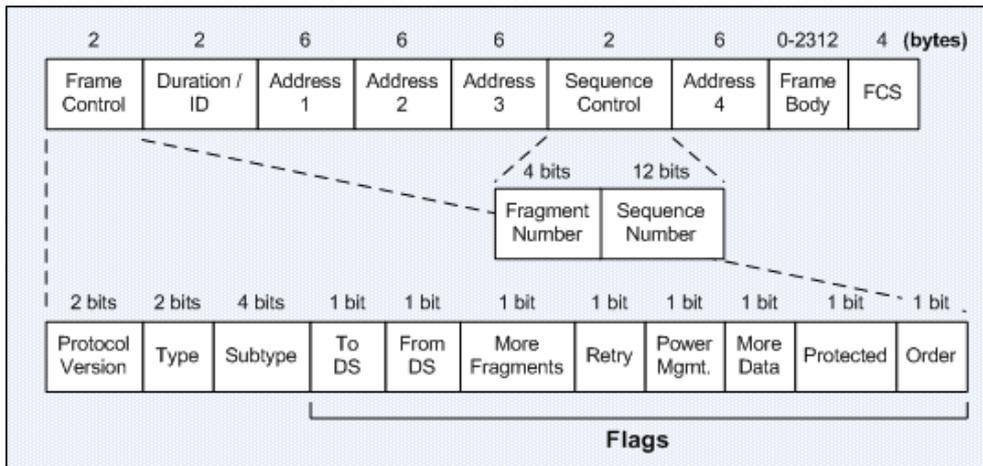


Figura 2.3: Formato de la trama IEEE 802.11.

Gracias al campo de tipo y subtipo de la trama, es posible diferenciar la funcionalidad de cada paquete y, en ocasiones, incluso la procedencia del mismo. Esto, como se verá más adelante, será útil a la hora de filtrar los paquetes que llegan a un dispositivo IEEE 802.11 y dará la posibilidad de procesar los distintos tipos de paquetes de la manera que corresponda en cada caso.

Además de esta estructura de cabecera, es importante destacar un apartado más que será necesario para la consecución del proyecto: la cabecera Radiotap. Esta cabecera, que no se transmite, es añadida por el driver de la tarjeta de red y aporta información de control interesante sobre el medio radio, como por ejemplo el ruido que introduce la antena, la potencia recibida por la tarjeta o la velocidad binaria a la que se transmiten los datos.

Paquetes de tipo Probe Request

Existen dos maneras de realizar la búsqueda de Puntos de Acceso IEEE 802.11. La primera es la búsqueda pasiva, en la que un dispositivo espera a escuchar los mensajes de tipo Beacon que envían periódicamente los AP. Esta forma es energéticamente costosa, ya que obliga al dispositivo móvil a estar escuchando el canal todo el tiempo para encontrar los paquetes necesarios. La otra manera es la búsqueda activa, en la que el dispositivo envía un mensaje al canal y espera ser respondido por los AP.

Los paquetes de tipo 0 y subtipo 4 se denominan Probe Request. Estos paquetes son los que se utilizan en la búsqueda activa. El funcionamiento de la búsqueda activa es el siguiente:

1. El dispositivo que busca conectarse a un AP envía este paquete a toda la red (dirección Broadcast). En este paquete se incluye una lista de los Puntos de AP a los que desea conectarse, por ejemplo, los que se ha conectado previamente y conoce.
2. Los AP que cumplan las características para que el dispositivo se conecte a ellos le contestan con un mensaje de Probe Response.
3. El dispositivo escoge un AP de la lista de los que le responden y le envía una petición de asociación, con su correspondiente sistema de autenticación si lo hubiera, por ejemplo, IEEE 802.1X.

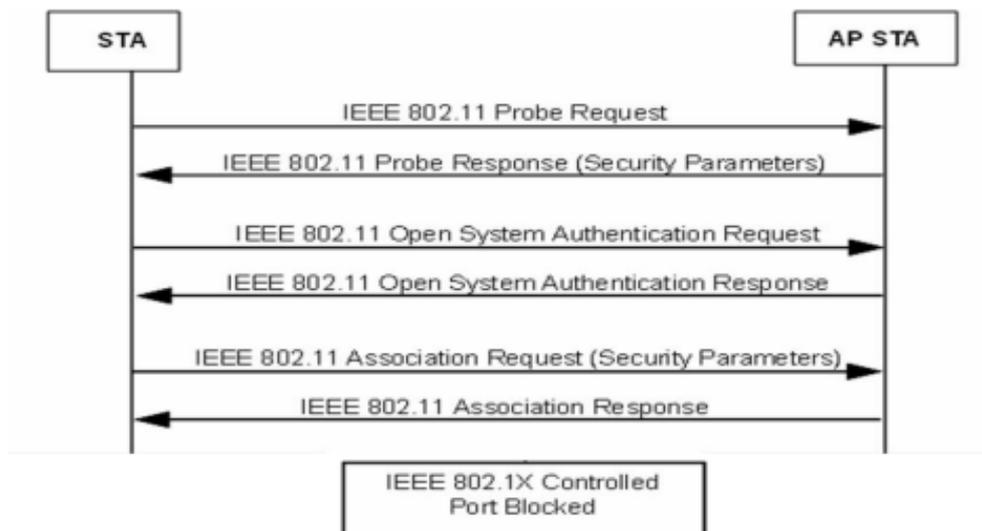


Figura 2.4: Asociación a AP IEEE 802.11.

En el marco de este trabajo, este tipo de paquetes es relevante dado que es el único formato de paquete que cualquier dispositivo va a enviar independientemente de su estado, es decir, donde no necesariamente estén conectados a una red IEEE 802.11. Además de esto, usar este tipo de paquete nos asegura que el dispositivo que estamos detectando no será un AP, ya que éstos no los envían. Por lo tanto, lo que se aprovechará durante este proyecto será, en principio, detectar este tipo de paquetes en la red y extraer de ellos la información que nos permita localizar al dispositivo móvil que lo ha enviado.

Modos de monitorización de la red IEEE 802.11

Las interfaces de acceso WiFi, más concretamente sus drivers, ofrecen dos maneras de escuchar el canal radio inalámbrico en busca de paquetes IEEE 802.11. La primera es el denominado modo Promiscuo de Datos, en el que un dispositivo se conecta a una red y captura todos los paquetes que circulan por ella, incluso aquellos cuya dirección de destino no es la suya. El otro modo es el modo Monitor, en el que un dispositivo captura todos los paquetes que aparecen en el entorno IEEE 802.11. Los dos modos son muy similares, pero la mayor diferencia radica en el hecho de que en el modo Promiscuo es necesario estar previamente conectado a la red. Por ello, no es apropiado si lo que se quiere es tener una visión de todos los dispositivos que aparecen en la red.

2.1.2. Parámetros relevantes para el cálculo de la distancia

En el ámbito de la localización, existen diversas maneras de calcular la distancia entre dos dispositivos en función de los parámetros que presentan sus transmisiones inalámbricas. A continuación se presentan varias de estas opciones:

- Time of Arrival [15]. En este modo de posicionamiento se tiene en cuenta el tiempo que ha tardado un paquete en atravesar una distancia. Conocido ese tiempo y la velocidad a la que se transmite la onda (dependiente de la frecuencia de emisión), es posible estimar la posición del emisor de la onda. Por lo tanto, en el caso de tener varias antenas receptoras situadas en distintos puntos, sería posible estimar con una precisión bastante alta la posición real del dispositivo emisor. Para estimar una posición utilizando este método es necesario que las antenas receptoras y el dispositivo emisor tengan una sincronización exacta, ya que una diferencia de varios milisegundos podría corromper la medida. Además de esto, es necesario tener control absoluto de ambos tipos de dispositivos, ya que es necesario forzar el envío de tramas desde el dispositivo emisor, conocer el momento de emisión y conocer el momento en el que la onda llega a la estación receptora. Por este motivo, este método no es válido para su uso en este proyecto.
- Angle of Arrival [24]. En este caso, se tiene en cuenta el ángulo de llegada de una onda a la estación receptora. Si se conoce el ángulo con el que llega una onda a varias antenas, es posible estimar la intersección en un punto, que tomaremos como posición estimada del dispositivo. Para utilizar este método, es necesario que en cada estación receptora existan varias antenas direccionales que se ocupen de estimar el ángulo de llegada de las ondas, ya que con una sola antena omnidireccional es imposible

estimar dicho valor. Por lo tanto, este método tampoco es válido para este proyecto.

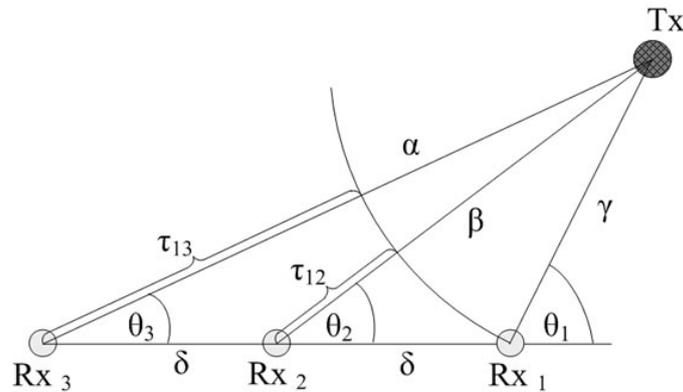


Figura 2.5: Ejemplo de sistema con Angle of Arrival.

- **Potencia recibida.** Este método es el más usado, debido a su "simpleza". En base a la potencia recibida por una estación receptora y la potencia transmitida por el emisor, es posible calcular la atenuación sufrida por la onda en la distancia que ha recorrido. A partir de dicha atenuación es posible estimar, basándonos en un modelo de propagación, la distancia del emisor al receptor. Aunque pueda parecer sencillo, es necesario usar aspectos que complican su uso y que afectan a su precisión ya que encontrar el modelo de canal adecuado es laborioso y el medio en el que se encuentran los dispositivos puede variar sin previo aviso, lo que daría lugar a errores en el cálculo. En este proyecto se utilizará este método para realizar el posicionamiento de los dispositivos.

2.1.3. Modelos de propagación

Cuando se transmite una onda, esta se ve afectada por fenómenos como la atenuación, lo que la hace perder potencia hasta que desaparece. Esta atenuación, además de otros factores, depende de la distancia que recorre la onda. Por lo tanto, sería posible estimar la potencia que se pierde en una transmisión conociendo la distancia entre dos puntos y los parámetros electromagnéticos del medio. O, a la inversa, conociendo la atenuación que se ha producido entre dos puntos, se podría estimar la distancia entre esos dos puntos. La dificultad en la elaboración de estos modelos reside en que es necesario conocer todas las variables presentes en el medio (paredes, obstáculos, interferencias) y que, si se trata de un medio dinámico, las condiciones del entorno cambiarían constantemente.

A continuación se presentan varios modelos de propagación que hemos estudiado durante la realización de este proyecto.

Modelo de Friis

La ecuación de Friis [23] proporciona la potencia recibida por una antena bajo condiciones ideales, dada otra antena a una distancia conocida, transmitiendo una potencia fija.

$$\frac{P_r}{P_t} = G_r * G_t * \left(\frac{\lambda}{4 * \pi * R}\right)^2$$

Donde:

- P_r, P_t : Potencia recibida por la antena receptora y potencia transmitida por la antena transmisora.
- G_r, G_t : Ganancias de las antenas receptora y transmisora.
- λ : Longitud de onda de la transmisión. En IEEE 802.11 (2.4GHz), 0.125 metros.
- R : Distancia entre las dos antenas.

Esta ecuación puede ser también expresada en escala logarítmica, para simplificar los cálculos:

$$P_r = P_t + G_r + G_t + 20 * \log\left(\frac{\lambda}{4 * \pi * R}\right)$$

Considerando que las ganancias de las dos antenas son nulas ($G_t = G_r = 1$), operando con la ecuación, se puede despejar la distancia entre las antenas como:

$$R = \frac{\lambda}{4 * \pi * 10^{\frac{P_r - P_t}{20}}}$$

Esta ecuación es solamente válida para entornos en línea de vista y por lo tanto no es válida para este sistema. Sin embargo, es válido como una primera aproximación.

Modelo de propagación de la ITU para propagación en interiores

Este modelo [21] estima la atenuación sufrida por una onda transmitida dentro de una habitación o un recinto cerrado delimitado por paredes. Este modelo, al contrario que el anterior, es capaz de tener en cuenta factores como paredes o distintos pisos para el cálculo de la atenuación.

$$L = 20 * \log f + N * \log d + Pf(n) - 28$$

Donde:

- L : Atenuación sufrida por la señal a lo largo del camino.

- f: Frecuencia de transmisión, en Megahercios
- d. Distancia recorrida por la señal.
- N: Coeficiente de pérdida entre transmisor y receptor.
- Pf(n): Factor de penetración del suelo.

N y Pf son parámetros dependientes, de la frecuencia de transmisión y del medio en el que se encuentren las antenas. Es posible operar con la ecuación, colocando la frecuencia como un valor conocido (2412 Megahercios, frecuencia central de una transmisión IEEE 802.11). Además, se va a estimar que la pérdida del suelo es 0, ya que en nuestro caso solamente hay un piso y, como valor inicial, se probará N=20.

$$d = 10^{\frac{L-39,65}{20}}$$

Puede observarse que al elegir N=20, la pérdida de potencia pasa a tener una dependencia cuadrática con la distancia recorrida, lo que coincide con el modelo de Friis visto anteriormente.

2.1.4. Algoritmos existentes para posicionamiento

Cálculo por centroides

El centroide o centro de simetría de una figura geométrica es el punto medio de todos los puntos de la figura. Es el punto sobre el que una hipotética figura de masa uniforme podría aguantarse sobre una chincheta.

El funcionamiento de este algoritmo se basa en calcular el punto medio de una serie de puntos de referencia, que estarán en unas posiciones conocidas. Es posible estimar la posición de dicho punto central si se considera que la distancia de dicho punto a todos los puntos de referencia es la misma. El problema que se presenta es que en el caso real presentado a lo largo del proyecto, esta distancia no es la misma y, por lo tanto, se introduce un error muy grande en los cálculos.

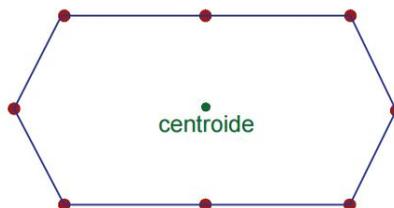


Figura 2.6: Ejemplo de centroide con 8 puntos de referencia.

La solución que se ha tomado es ponderar la posición de cada nodo en función de la potencia recibida por el dispositivo, virtualmente alejando y acercando los nodos al punto de referencia. Con esto se consigue que la medición sea mucho más exacta y la posición estimada pueda ser confiable. Para realizar la ponderación se han tenido en cuenta los modelos anteriormente introducidos, considerando que la relación entre la pérdida de potencia y la distancia es cuadrática. Por lo tanto, se ponderará la posición de cada nodo con la raíz cuadrada de la potencia que hemos recibido. Es importante ver que al calcular la posición de esta manera se evita el uso de valores que desconocidos y que sólo pueden ser estimados, como la potencia que transmite cada dispositivo, ya que las operaciones que se realizan son relativas entre los distintos nodos. Es, por lo tanto, una solución precisa para el problema de la localización.

Para realizar estos cálculos, se realiza un bucle en el que se van haciendo las sumas ponderadas relativas a la posición en el eje X y el eje Y. También se van sumando las distintas ponderaciones en un contador de manera que al final pueda deshacerse esa ponderación. Evidentemente, cuantos más puntos se tengan al realizar la operación, mucho más precisa será la medida. Se presenta a continuación el pseudo código que permite realizar los cálculos mencionados:

```
x = 0;
y = 0;
count = 0;
for node in nodelist:
    if power(node) !=0:
        x = x + x_node * sqrt(power(node));
        y = y + y_node * sqrt(power(node));
        count = count + sqrt(power(node));
x = x/count;
y = y/count;
```

El mayor problema en el uso de este sistema es el modelo de propagación, que no es del todo preciso ya que se ha escogido uno previamente calculado. Para solucionar esto, se ha propuesto el cálculo de índices de propagación distintos para cada pareja de nodos. Después, en el momento de la ejecución, se considerará la potencia recibida por cada nodo en el factor que corresponda al nodo más cercano al dispositivo en cuestión. De esta manera, se asume que las condiciones del entorno del dispositivo son las mismas que las del nodo más cercano y por lo tanto, se pueden aproximar los parámetros de atenuación por los del mismo. Este proceso se extenderá más adelante en las líneas futuras del proyecto.

Trilateración

La trilateración [22] es un modelo matemático utilizado para determinar las posiciones relativas de objetos usando la geometría de triángulos. Usando las localizaciones de al menos 3 puntos de referencia y la distancia entre esos puntos al punto requerido, es posible estimar la posición del punto que se busca. En un sistema de trilateración común, es posible considerar en cada uno de los puntos de referencia una esfera con radio la distancia al punto buscado. Por lo tanto, se obtienen 3 esferas que, en un caso ideal, intersectan en un punto, que será la posición que se está calculando. En un sistema de trilateración tradicional, las ecuaciones buscadas son:

$$(a_1 - x)^2 + (b_1 - y)^2 + (c_1 - z)^2 = R_1^2$$

$$(a_2 - x)^2 + (b_2 - y)^2 + (c_2 - z)^2 = R_2^2$$

$$(a_3 - x)^2 + (b_3 - y)^2 + (c_3 - z)^2 = R_3^2$$

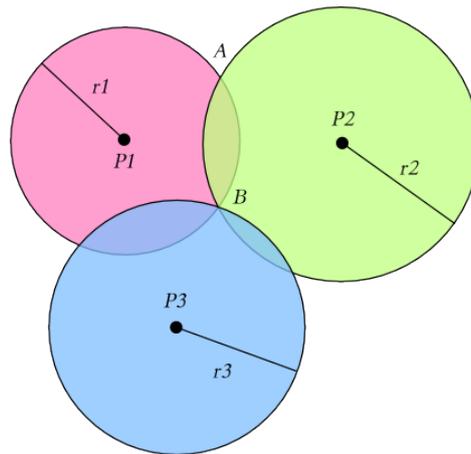


Figura 2.7: Ejemplo de trilateración.

Donde:

- a_1, a_2, a_3 : Posición en el eje X de los puntos de referencia 1,2,3.
- b_1, b_2, b_3 : Posición en el eje Y de los puntos de referencia 1,2,3.
- c_1, c_2, c_3 : Posición en el eje Z de los puntos de referencia 1,2,3.
- R_1, R_2, R_3 : Distancia entre el punto de referencia 1,2,3 y el punto a localizar (x, y, z)

Por lo tanto, la solución del problema se reduce a encontrar la solución de ese sistema de ecuaciones de segundo grado.

Se ha mencionado que para el cálculo de la posición buscada se parte de las distancias entre el punto buscado y los puntos de referencia. En el ámbito de este proyecto no se dispone del dato de esa distancia, pues para calcularla es necesario conocer la potencia de transmisión del dispositivo que emite el paquete. Por lo tanto, una aproximación a la trilateración tradicional es imposible. Para saltar esta restricción se ha pensado en la posibilidad de trabajar con valores relativos entre cada nodo, es decir, usar las relaciones que hay entre las distancias nodo-dispositivo de cada uno de los medidores. En lugar de trabajar con los valores absolutos de las distancias, se trabajará con las distancias en función de una de las distancias de algún nodo, que se usará como referencia, añadiendo una nueva variable al sistema de ecuaciones. Además, se ha decidido prescindir de los cálculos del eje Z para simplificar el funcionamiento, quedando el siguiente sistema de 3 ecuaciones y 3 incógnitas:

$$\begin{aligned}(a1 - x)^2 + (b1 - y)^2 &= R1^2 \\(a2 - x)^2 + (b2 - y)^2 &= (\alpha * R1)^2 \\(a3 - x)^2 + (b3 - y)^2 &= (\beta * R1)^2\end{aligned}$$

Donde :

$$\alpha = \sqrt{\frac{P1}{P2}}, \beta = \sqrt{\frac{P1}{P3}}$$

Esto es debido a que:

$$P = \frac{C}{d^2}$$

Donde C es una constante que depende de las características de la transmisión. Evidentemente, en la captura de un mismo paquete, las características serán las mismas para todos los nodos receptores. Y por lo tanto:

$$\frac{P1}{P2} = \frac{C * d2^2}{C * d1^2}$$

$$\alpha = \frac{d2}{d1} = \sqrt{\frac{P1}{P2}}$$

Es decir, se sigue otra vez el criterio de que la pérdida de potencia aumenta de una manera cuadrática con la distancia. Una vez conocidas estas ecuaciones, la resolución se reduce a calcular la solución del sistema.

Este sistema tiene 3 ecuaciones con 3 incógnitas y, por lo tanto, solamente se puede calcular la posición del dispositivo en función de 3 nodos de medida. Si se incluye una nueva ecuación, el sistema pasaría a estar indeterminado y no tendría solución. Se encuentra entonces un problema, ya que al recurrir

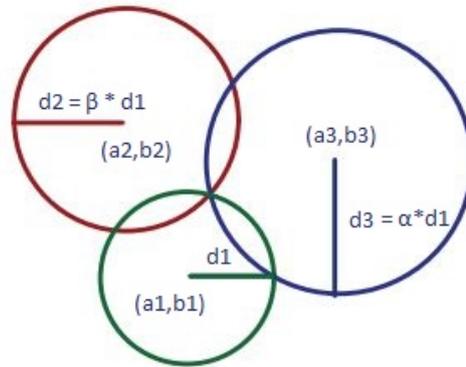


Figura 2.8: Ejemplo de trilateración con el sistema propuesto.

solamente a 3 de los nodos, se está perdiendo información. En un principio se decidió realizar el cálculo para todas las combinaciones posibles de ecuaciones, para posteriormente realizar una media aritmética de todas las posiciones calculadas. Esta solución se descartó rápidamente, ya que al usar las combinaciones de ecuaciones relativas a nodos alejados de la posición real, la posición relativa que se calculaba era muy alejada de la real y por tanto se adulteraba el resultado. La solución final que se ha propuesto es la utilización de los 3 nodos que más cerca de la posición a determinar estén, entendiendo por cercanía que reciban una mayor potencia. De esta manera, se calcula siempre la posición en base a las 3 más cercanas, lo que se espera aporte una solución más fiable.

2.1.5. Localización por Tags Zigbee

Una de las ideas existentes [18] en el tema de la localización consiste en entregar a cada usuario u objeto que queremos localizar una etiqueta cuya funcionalidad es la de emitir datos de manera inalámbrica para que otros nodos medidores puedan medir esas transmisiones con el fin de localizar esa etiqueta en el espacio. Normalmente se suelen utilizar etiquetas con tecnología ZigBee, debido a su bajo consumo de energía, lo que permite una alta autonomía de dichas etiquetas para que no sea necesario preocuparse por la carga de batería con frecuencia.

2.1.6. Huella Digital

Este concepto, trata de que cada usuario, en base a sus dispositivos, configuraciones, etc, es único. Por lo tanto, podría no ser necesario el uso de cuentas de usuario o contraseñas para que un usuario se autentique, de este modo se evitarían los posibles robos de identidad y la autenticación se podría hacer en

base a los parámetros únicos de cada usuario.

El uso de este tipo de medidas trae consigo ventajas y desventajas. Por un lado, al disponer de datos que identifiquen al usuario unívocamente, desaparecen preocupaciones del tipo olvido de contraseñas, phishing, etc. Sería posible que, simplemente con entrar desde su ordenador se ofrezca ya una capa de autenticación básica. Por otro lado, el uso de esta información se podría volver fraudulento, ya que un intruso podría falsificar su información de usuario y hacerse pasar por otros para acceder a su información. Además de eso, sería posible localizar a un individuo gracias a la información que aportan factores como la dirección IP desde la que se conecta, los sitios que visita, etc.

2.2. Soluciones de posicionamiento en interiores existentes en el mercado

2.2.1. Ekahau

Ekahau [4] es una empresa dedicada a la planificación de redes. Sus productos incluyen herramientas para el diseño de redes, para su mantenimiento, o análisis de espectro. Una de estas herramientas es Ekahau Vision, un servicio que permite a una empresa conocer en todo momento dónde se encuentran en todo momento sus activos y trabajadores, con el fin de poder aumentar la productividad al máximo y poder automatizar tareas o mantener un inventario estricto.

Esta solución propone el uso de Tags que porta cada activo o usuario y que se encargan de transmitir la información requerida a las estaciones base que posteriormente proceden a calcular las posiciones de los tags. Esta información puede ser representada en un plano, mapa de calor, o puede ser usada para operar con ella y ofrecer distintos servicios.

El funcionamiento de esta tecnología se basa en Fingerprinting con RFID sobre WiFi. Se usa la potencia de las señales recibidas para calcular un número de posibles localizaciones en un mapa virtual y comparar las medidas de los tags con las medidas previamente guardadas en una base de datos. Esto se basa en el principio de que dos "huellas" no pueden ser iguales, es decir, cada localización tiene su única huella. El software asigna niveles de confianza a cada localización en base a la frecuencia de los datos, la calidad, el historial del dispositivo y un análisis predictivo.

2.2.2. Servicios de Localización de Google

En dispositivos Android, Google ofrece dos maneras para encontrar la localización de un dispositivo. La primera es usando localización por GPS, conocida por ser muy precisa en espacios abiertos, bastante imprecisa en entornos

de interiores, y por tener un consumo de batería bastante alto. La segunda manera que ofrece Google es posicionamiento en base a WiFi [5].

En este modo, en lugar de tener estaciones base que se ocupen de buscar dispositivos, es el propio dispositivo el que se encarga de buscar puntos de acceso a su alrededor. Una vez tomados los datos, se recurre a la base de datos de Puntos de Acceso de Google, en la que aparecen todos los Puntos de Acceso conocidos (exceptuando los que se excluyen usando el comando `_nomap`) y se comprueba dónde están los Puntos de Acceso encontrados. De esta manera se puede obtener una localización estimada del dispositivo que, evidentemente, será más precisa cuantos más Puntos de Acceso se encuentren en el entorno.

Igualmente, este sistema también es capaz de posicionar un dispositivo a partir de las torres de telefonía celular que tiene alrededor, o gracias a la dirección IP pública que se le asigna. Es dependiente del dispositivo y de la configuración cuál de todas las alternativas se utiliza, incluso es posible utilizar varias a la vez con el fin de obtener resultados más precisos.

2.2.3. Accuware Indoor Navigation

Esta herramienta [1] permite la localización en tiempo real de dispositivos Android e iOS. Se basa en una aplicación nativa para cada teléfono, que periódicamente escanea el medio en busca de señales WiFi o iBeacons. Una vez detectados estos datos, es posible realizar el posicionamiento en un entorno local, en el dispositivo, o enviar la información a un servidor que la procese y devuelva la posición estimada. Esta solución requiere un entrenamiento previo del medio, en un proceso de Fingerprinting.

Es evidente que esta solución no es válida, primero porque se requiere de una aplicación instalada en cada dispositivo a localizar, cosa que no es realizable y segundo, porque la localización se realiza en base a los datos que captura el dispositivo, por lo tanto se depende del acceso a los datos que recibe el teléfono, que tampoco es posible en este caso.

Queda claro que ninguno de estos sistemas son válidos para la solución que se busca, que debe ser abierta y transparente para el usuario. En estas soluciones que se han mostrado es preciso tener control sobre los dispositivos que se localizan y control sobre las estaciones que se ocupan de localizarles. En el ámbito de este proyecto, solo es posible controlar las estaciones que localizan, pero no tenemos posibilidad de controlar los dispositivos que portan los usuarios, ni tampoco las bases de datos que controla Google por ser una empresa grande. Por lo tanto, es necesario buscar una solución que nos permita saltar esas limitaciones.

Diseño del sistema de posicionamiento

En este capítulo se procederá a comentar las decisiones referentes al diseño del sistema. Por un lado, se comentarán los aspectos relativos al lugar elegido para el desarrollo y los aspectos hardware del sistema. Por otro, se explicará el funcionamiento lógico de la herramienta desarrollada.

3.1. Entorno de despliegue y sensores a desplegar

En esta sección se va a comentar la elección del lugar de despliegue del sistema, así como las características generales de los nodos que componen el sistema.

3.1.1. Elección del entorno de despliegue

Para el despliegue de la red de sensores utilizada en el desarrollo de este trabajo, se ha escogido un entorno real. Este ha sido el edificio conocido como “Mercado del Este” en Santander. Este Mercado, cuya construcción data de 1842, alberga una serie de pequeños comercios. Es un lugar de encuentro popular y, por lo tanto, es interesante el estudio de la afluencia de gente en este lugar.

En cuanto a las características técnicas del edificio, es un recinto rectangular de 40x60m, con una estructura simétrica en la que se encuentran isletas con comercios que forman pasillos. Al estar estructurado de esta manera, se simplifica en gran medida la tarea de realizar localización de dispositivos, ya que es posible cubrir toda la superficie del recinto con los nodos que se instalan y todos tendrán una amplia línea de vista a su alrededor.

3.1.2. Sensores desplegados

En cuanto a la parte hardware, es necesario destacar una serie de componentes necesarios para el funcionamiento del sistema. Se ha decidido desplegar



Figura 3.1: Vista exterior del Mercado del Este.

una serie de sensores, que cumplirán distintas funciones en el sistema desarrollado.

En primer lugar, se ha instalado un adaptador de red WiFi. Este adaptador hará las veces de comunicador entre la parte central del nodo y el medio inalámbrico. Este adaptador permitirá la monitorización del canal, capturando los paquetes que circulan por el medio y procesándolos de la manera que describimos en el siguiente capítulo. Para realizar esta función, es necesario que tanto el adaptador como el driver que lo controla soporten el Modo Monitor de IEEE 802.11.

Se ha decidido instalar también un adaptador de red a Bluetooth, con el objetivo de poder también detectar los dispositivos Bluetooth presentes en el entorno. Actualmente este dispositivo no está siendo utilizado, pues se ha centrado el funcionamiento del sistema en la tecnología WiFi, pero se decidió instalarlo para poder trabajar con él en un futuro.

En el apartado de los sensores ambientales, se ha elegido un sensor de temperatura y humedad relativa. Este sensor permitirá controlar dichas variables y conocer si alguna de ellas ha llegado a un valor crítico. Además, servirá para relacionar los datos obtenidos mediante los otros sensores con los de temperatura y humedad relativa.

Por último, se ha instalado también un dosímetro de campo radioeléctrico, sensor que puede medir los campos electromagnéticos en distintas bandas de frecuencia y detectar la energía que está siendo radiada por los dispositivos inalámbricos que emiten en dichas frecuencias. Se utilizará en este proyecto como una manera de comparar los resultados proporcionados por los sensores inalámbricos con los que proporcione el propio dosímetro, ya que es posible hacerlos operar en la misma frecuencia.

3.2. Arquitectura lógica del sistema

En esta sección se presentará la arquitectura del sistema desarrollado.

Tal y como se ve en la figura 3.2, en el primer nivel de la arquitectura se encuentran los dispositivos de captación utilizados para monitorizar tanto los distintos fenómenos ambientales como la actividad de red IEEE 802.11. Esta función la desarrollarán los nodos medidores que se han instalado en el Mercado. Una vez adquiridas las diferentes observaciones estas se envían a un servidor central que primero las organizará según el parámetro al cual se refiera y posteriormente las almacenará en la base de datos para su posterior tratamiento.

El análisis de la información también tiene lugar a nivel de servidor mediante una aplicación local que accede a la información en bruto de las observaciones (nivel de señal por dispositivo) y a partir de ellas genera información procesada como es la localización del dispositivo o el número de personas.

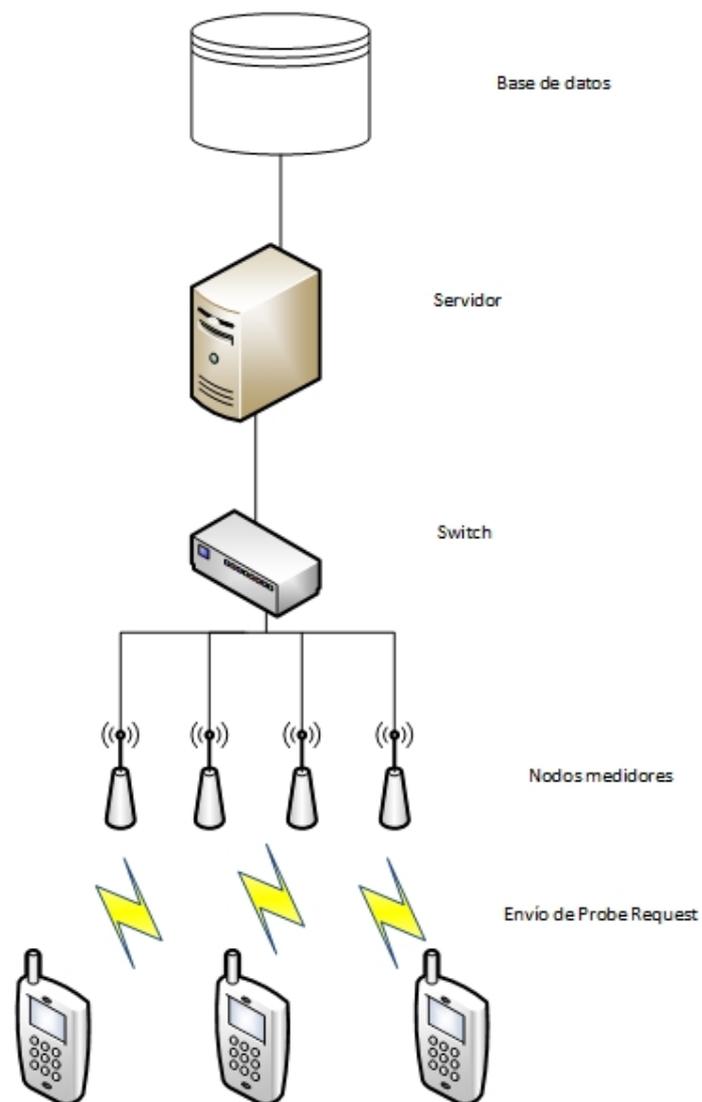


Figura 3.2: Arquitectura lógica del sistema.

Implementación y despliegue del sistema

En este capítulo se van a describir los aspectos técnicos del desarrollo del sistema: montaje, dispositivos hardware e implementación software.

4.1. Descripción del despliegue

En esta sección, se va a proceder a comentar la instalación de los nodos en el Mercado.

4.1.1. Instalación

Al ser el Mercado un recinto simétrico, con puntos clave fácilmente reconocibles, resulta sencillo escoger los lugares de despliegue de los nodos. Por simple inspección, resulta evidente que es necesario instalar nodos en cada puerta del Mercado. Como se disponía de 8 nodos para la instalación y hay 6 puertas, se decidió que para completar la cobertura del Mercado los dos nodos restantes se colocarían en el centro de los dos extremos cortos del mismo.

Estas posiciones benefician claramente a los algoritmos que se van a usar para el cálculo de las posiciones, además de que se controlan las entradas y salidas correctamente. En especial la localización mediante nuestro algoritmo de centroides se ve beneficiado, ya que todos los puntos posibles en los que un dispositivo puede estar se encuentran en el medio de varios nodos y, por lo tanto, es posible calcular cualquier posición en la que se encuentre un usuario móvil.

Como ya se ha comentado anteriormente, cada nodo está conectado a la red de datos, así como a un cable PoE (Power over Ethernet), que alimenta al nodo a través de un adaptador micro USB. De este modo, la instalación física de la infraestructura se simplifica en gran medida ya que se restringe el despliegue del cable de datos a todos los puntos. Para su conexión a la red se ha empleado un switch, colocado en una sala adjunta al Mercado, que permite tanto su gestión como la de sus conexiones de forma remota por lo

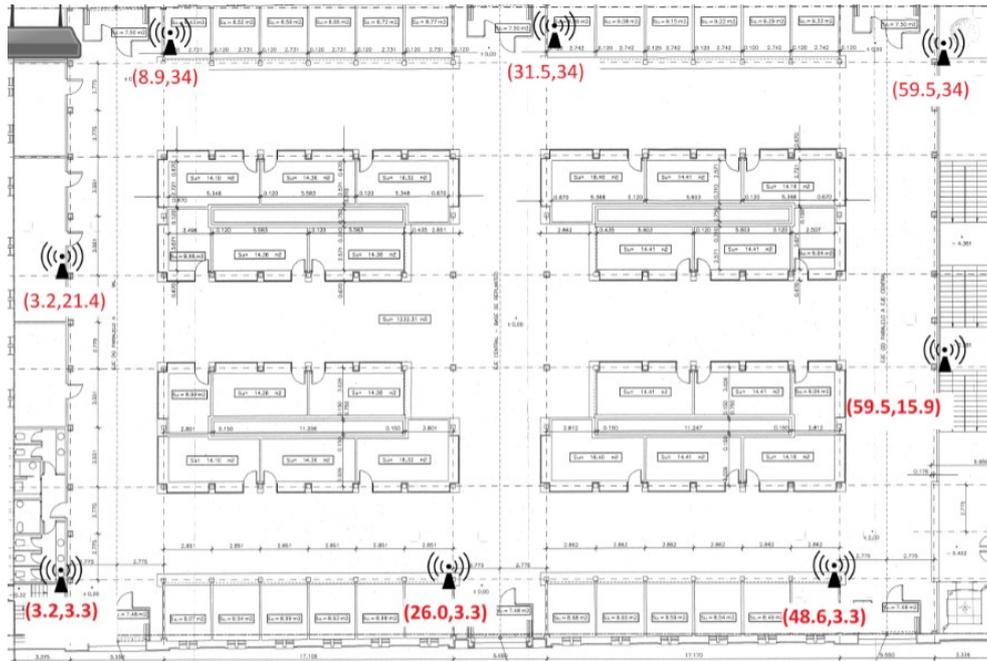


Figura 4.1: Plano del Mercado del Este con la posición de los nodos instalados.

que, en caso de fallo de alimentación o de algún nodo, es posible recuperar el funcionamiento sin necesidad de desplazarse hasta allí.

En la figura 4.1 se puede observar que en prácticamente todos los puntos en los que se puede encontrar un usuario existe línea de vista entre su dispositivo y algún nodo medidor, lo que hace que el funcionamiento del sistema mejore de una manera notable.

Como puede observarse en el plano, se ha establecido un sistema de coordenadas con el fin de poder situar claramente los puntos de referencia y los puntos medidos al ejecutar los programas implementados. Para ello, se ha establecido el punto (0,0) en el origen del plano (esquina izquierda de abajo) y se van aumentando las coordenadas a partir de ahí.

4.1.2. Hardware instalado

Aunque anteriormente se han comentado los tipos de sensores utilizados, se va a describir ahora más en detalle las especificaciones hardware de cada uno de los sensores instalados, así como de la placa que sirve como procesador.

Raspberry Pi 2 Model B

La Raspberry Pi 2 Model B [8] es una placa de prototipado que cuenta con procesador, memoria RAM y procesador gráfico. Esta placa se caracteriza por ser de hardware libre y tener un precio muy reducido, lo que la hace muy

popular para aplicaciones de DIY o para su instalación en lugares donde no es posible la colocación de equipos completos.

<p>Specifications</p> <p>Chip Broadcom BCM2836 SoC</p> <p>Core architecture Quad-core ARM Cortex-A7</p> <p>CPU 900 MHz</p> <p>GPU Dual Core VideoCore IV® Multimedia Co-Processor Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure</p> <p>Memory 1GB LPDDR2</p> <p>Operating System Boots from Micro SD card, running a version of the Linux operating system</p> <p>Dimensions 85 x 56 x 17mm</p> <p>Power Micro USB socket 5V, 2A</p>	<p>Connectors:</p> <p>Ethernet 10/100 BaseT Ethernet socket</p> <p>Video Output HDMI (rev 1.3 & 1.4)</p> <p>Audio Output 3.5mm jack, HDMI</p> <p>USB 4 x USB 2.0 Connector</p> <p>GPIO Connector 40-pin 2.54 mm (100 mil) expansion header: 2x20 strip Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines</p> <p>Camera Connector 15-pin MIPI Camera Serial Interface (CSI-2)</p> <p>JTAG Not populated</p> <p>Display Connector Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane</p> <p>Memory Card Slot Micro SDIO</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

(a) Especificaciones

(b) Conectores

Figura 4.2: Características técnicas de la Raspberry Pi 2 Model B.

Este pequeño ordenador está preparado para funcionar con distribuciones de Linux especializadas, que cumplen distintas tareas en función del uso que se le quiera dar. En este caso utilizamos una distribución basada en Debian llamada *Raspbian*. Esta distribución cumple con las necesidades generales que se requieren en el proyecto.

La placa está dotada de diversos interfaces de entrada y salida para interactuar con los dispositivos externos. Es posible conectarlo a una red cableada a través de Ethernet, insertar o recuperar datos a través de USB, tiene posibilidad para conectar una cámara, etc...

Para que la placa funcione correctamente, es necesario añadirle una serie de periféricos. Lo primero es, evidentemente, la alimentación. La Raspberry Pi utiliza una alimentación de 5 V y como máximo 2 A, a través de un conector Micro USB o alimentándolo directamente al puerto GPIO. El siguiente elemento necesario para su correcto funcionamiento es una tarjeta micro SD. En esta tarjeta se almacenan tanto el Sistema Operativo como los datos de usuario, ya que la Raspberry no dispone de memoria interna. Por último, aunque no es estrictamente necesario, se le ha añadido una carcasa para evitar que se dañe por estar en un entorno exterior.

Adaptador de red WiFi

Una vez seleccionado el dispositivo básico del sistema, es necesario dotarlo de medios para interactuar con el entorno. En el apartado de WiFi, se ha instalado una tarjeta de red que permite el modo monitor, ya que, como se ha descrito en el capítulo anterior, esta será la forma en la cual sensaremos el canal y se capturarán los paquetes necesarios para la localización de los dispositivos que emiten.

Se ha elegido el modelo TL-WN722N de TP LINK [11], por ser un compo-

nente barato y de altas prestaciones en ese rango de precios. Este adaptador posee una conexión USB que le permite conectarse a la Raspberry Pi sin la necesidad de usar los puertos GPIO, lo que simplifica en gran manera la configuración. Además, su driver es soportado por el sistema *Raspbian*.

Standards	IEEE 802.11g, IEEE 802.11b
Interface	USB 2.0 interface
Wireless Signal Rates	Up to 150Mbps
Frequency Range	2.4-2.4835GHz
Wireless Transmit Power	20dBm(MAX)*
Modulation Type	DBPSK, DQPSK, CCK, OFDM, 16-QAM, 64-QAM
Receiver Sensitivity	130M: -68dBm@10% PER
	54M: -88dBm@10% PER
	11M: -85dBm@8% PER
	6M: -88dBm@10% PER
	1M: -90dBm@8% PER
Antenna	4dBi Detachable Omni Directional Antenna with RP-SMA connector
Security	64/128-bit WEP, WPA/WPA2, WPA-PSK/WPA2-PSK (TKIP/AES)
Support Operating System	Windows 7/Vista/XP/2000
Operating temperature	0°C~40°C (32°F~104°F)
Storage temperature	-40°C~70°C (-40°F~158°F)
Relative humidity	10% ~ 90%, non condensation
Storage Humidity	5%~95% non-condensing
Dimensions	3.7 x 1.0 x 0.4in. (93.5 x 26 x 11mm)

(a) Especificaciones



(b) Adaptador

Figura 4.3: Características técnicas del adaptador WiFi.

Adaptador de red Bluetooth

Este adaptador, de manera análoga al de WiFi, nos servirá para detectar a los usuarios de Bluetooth y poder operar con ellos. En este caso, hemos utilizado un adaptador Bluetooth que contiene el chip CSR 4.0. Al contrario que en WiFi, no se presenta en este área ningún requerimiento especial, por lo tanto es válido cualquier chip. Se ha elegido éste por su relación calidad-precio y porque con sus prestaciones es válido para lo que se necesita.

Sensor de Temperatura

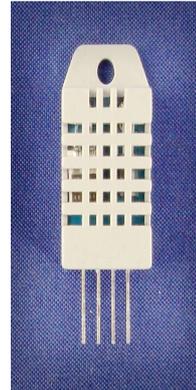
Por último, en la mitad de los nodos sensores se ha colocado un sensor de temperatura, a fin de controlar posibles subidas de temperatura y hacer estudios de la influencia del número de personas en la temperatura. Se ha elegido el popular DHT-22 [3], ya que sus prestaciones son altas y teniendo en cuenta lo reducido de su precio.

Es necesario mencionar que este sensor no tiene un mecanismo de entrada USB como los anteriores, sino que se conecta directamente a los pines de GPIO presentes en la Raspberry Pi, por lo que ha sido necesario implementar el consiguiente circuito de adaptación.

4.2. Implementación de la plataforma software para la captación de datos

Una vez visto el detalle de los dispositivos físicos empleados, en este apartado se va a proceder a explicar el funcionamiento del software desarrollado,

Model	DHT22	
Power supply	3.3-6V DC	
Output signal	digital signal via single-bus	
Sensing element	Polymer capacitor	
Operating range	humidity 0-100%RH;	temperature -40-80Celsius
Accuracy	humidity +2%RH(Max +5%RH); temperature \pm 0.5Celsius	
Resolution or sensitivity	humidity 0.1%RH;	temperature 0.1Celsius
Repeatability	humidity +1%RH;	temperature +0.2Celsius
Humidity hysteresis	+0.3%RH	
Long-term Stability	+0.5%RH/year	
Sensing period	Average: 2s	
Interchangeability	fully interchangeable	
Dimensions	small size 14*18*5.5mm,	big size 22*28*5mm



(a) Especificaciones

(b) Sensor

Figura 4.4: Características técnicas del sensor de temperatura DHT-22.

tanto en el lado de los nodos medidores como en el del servidor.

4.2.1. Tecnologías utilizadas

El primer aspecto a destacar en el desarrollo software es la elección del lenguaje de programación, ya que esta decisión tiene un impacto en el resto del proceso. En este sentido, se ha utilizado el lenguaje *Python* para el desarrollo de todo el software. Se ha elegido este lenguaje debido a su simpleza, ligereza y su gran apoyo de la comunidad. Esto se traduce en que es un lenguaje ideal para este tipo de sistemas, en el que todo apoyo de otros usuarios implica una gran ayuda.

Una vez escogido el lenguaje, es necesario el uso de librerías de *Python* para realizar todas las tareas que se requieren en el sistema. La parte más importante del desarrollo ha sido la de la captación de las señales IEEE 802.11 en los nodos medidores. Para ello, se han utilizado las siguientes librerías.

Airodump-ng

Aircrack-ng [2] es una suite de herramientas que permiten realizar tareas de seguridad en redes IEEE 802.11. La herramienta que se ha utilizado dentro de la suite es *airodump-ng*, que realiza tareas de monitorización de la red. Permite controlar qué AP están presentes en el entorno y qué dispositivos IEEE 802.11 están alrededor. Todo esto se realiza sobre una interfaz en modo monitor.

La operativa de esta herramienta es la siguiente:

1. Se configura la interfaz IEEE 802.11 como modo onitor, de manera que sea capaz de escuchar cualquier paquete en el canal. Para llevar a cabo esta configuración se puede emplearotra de las herramientas presentes en

la suite *Aircrack-ng*, o utilizarcon el comando *iwconfig* de las *wireless-tools* incluidas en el sistema *Raspbian*.

2. Se comienza a escuchar el canal con la herramienta *Airodump-ng*. Se detectan los paquetes que circulan por la red y se procesan los paquetes para extraer la información relevante como por ejemplo la potencia recibida o la dirección MAC de los dispositivos.
3. La herramientaes capaz de mostrar a través de su interfaz gráfica los puntos de acceso y sus parámetros representativos así como los dispositivos detectados.

```

CH 8 ][ Elapsed: 0 s ][ 2016-06-20 13:00
BSSID          PWR  Beacons    #Data, #/s  CH  MB  ENC  CIPHER AUTH ESSID
00:25:9C:3C:2D:79 -62      3           0  0  11  54  WPA  CCMP  PSK  LTELE
1C:B7:2C:D0:13:66 -39      3           0  0  6   54e WPA2 CCMP  PSK  <length: 7>
38:2C:4A:65:6F:A0 -42      4           0  0  6   54e WPA2 CCMP  PSK  <length: 8>
00:1D:7E:28:20:0D -75      2           0  0  6   54e WPA2 TKIP  PSK  o12
A8:54:B2:92:FA:11 -38      1           0  0  6   54e WPA2 CCMP  PSK  gwMDEV
E0:3F:49:6B:9F:88 -46      3           0  0  7   54e WPA2 CCMP  PSK  RADICAL
00:0F:66:C8:99:54 -46      5           0  0  4   54e WPA2 CCMP  PSK  <length: 5>
52:46:5D:01:7E:C1 -33      6           0  0  1   54e WPA2 CCMP  PSK  mshop
50:46:5D:01:7E:C0 -34      6           0  0  1   54e WPA2 CCMP  PSK  <length: 7>
F8:1A:67:5A:99:75 -70      5           0  0  1   54e WPA2 CCMP  PSK  LUSITANEA-2
20:B3:99:A8:BA:A8 -67      4           1  0  1   54e WPA2 CCMP  MGT  eduroam
20:B3:99:A9:79:D8 -44      5           1  0  1   54e WPA2 CCMP  MGT  eduroam
00:12:17:A7:C6:78 -72      1           5  2  1   54e WPA  TKIP  PSK  RTLS
52:46:5D:03:02:E9 -41      6           0  0  1   54e WPA2 CCMP  PSK  LEXNET2
50:46:5D:03:02:E8 -42      8           0  0  1   54e WPA2 CCMP  PSK  BEE_TLMAT
00:16:B6:2B:84:CE -46      6           0  0  1   54e WPA2 CCMP  PSK  TLMAT_LAB

BSSID          STATION          PWR  Rate  Lost  Frames  Probe
(not associated) 9C:4E:36:94:23:A4 -65    0 - 1    0      1
(not associated) 20:B3:99:A9:CF:F0 -45    0 - 6    5      2  MyDomain
00:12:17:A7:C6:78 A8:66:7F:0C:EA:6C -1     1e- 0    0      5
    
```

Figura 4.5: Ejemplo de ejecución del programa airodump-ng.

Una vez se han procesado los paquetes y la herramienta nos ha proporcionado la información que necesitamos, se insertaesa información en un archivo de texto y se analiza. Esto presenta dos inconvenientes: el primero, es necesario generar un archivo con toda la información que se nos ofrece. Este archivo escala rápidamente ya que aparecen muchas entradas y por lo tanto es preferible usar una solución que no genere ficheros de tal magnitud. El segundo problema es que al realizar el análisis se pueden dar errores ya que la herramienta no nos ofrece la salida en un formato demasiado adecuado.

Otro problema que nos encontramos es que esta herramienta escanea la red y captura todos los paquetes presentes en ella. Esto nos lleva a un problema de saturación en el driver de la tarjeta de red, que ocasiona que algunos paquetes se pierdan y por lo tanto las medidas no sean totalmente fiables.

Scapy

Scapy [10] es un programa de manipulación de paquetes interactivo. Permite decodificar paquetes de un amplio número de protocolos, enviarlos o capturarlos. En esta ocasión, se aprovecharán sus capacidades de captura y decodificación de paquetes.

Se pone el sistema a funcionar, de nuevo, en modo monitor. La herramienta escanea todos los paquetes y cada vez que llega uno se lo pasa a un manejador. Este manejador comprueba si el paquete es de tipo Probe Request y, si lo es, lo inserta en la base de datos. Con esta solución se ahorra el paso intermedio de un fichero para el funcionamiento del sistema. Sin embargo, persiste el problema de saturación en el driver y por tanto se pierden paquetes.

Tshark

Tshark [12] es un analizador de protocolos de red. Permite capturar datos en vivo de una red, o leer paquetes de un archivo previamente guardado. Es sobre todo conocido por su aplicación de escritorio, *Wireshark*. En este caso, se va a utilizar un wrapper de *Python* llamado *Pyshark*, que facilita el trabajo en gran medida.

El funcionamiento de esta herramienta es el siguiente:

1. Se configura la interfaz IEEE 802.11 en modo monitor para lo cual se emplea el comando *iwconfig*.
2. Se inicia la captura con Tshark desde la aplicación *Python*, realizando un filtrado previo para que todos los paquetes que se capturen sean de tipo Probe Request.
3. El wrapper nos devuelve objetos de tipo paquete, implementados por *Pyshark*. De estos paquetes se cogen los parámetros relevantes, como potencia recibida y dirección MAC y se insertan en la base de datos.

Una vez probadas y estudiadas todas las alternativas, se ha decidido usar Tshark para esta tarea, ya que es la herramienta que mejor responde a los requerimientos del sistema.

Además de captar la información de red IEEE 802.11, también ha sido necesario obtener los datos de temperatura y humedad provenientes del sensor DHT-22. Para ello, se ha hecho uso de la librería *Adafruit_DHT*, que opera directamente sobre los pines de GPIO y realiza una conversión de las señales que recibe del sensor y las transforma en datos numéricos.

Una vez captadas las señales, se ha procedido a enviar las observaciones resultantes al servidor que se ha montado para almacenar la información en la base de datos. Para realizar esa tarea se ha utilizado la librería ZMQ [13]. Esta

herramienta provee mecanismos de alto nivel para habilitar la comunicación entre distintas aplicaciones. Para ello, se apoya en el uso de sockets TCP, a través de los cuales envía la información deseada. En este caso, se ha habilitado un puerto en el servidor al que envían la información los ocho nodos medidores. En el lado del servidor, se han desplegado diferentes workers, que serán los encargados de escuchar los datos enviados por cada uno de los nodos e insertarla en la base de datos adecuada. Se han usado 8 workers, uno por cada nodo. Esto se ha hecho de esta manera porque se ha observado que si cada nodo realiza las inserciones por su cuenta, el retardo aumenta en gran medida, así que se ha optado por implementar esa tarea en el servidor.

ZMQ provee un medio en el que dos aplicaciones pueden comunicarse utilizando sockets TCP. De esta manera, la aplicación que envía genera un socket de tipo PUSH y la aplicación que se encarga de recibir los datos genera otro de tipo SUB. Cada vez que la aplicación emisora envíe algo, le llegará por su socket y ZMQ se encarga de procesar los datos y convertirlos de manera que puedan ser utilizados por el receptor. Por otro lado, existe otro tipo de socket de tipo PUB, que en el caso de este proyecto se utiliza para realizar las peticiones a la base de datos.

4.2.2. Descripción técnica de la implementación en el servidor

Para el funcionamiento del servidor se ha implementado también un software, compuesto por varias clases y un programa principal.

Clase *db.handler*

A la hora de insertar datos en una base de datos MySQL, hay que realizar dos instrucciones. La primera, que sería la ejecución de la sentencia SQL, indica los datos que se quieren insertar, en que tabla, etc. Con esta instrucción, sin embargo, no se produce ningún cambio en la base de datos, sino que los cambios se quedarían encolados a la espera de la siguiente. Esta siguiente instrucción sería el commit, que se encarga de que los cambios queden reflejados en la base de datos.

En esta clase se realiza la tarea de inserción en la base de datos. Se compone de varias sentencias SQL que se ejecutarán en función de qué tipo de datos queramos insertar. Se ha decidido que, para acelerar el funcionamiento del sistema, no se realizará el commit de los datos cada vez que se realice una inserción, sino que se insertará cuando se hayan insertado 5 mensajes o hayan pasado 10 segundos.

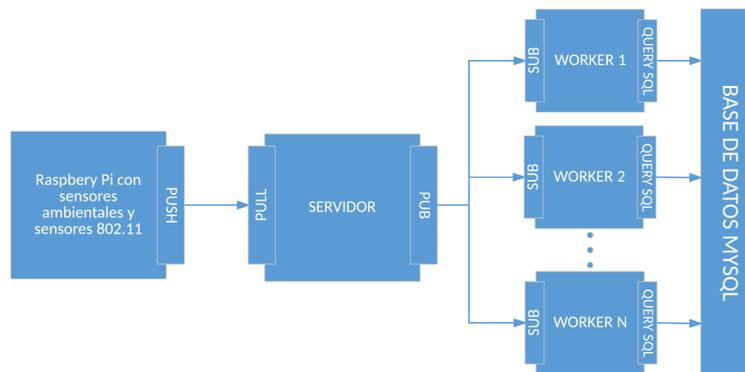


Figura 4.6: Diagrama de flujo que muestra el funcionamiento del servidor.

Clase *serverHandler*

Esta clase se encarga de manejar los sockets de pull y pub y permite operar con los mensajes que llegan al servidor.

Clase *workerHandler*

En esta clase se implementa el funcionamiento de los workers que se encargarán de manejar la información de cada uno de los distintos nodos. Cada vez que se crea un worker, éste se suscribe a los datos provenientes de uno de los nodos. A partir de ahí, cada vez que llegue un mensaje proveniente de ese nodo, lo insertará en la base de datos.

locser.py

Este es el programa principal que inicia todas las funciones del servidor. Al comienzo de su ejecución se establecen los sockets en los que escuchará y por los que enviará la información que le llegue. Después de establecer los sockets, se encarga de activar todos los hilos de workers para la ejecución.

Estructura de la base de datos

Para el almacenamiento de la información contenida en las observaciones capturadas por los nodos medidores, se utilizará una base de datos MySQL para el almacenamiento de los datos. Se ha elegido utilizar este tipo de bases de datos por su simpleza de operación y su velocidad de trabajo. El acceso a los datos de la misma se puede hacer tanto mediante comandos MySQL o

como utilizando cualquiera de los programas con interfaz gráfica, en nuestro caso HeidiSQL.

En la base de datos creada, se han generado varias tablas, cada una con un propósito específico dentro del proyecto. Se puede ver claramente su aportación si se distinguen correctamente las dos partes del sistema: Localización vía WiFi y control de temperatura. Además de las tablas creadas para estos propósitos, se ha incluido una tabla *NODES* con información de los nodos de medida.

A continuación, se detallan los campos que se han insertado en la base de datos:

Tabla WIFI_DATA

- **id:** Valor asignado a la entrada a la base de datos en orden creciente. Sirve como control del número de inserciones que se realizan y como clave primaria de la tabla.
- **nodeid:** Identificador del nodo que ha realizado la inserción en la base de datos.
- **mac.hash:** Dirección física del dispositivo que se ha detectado, previamente hasheada para evitar problemas de privacidad.
- **rss:** Medida de la potencia detectada por el nodo.
- **ssid:** String de la estación base a la que se quiere conectar el dispositivo. Se utiliza como elemento diferenciador en caso de que alguno de los terminales detectados cambie su dirección física (como medida adicional de seguridad)
- **timestamp:** Marca de tiempo de la base de datos. Usado para poder estudiar el retardo entre el envío del insert por el nodo medidor y la propia inserción en la base de datos.
- **timestamp_raspberry:** Marca de tiempo fijada por el nodo medidor.
- **timestamp_packet:** Marca de tiempo de la detección del paquete en cuestión. Sirve como medida del retardo producido por la congestión de paquetes en el nodo medidor (si lo hubiera)
- **packet_type:** Tipo del paquete detectado. Usado en el caso de capturar varios tipos de paquetes, con el objetivo de poder diferenciarlos.

Tabla ENV_DATA

- **id:** Valor asignado a la entrada a la base de datos en orden creciente. Sirve como control del número de inserciones que se realizan y como clave primaria de la tabla.

- `nodeid`: Identificador del nodo que ha realizado la inserción en la base de datos.
- `temperature`: Valor de la temperatura detectado por el nodo.
- `humidity`: Valor de la humedad relativa detectado por el nodo.
- `timestamp`: Marca de tiempo de la base de datos. Usado para poder estudiar el retardo entre el envío del insert por el nodo medidor y la propia inserción en la base de datos.
- `timestamp_raspberry`: Marca de tiempo del nodo medidor.

4.2.3. Descripción técnica de la implementación en los nodos

Una vez aclaradas las herramientas utilizadas, a continuación describiremos el funcionamiento técnico del software desarrollado. El funcionamiento se ha dividido en distintas clases y dos programas principales, que se ocuparán de unificar el funcionamiento de esas clases.

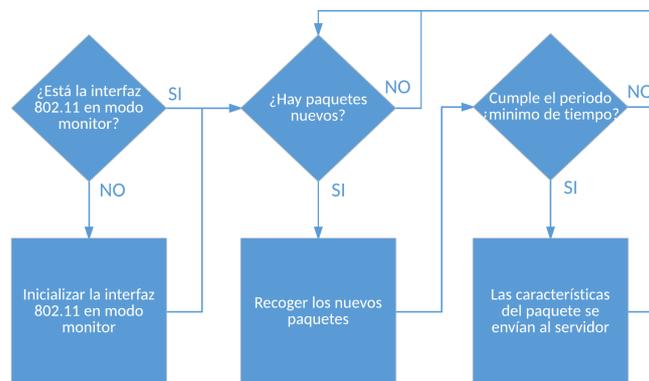


Figura 4.7: Diagrama de flujo que muestra el funcionamiento de la captación de IEEE 802.11.

Clase `wifi_handler`

En esta clase se realizan las tareas de captación de los paquetes IEEE 802.11. Cuando se crea la instancia de esta clase, la primera tarea a realizar es la de activar y configurar la interfaz IEEE 802.11 en modo monitor. Para ello se emplea el comando del sistema `iwconfig`. Una vez levantada la interfaz, se procede a iniciar la captura de los paquetes. Al realizar dicha captura se aplica

un filtro para solamente tener en cuenta los paquetes de tipo Probe Request, ahorrando así el uso de memoria al tener que capturar todos los paquetes que circulan por la red. Cada vez que llega un paquete nuevo se extraen de él la dirección MAC del emisor, la RSSI con la que llega al nodo y la hora con la que llegó al driver (para evitar problemas de retardo). Una vez extraídos los parámetros, se comprueba que el paquete cumple los requisitos para ser enviado a la base de datos (en base a una caché que se explicará más adelante) y, si hay que enviarlo, se lo pasará al hilo que enviará ese paquete hacia la base de datos.

Caché implementada

Con el fin de evitar saturación en la base de datos y de limitar la cantidad de información generada por cada nodo medidor, se ha elaborado una caché, que se encarga de que solamente se insertan los datos en la base de datos cuando se cumplan una serie de condiciones.

En un principio, se eligió que solamente se insertara un paquete cuando hubieran pasado 10 segundos desde la última inserción de un paquete proveniente de ese dispositivo en la base de datos. Esto generaba un problema importante, ya que obligaba a tener todos los nodos medidores perfectamente sincronizados para poder identificar el mismo paquete en dos inserciones distintas.

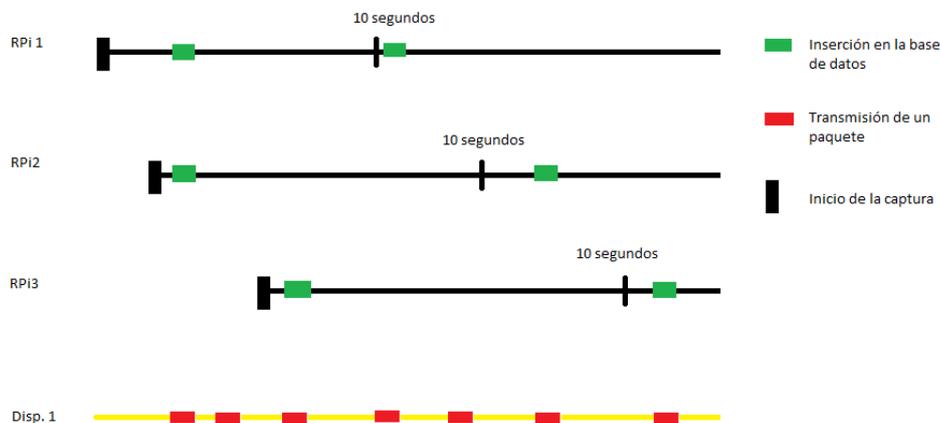


Figura 4.8: Línea de tiempos mostrando que insertando cada 10 segundos no hay sincronización.

Para solventar ese problema, se decidió que en lugar de enviar cada 10 segundos, se enviaría el primero que llegara en cada decena de segundos. Así por ejemplo, si el último paquete se envió en el segundo 38 y el siguiente llegara en el segundo 43, este también se enviaría. De esta manera, además de conseguir una caché con la que se tamiza la cantidad de paquetes a almacenar, consigui-

mos una sincronización parcial entre los nodos medidores, ya que siempre van a enviar el primer paquete que reciban en cada decena de segundos. Con esto se logra también que los cálculos sean mucho más precisos, ya que conseguimos que todos los valores que se usan pertenezcan al mismo paquete transmitido y por lo tanto tendrá los mismos parámetros de transmisión.

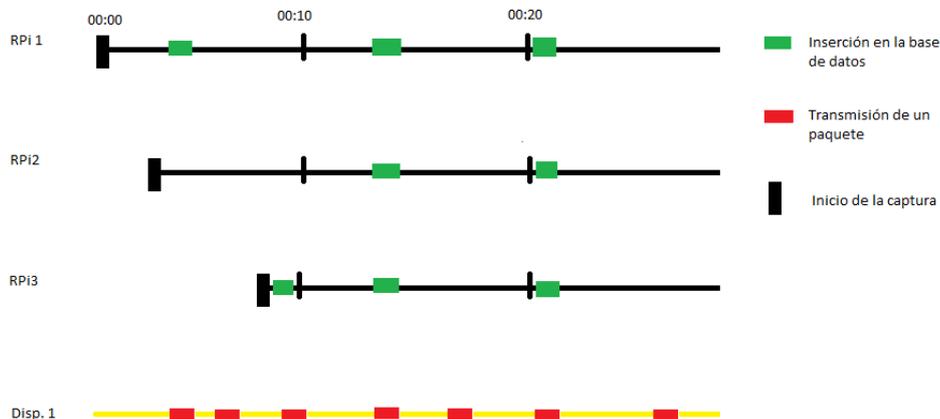


Figura 4.9: Línea de tiempos realizando las inserciones cuando pasan las decenas de segundo. Se demuestra que hay una pequeña sincronización entre los nodos.

Clase *zmq_handler*

En esta clase se abre la conexión con el socket del servidor al que se enviará la información. Una vez establecida esa conexión, se enviará la información requerida en forma de array, con un parámetro “E” o “W” en función de si los datos son de sensor de temperatura/humedad o de señales IEEE 802.11, respectivamente.

Clase *hash_handler*

Esta clase se ocupa de realizar el hashing de los datos si en los parámetros de configuración se ha configurado que se realice esta medida de seguridad. Se utiliza la librería *hashlib* de *Python*.

locffer.py

Este es el programa principal, que unifica todas las clases anteriormente comentadas. Básicamente inicializa todos los objetos necesarios para el correcto funcionamiento del sistema. Además, se ha implementado un chequeo para considerar el problema de que el driver de la tarjeta de red se colapse y deje de capturar, dejando así de funcionar el programa sin dar un mensaje de error.

Para ello, se ha optado por comprobar en intervalos de tiempo regulares que se hayan capturado nuevos paquetes y reiniciar el programa en caso de que no se haya recibido nada.

environmental_main.py

Este programa se encarga de realizar las mediciones relativas a temperatura y humedad y enviarlas al servidor para ser procesadas. Con el fin de evitar que aparezcan valores extraños debido a los pines de GPIO, que a veces introducen error, se realizan 5 medidas sucesivas y se toma la mediana de todas ellas. De esta manera, reducimos en gran medida la probabilidad de obtener valores erróneos.

Cabe destacar que los dos programas principales hacen uso de la herramienta supervisor, que se encarga del mantenimiento e inicio de procesos en sistemas *Linux*. Esta herramienta se encarga de que los programas se inicien cuando se enciende el nodo y de que, en caso de fallo, vuelvan a operar automáticamente.

Validación del sistema y resultados

En este capítulo se va a analizar el comportamiento del sistema a partir de los resultados obtenidos tras la puesta en funcionamiento del sistema implementado.

5.1. Metodología para la validación del sistema

En esta sección se comentarán las reglas seguidas a la hora de validar el sistema. Se explicarán los pasos a seguir a la hora de obtener resultados y las pruebas realizadas con el fin de comprobar el correcto funcionamiento de la herramienta.

5.1.1. Pruebas de campo realizadas

Para evaluar el funcionamiento del sistema, se han realizado una serie de medidas de campo para poder comparar los valores reales medidos con los resultados obtenidos a partir de la herramienta. Se han realizado 4 tipos de pruebas: pruebas de potencia, pruebas de posición, pruebas de conteo y pruebas de cadencia de Probe Request.

Para simplificar la tarea de realizar estas medidas, se ha desarrollado una herramienta para dispositivos Android que se encarga de forzar el envío de mensajes de tipo Probe Request cada vez que se pulsa un botón. De esta manera se reduce en gran medida la espera que hay que realizar entre mensajes y se pueden realizar las pruebas de una manera más rápida.

Pruebas de potencia

Estas pruebas han consistido en colocarse en varios puntos del Mercado y sus exteriores y medir la potencia que reciben los nodos medidores del dispositivo móvil utilizado durante las pruebas. De esta manera ha sido posible estimar las capacidades del medio y el comportamiento de las señales en

presencia de paredes, personas, etc. Como resultado de estas medidas, se ha comprobado que aún teniendo en cuenta la atenuación introducida por las paredes exteriores, todos los nodos son capaces de detectar al dispositivo cuando éste se encuentra en el exterior del Mercado. Por lo tanto, aún en el caso de que un dispositivo se encuentre detrás de las paredes del Mercado, es posible que todas los nodos medidores lo detecten.

También se ha observado que, cuando un dispositivo pasa junto a alguna de las puertas, la potencia recibida por los nodos que tienen línea de vista hacia esa puerta es mucho mayor que la que se recibe en los nodos que están tras una pared. Por lo tanto, resulta evidente que la mayor atenuación que reciben las señales viene dada por las paredes que tiene que atravesar dicha señal. Tan considerable es el descenso de la atenuación al no haber paredes, que se ha comprobado que un dispositivo puede ser detectado a distancias de unos 20 metros si se encuentra directamente frente a una puerta.

Pruebas de posición

Estas pruebas se han realizado con el fin de comparar los resultados obtenidos de los algoritmos de posicionamiento con la localización real de un dispositivo dado. Se ha realizado un seguimiento de un dispositivo a lo largo de varias posiciones y se han procesado los datos relativos a dichas posiciones conocidas, comparándolas para comprobar el comportamiento de los algoritmos.

Al haber utilizado la herramienta anteriormente comentada, por cada posición real se han conseguido en torno a unas 7-8 medidas estimadas. De esta manera se consigue un gran número de muestras, de las que podemos sacar datos relevantes.

Posición	Número de medidas	Posición	Número de medidas
[6,6.1]	7	[6,12.4]	7
[6,18.7]	7	[6,24.9]	7
[6,30.9]	6	[17.4,30.9]	8
[28.8,30.9]	9	[28.8,24.9]	8
[28.8,18.7]	7	[28.8,12.4]	6
[28.8,6.1]	7	[17.4,18.7]	7
[17.4,6.1]	9		

Cuadro 5.1: Medidas de posicionamiento realizadas

Pruebas de conteo

Estas medidas tienen como función la comparación entre el número real de personas en un momento dado dentro del Mercado con la estimación de

dispositivos realizada por la herramienta. Una vez tomadas estas medidas, se tratará de buscar una relación entre ambos valores y, si existe, aplicarlo sobre una serie de datos para comprobar la correcta funcionalidad del sistema.

Para la realización de estas medidas, se ha decidido contar el número de personas en intervalos determinados. Se han realizado dos conjuntos de datos en dos días distintos, uno realizando el conteo de las personas presentes cada 6 minutos y otro contando cada 3 minutos. De esta manera se intentará utilizar uno de los conjuntos para comprobar la funcionalidad de la relación obtenida a partir del otro.

Hora	Número de personas	Hora	Número de personas
15:51	55	15:57	60
16:03	51	16:09	48
16:15	43	16:21	53
16:27	55	16:33	58
16:39	52	16:45	47
16:51	44	16:57	41
17:03	44	17:09	38
17:15	30	17:21	37
17:27	30	17:33	40
17:39	37	17:45	42

Cuadro 5.2: Primer conjunto de datos relativos al conteo

Hora	Número de personas	Hora	Número de personas
16:40	45	16:43	40
16:46	40	16:49	56
16:52	42	16:55	43
16:58	42	17:01	45
17:04	47	17:07	42
17:10	41	17:13	38
17:16	38	17:19	35
17:22	37	17:25	42
17:28	40	17:31	39
17:34	41	17:37	36
17:40	38	17:43	35
17:46	40	17:49	42
17:52	35	17:55	39
17:58	37	18:01	42
18:04	45	18:07	40

Cuadro 5.3: Segundo conjunto de datos relativos al conteo

Se puede observar que las medidas son bastante variables y aparecen picos sin razón aparente. Esto se debe a que en un momento puede entrar mucha gente al Mercado y en otro momento se puede vaciar repentinamente. Por otro lado, se ha observado que hay un alto número de personas que probablemente no tenga un smartphone, como por ejemplo personas mayores o niños. Se les ha tenido en cuenta para el conteo, pero más adelante será necesario tener en cuenta este detalle al realizar los cálculos. Además de esto, se ha observado también que hay un número de personas que se mantenían presentes en el recinto durante todo el periodo de medidas, ya fueran trabajadores del Mercado o gente que estaba en el restaurante comiendo. En el caso de que estas personas no posean dispositivos con el WiFi activado, se arrastra ese error durante todo el proceso de medidas.

Cadencia de Probe Request

Durante el desarrollo de la herramienta, se observó que no existe una regla para el envío de mensajes de tipo Probe Request. Estos mensajes no se envían a unos intervalos conocidos, sino que dependen de muchas variables, entre ellas el fabricante, el modelo o el estado del teléfono (Conectado a WiFi, desconectado, bloqueado, desbloqueado, etc.). Se hicieron pruebas con una serie de terminales, de distintos fabricantes, todos bloqueados. Se observó claramente que no es posible establecer un tiempo medio entre mensajes de tipo Probe Request, y este tiempo se puede considerar totalmente aleatorio. Se ha destacado que uno de los factores clave en el envío de este tipo de mensajes es el uso del teléfono. Si el teléfono se está usando, el tiempo entre envíos decrece en gran medida, ya que el teléfono intenta conectarse a una red WiFi, mientras que cuando está bloqueado no lo considera tan necesario.

Modelo	Resultados
LG Nexus 5	Envía cada 5 minutos
Iphone 6	Envía cada pocos segundos, aumenta cuando no se usa
Meizu m2 Note	Envía cada pocos segundos, aumenta cuando no se usa
Samsung Galaxy Note 10.1	Envía cada 1-2 minutos, aumenta a 15 cuando no se usa
Samsung NOTE 1 GT-N7000	Entre 1 y 7 minutos, aleatoriamente
Sony Xperia 4	Envía cada minuto, aumenta a 20 cuando no se usa

Cuadro 5.4: Medidas de cadencia de Probe Request

Como se puede observar en la tabla 5.4, los resultados no son favorables y dejan claro que el carácter de estos mensajes es, a priori, aleatorio. Por lo tanto, queda claro que el parámetro de tiempo entre paquetes Probe Request no puede ser usado para encontrar ningún resultado.

5.1.2. Metodología utilizada para la obtención de resultados

A continuación se van a explicar las directrices tomadas para la obtención de los resultados. Se va a dividir la sección en torno a las diferentes partes de la generación de resultados.

Conteo de personas

Para este resultado, se ha calculado por un lado el número estimado de dispositivos con WiFi activado dentro del Mercado. Este cálculo se ha hecho en función del número de nodos que detectan al dispositivo. Aunque anteriormente se ha comentado que la atenuación presentada por las paredes no es suficiente como para que un dispositivo que se encuentre fuera no sea detectado por algún nodo, se ha observado que son necesarias varias transmisiones para que un dispositivo sea detectado por todos los nodos. Por lo tanto, se ha decidido que el criterio a seguir para considerar un dispositivo como dentro del Mercado sea que haya sido detectado por un número mínimo de nodos. De esta manera, se descartarán los paquetes enviados por dispositivos que pasen junto al Mercado y envíen un paquete esporádico que sea detectado por un reducido número de nodos.

Se varió este umbral mínimo de nodos que detecten un dispositivo para comparar los resultados obtenidos en varios casos. Así se conseguirá aproximar la mejor solución, si la hubiera.

Además, debido al carácter aleatorio del envío de mensajes de Probe Request, se cogerán los paquetes recibidos en un intervalo de 1 minuto después del momento en el que se desea medir. Con esto se consigue captar todos los dispositivos que potencialmente han enviado algo en ese intervalo.

Una vez encontrado el número de dispositivos, se compara este valor con el valor real que se ha medido en las pruebas de campo. Se busca una relación matemática entre ambos valores con la que poder modelar un valor estimado de personas en función del número de dispositivos detectados.

Además de estas medidas, se ha decidido también sacar los resultados del número de dispositivos en un periodo de un día y de una semana. Estos datos no pueden ser correlados con datos reales, porque no se dispone de ellos, pero es posible guiarse por patrones de conducta, como el descenso por la noche o el aumento de dispositivos a la hora de comer. De esta manera se comprobará la validez del sistema también en un intervalo a largo plazo.

Medidas de localización

Para esta parte de la herramienta, se han realizado los cálculos tanto para el algoritmo de centroides como para el de trilateración. De esta manera se

podrán comparar ambos algoritmos tanto entre ellos como con los valores reales.

Para el cálculo de la localización, se han seleccionado los valores tomados por los nodos cada 10 segundos, resultando así en matrices de potencias. Con estas matrices de potencias se ha procedido a calcular las posiciones de cada una de las posibles medidas obtenidas por ambos criterios, para después compararlos con los valores reales usados en la fase de pruebas, calculando los errores encontrados en cada medida, así como el error medio en todas las medidas. Se han representado los errores de cada medida en un histograma para observar la tendencia del sistema. Para finalizar, se han representado una serie de puntos reales frente a puntos medidos, con el fin de hacer una comparación gráfica bidimensional de ambos valores y poder observar la diferencia entre unas y otras medidas.

Una vez realizados estos cálculos, se procede a realizar el mismo proceso para un número menor de nodos medidores. Se han escogido los 4 nodos que forman una cruz, para de esta manera poder comprobar el funcionamiento del sistema en caso de tener un menor número de referencias.

Correlación entre número de dispositivos y temperatura/humedad

Parece evidente que la relación entre temperatura, humedad y número de dispositivos será que cuanta más gente, mayor temperatura y humedad habrá en el ambiente. Sin embargo, al ser un lugar grande y con numerosas entradas, es posible que el influjo de la gente no tenga un gran efecto en estas medidas.

Para realizar la comparación de los resultados, se ha decidido representar ambos valores en la misma gráfica, para de esta manera poder visualizar la relación entre ambos valores. Al tener varios nodos con sensores de ambiente, se van a representar los valores de ambiente de todos los nodos en diferentes gráficas, para ver así si hay diferencia entre las medidas.

Al igual que antes, se van a realizar las medidas tanto a corto plazo como a largo plazo, para comprobar el rendimiento del sistema. Lógicamente, por las noches bajará la temperatura, pero esos valores pueden ser descartados por no depender del número de dispositivos presentes.

Representación de un momento concreto

Por último, se va a realizar una fotografía de la situación del Mercado en un momento dado, para comprobar la afluencia en las distintas partes del Mercado. Para esta imagen se tendrán en cuenta los dispositivos presentes en el Mercado en un momento dado y se calcularán sus posiciones mediante el criterio de centroides. De esta manera se podrán ver las tendencias que sigue la gente a lo largo del tiempo en el Mercado. Se han elegido momentos clave en

los que se conocen alguna posición clave para poder comparar con momentos reales.

5.2. Resultados obtenidos

5.2.1. Dificultades encontradas tras el primer análisis de los datos

Una vez analizados los datos obtenidos tras las distintas pruebas, se han encontrado una serie de problemas que afectan al rendimiento del sistema. Estos problemas se han intentado resolver en la medida de lo posible, teniendo en cuenta las restricciones de tiempo presentes a la hora del desarrollo del proyecto.

El primer problema que se ha encontrado ha sido con los dispositivos de Apple con sistema operativo a partir de iOS8. En este sistema se ha comenzado a implementar una aleatorización de las direcciones MAC al enviar los mensajes de tipo Probe Request [25]. Esta aleatorización, sin embargo, solo se produce cuando se cumplen una serie de condiciones necesarias: el dispositivo no está conectado a WiFi y el dispositivo se despierta del modo Sleep. Esta medida de seguridad no es un problema ya que estas condiciones raramente se dan, ya que al modo Sleep solamente se entra cuando no existe ningún servicio activo en el teléfono y normalmente siempre existe algún proceso funcionando en segundo plano, como por ejemplo las notificaciones. Aún así, se propone como solución el uso del campo Probed SSIDS de los paquetes recibidos en el caso de que en próximas actualizaciones del sistema cambien las condiciones para la aleatorización. Este campo que envían los dispositivos contiene las IDs de las estaciones base a las que ya se ha conectado previamente el dispositivo y por lo tanto confía. Por lo tanto, en caso de que un dispositivo cambie de dirección MAC, se podría asumir que es el mismo dispositivo si comparte el campo Probed SSIDS.

Otro de los problemas que se han encontrado es el comentado anteriormente por el carácter aleatorio de los paquetes Probe Request. El problema se ha conseguido esquivar al no usar ese valor de tiempo entre Probe Request en el desarrollo de la herramienta. Bien es cierto que es imposible evitar el problema de que un dispositivo se encuentre dentro del Mercado y no envíe ningún paquete, pero no hay nada que se pueda hacer al respecto.

A lo largo del desarrollo, se ha intentado la captura de un mayor rango de paquetes, a fin de aumentar las muestras y poder aportar funciones adicionales. Por este motivo, se encontró el problema de que si se pretende capturar todos los paquetes presentes en la red, se produce una saturación, bien en el driver o bien en la memoria de la Raspberry Pi, que provoca que se pierdan paquetes en la ejecución del programa. Por este motivo, se ha decidido realizar un filtro

previo, por el que solamente se tendrán en cuenta los paquetes de tipo Probe Request. Además, al capturar paquetes de tipo Datos, aparecían problemas de privacidad, que se han preferido evitar.

Se ha descubierto que aparece un retardo en el procesado de los paquetes en las Raspberry Pi, sobre todo cuando se trata de la Raspberry Pi 1, por tener una baja capacidad de procesado. Tras realizar pruebas, se descubrió que el mayor retardo lo aportaban las operaciones relativas a la base de datos. Por esto, se decidió que esta tarea se relegaría al servidor y por lo tanto las Raspberry Pi solamente realizarían la parte de captación de datos.

Por último, se ha visto que al cabo de un tiempo, sin razón aparente, la captura realizada se para sin dar ningún error y el programa deja de funcionar. No se ha conseguido descubrir la razón de este problema, probablemente radique en saturación de la memoria interna de la tarjeta de red. Se ha decidido solucionar este problema mediante el chequeo mencionado en el capítulo 4, mediante el cual se reiniciará el programa en caso de que no se haya recibido nada en un tiempo dado.

5.2.2. Resultados de la validación

En esta sección se presentarán los resultados obtenidos a partir de las reglas comentadas anteriormente.

Conteo de personas

En primer lugar se comentarán los resultados obtenidos al comparar los valores reales medidos del número de personas con el número de dispositivos presentes en el Mercado en un momento dado. Se han realizado distintas gráficas en función del número mínimo de nodos que se van a requerir que detecten a un dispositivo para considerar que esté dentro. Se ha realizado la comparación para valores 4, 5, 6 y 7.

Se puede comprobar claramente en la figura 5.1 que entre ambos valores no existe correlación alguna. Esto se debe a la naturaleza aleatoria del hecho de que una persona lleve consigo un teléfono móvil con WiFi conectado y que este teléfono envíe un mensaje de tipo Probe Request en el intervalo que se ha medido. Por este motivo, resulta complicado inferir el número de personas a través del número de dispositivos que se estiman en un momento dado.

En segundo lugar, se van a realizar las gráficas del número de dispositivos detectados en un intervalo más largo, de un día y una semana. De esta manera, se podrá comprobar si los cálculos realizados tienen validez a la hora de evaluar las tendencias del medio. Para este cometido se ha decidido usar como valor umbral 4 nodos, por haberse comprobado que es el más preciso.

Se puede asumir que el cálculo de las tendencias es correcto, ya que las

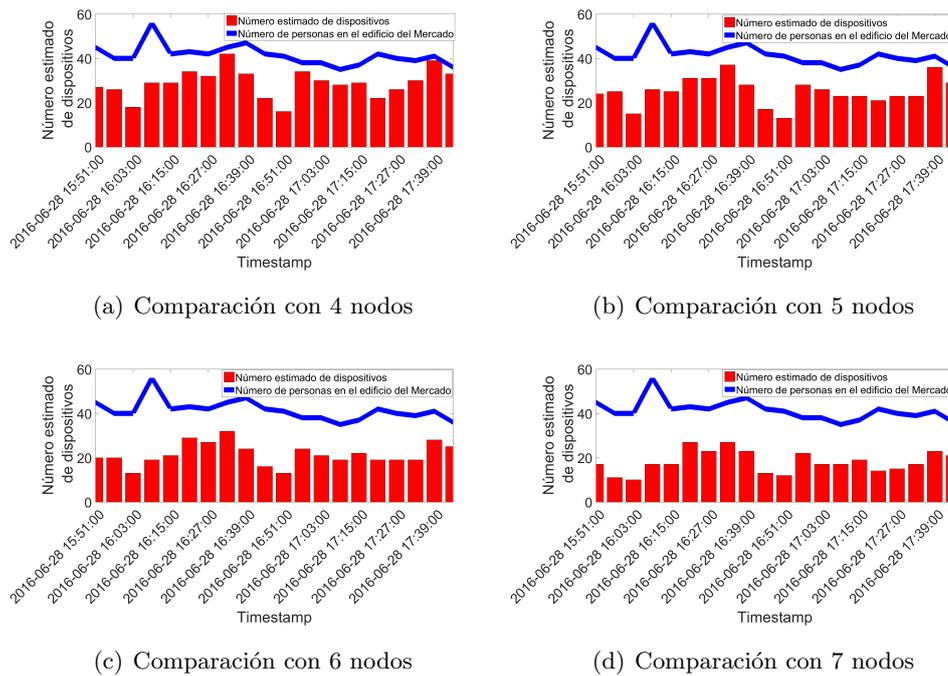


Figura 5.1: Número real frente a número estimado.

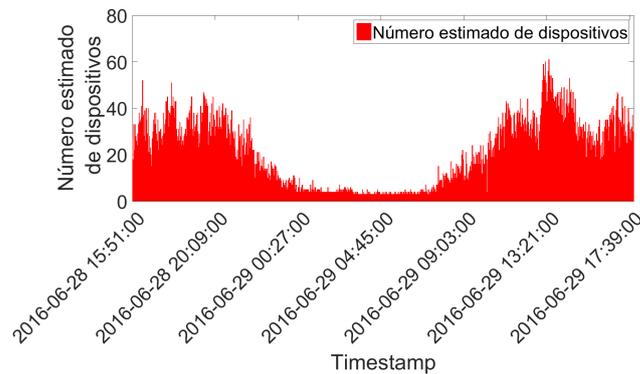


Figura 5.2: Dispositivos calculados a lo largo de un día.

tendencias que se siguen en la figura 5.2 y la figura 5.3 son las que se podrían esperar conociendo el terreno: baja el número de dispositivos por la noche a un mínimo (dispositivos pertenecientes a los comercios), aumenta a la hora de comer, baja otra vez sobre las 4 y vuelve a subir otra vez más tarde. Además, se ve que los domingos por la tarde hay poca gente, cosa que se puede ver si se va al Mercado una tarde similar.

Medidas de localización con el algoritmo de centroides

Se comentarán ahora los resultados obtenidos a partir de las medidas de localización. En primer lugar se presentarán los resultados obtenidos con el

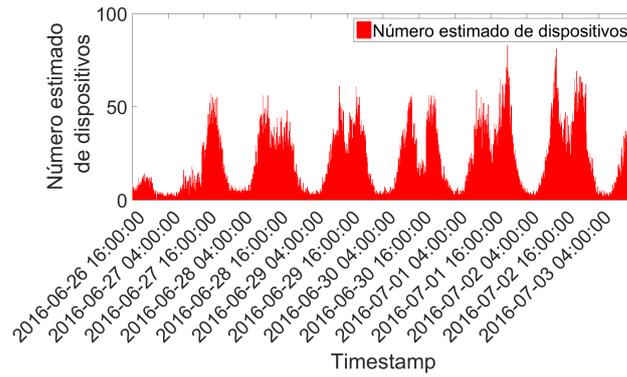


Figura 5.3: Dispositivos calculados a lo largo de una semana.

algoritmo de centroides.

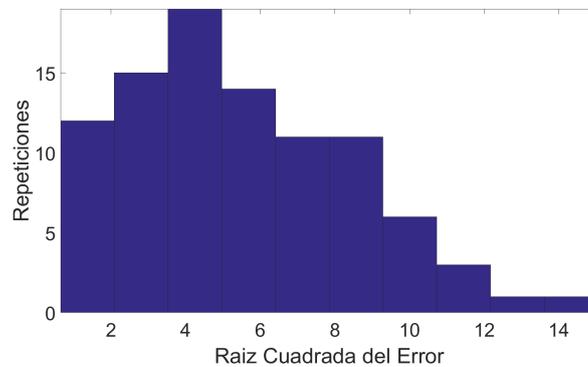


Figura 5.4: Histograma de los errores en las medidas con centroides usando 8 nodos medidores.

Puede verse en la figura 5.4 que la función de probabilidad de los errores se centra en 4.5 metros. En cambio, el error medio obtenido con estas medidas es de 5.7 metros. Este valor se considera correcto y dentro de un rango asumible.

En la figura 5.5 se representan algunos puntos reales frente a puntos estimados para tener una guía visual de la capacidad del sistema. Puede comprobarse que las medidas son bastante cercanas al punto real y que la varianza en los errores es bastante baja.

Por último, se ha realizado una captura de la situación medida en el Mercado en un momento dado, en el que se observó alguna característica del recinto para poder compararlo con los valores estimados. Se ha seleccionado un momento en el que se observó que solamente había una persona en el comedor, para poder comprobar si el funcionamiento es correcto. Se puede ver en la figura 5.6 que solamente se ha localizado a una persona en el recinto destinado al comedor, así que en esa parte el funcionamiento es correcto. No es posible comprobar las posiciones de las personas en todo el recinto, pues es imposible ver a todas las personas a la vez.

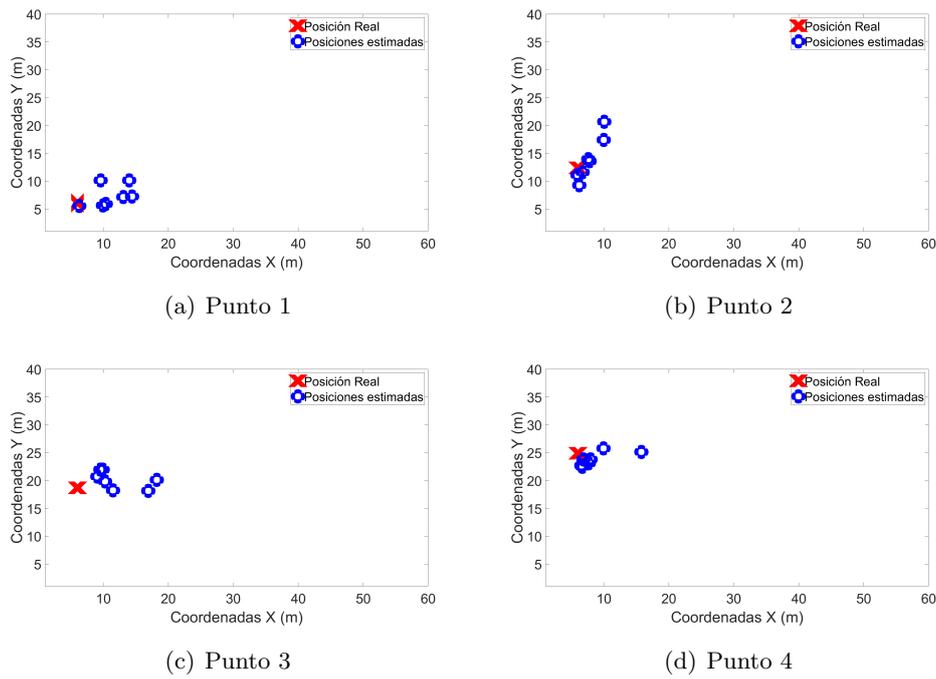


Figura 5.5: Puntos reales frente a puntos estimados con 8 nodos usando centroides.

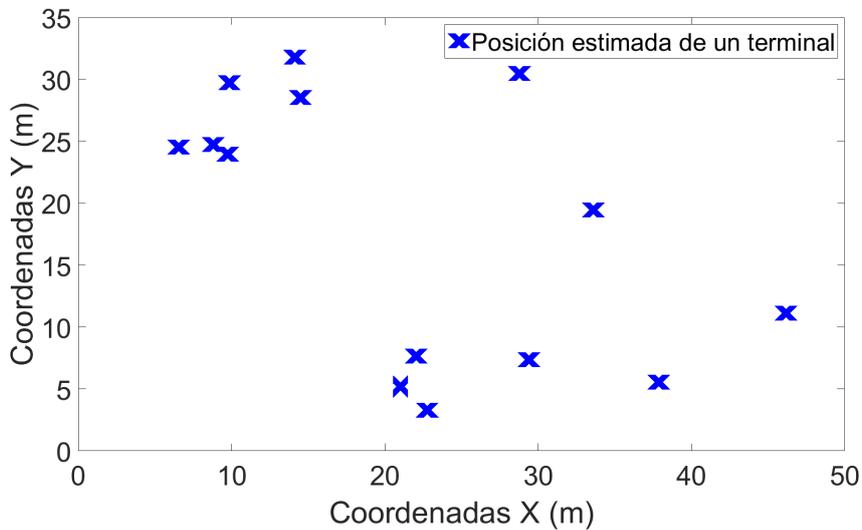


Figura 5.6: Representación de un momento dado utilizando centroides

Se procede a comentar los mismos resultados, utilizando solamente 4 nodos medidores.

En la figura 5.7 puede verse que la función de probabilidad de los errores se centra en 7.5 metros. En esta ocasión, el error medio que presenta el sistema es de 9.4 metros.

Se puede ver que el comportamiento del sistema empeora en gran medida

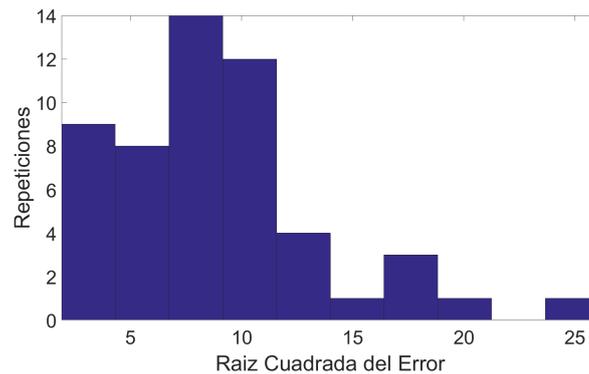


Figura 5.7: Histograma de los errores en las medidas con centroides usando 4 nodos medidores.

si se usan solamente 4 nodos. Sin embargo, el cambio no es tan notable como para hacer que el sistema se vuelva inutilizable.

Medidas de localización con el algoritmo de trilateración

Después de los centroides, se procede a analizar el comportamiento del algoritmo de trilateración.

De nuevo, los errores en las medidas se centran en torno a 7 metros. El error medio en este caso es de 8.5 metros. Queda claro que el rendimiento de este algoritmo es mucho peor que el de centroides. Se puede observar que con este algoritmo aparecen algunas medidas con un error muy grande, la varianza del sistema aumenta en gran medida. Esto se traduce en algunas medidas que distan mucho de la real. Además, debido a que su funcionamiento se basa en la resolución de sistemas de ecuaciones no lineales, presenta un tiempo de ejecución menor.

En la figura 5.10 se puede apreciar lo comentado anteriormente, hay algunas medidas que presentan un error muy grande.

De nuevo se ha realizado una captura de un momento dado, en el mismo momento que en el algoritmo de centroides. En este caso, ya no aparece el dispositivo que se había considerado en el comedor, así que en esa medida ha aparecido un error lo suficientemente grande como para sacarlo de esa zona.

En la figura 5.12 se puede observar que al disminuir el número de nodos medidores, aumenta el error, en este caso se centra en 10 metros. El error medio en este caso es de 9.1 metros. Se aprecia que el rendimiento del sistema sigue empeorando cuando se reduce el número de nodos medidores, aunque se puede considerar un error asumible.

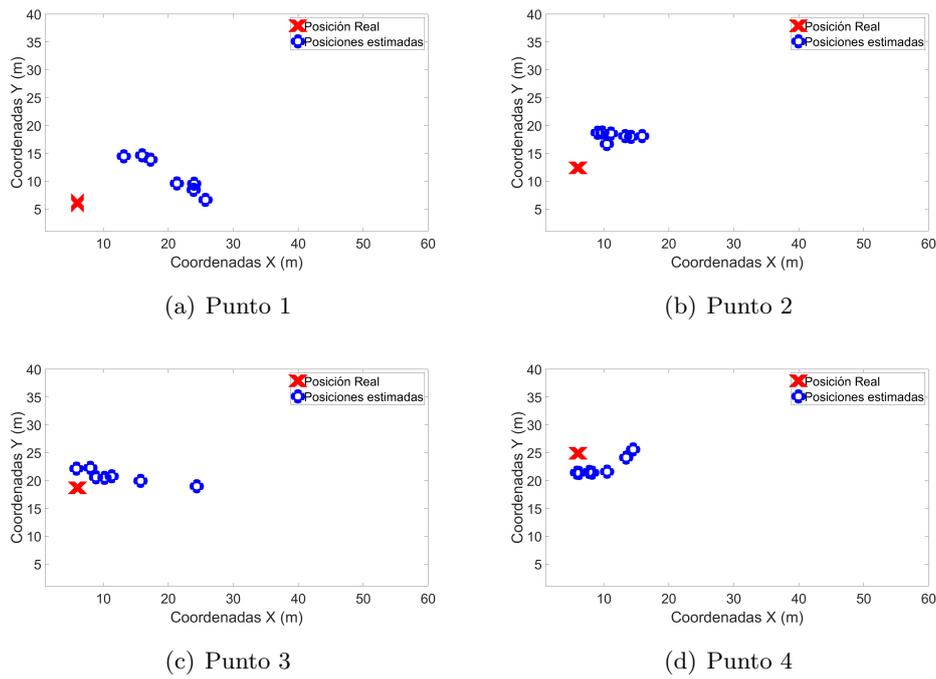


Figura 5.8: Puntos reales frente a puntos estimados con 4 nodos usando centroides.

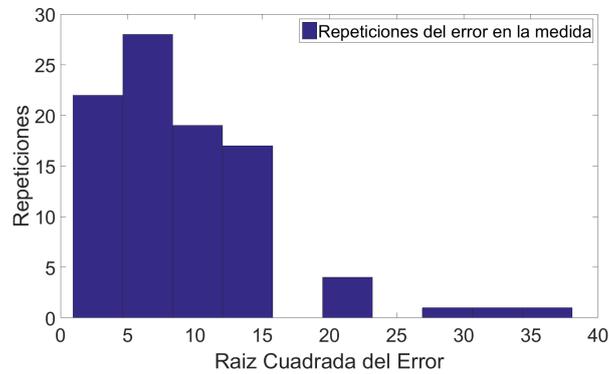


Figura 5.9: Histograma de los errores en las medidas con trilateración usando 8 nodos medidores.

5.2.3. Correlación entre número de dispositivos y temperatura/humedad

En este caso, de nuevo se realizarán las gráficas de número de dispositivos en función del parámetro de número mínimo de dispositivos medidores con valor 4.

Parece evidente que, a corto plazo, no existe una relación clara entre la temperatura, humedad y número de dispositivos presentes. Parece que, a priori, la influencia de los dispositivos no es suficiente para variar estos valores, que

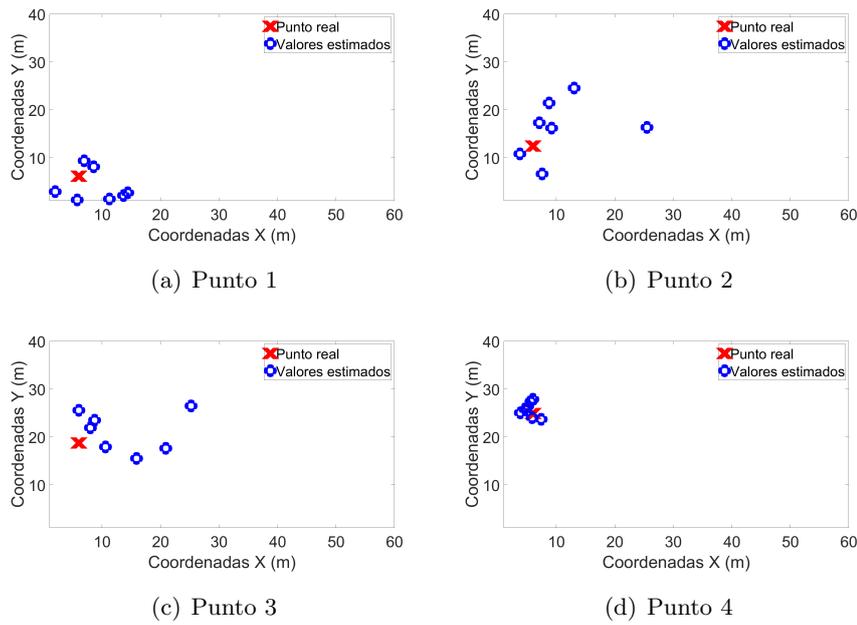


Figura 5.10: Puntos reales frente a puntos estimados usando trilateración con 8 nodos.

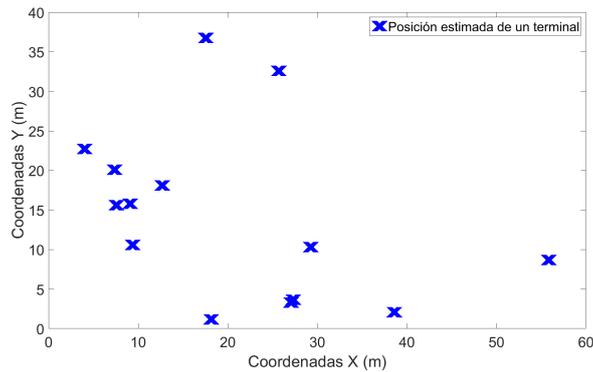


Figura 5.11: Representación de un momento dado usando trilateración.

varían por circunstancias ajenas al número de personas.

Se ha decidido también mostrar la temperatura en relación al número de dispositivos en un largo plazo. Se presenta a lo largo de una semana.

Se puede ver que, en una visión a largo plazo, sí que existe una relación entre el número de dispositivos estimados y la temperatura y humedad en el recinto. Lo que no está tan claro es si esa relación se establece porque los dispositivos influyen en la temperatura, o porque al variar la temperatura varían las costumbres de la gente, que entran más o menos al Mercado.

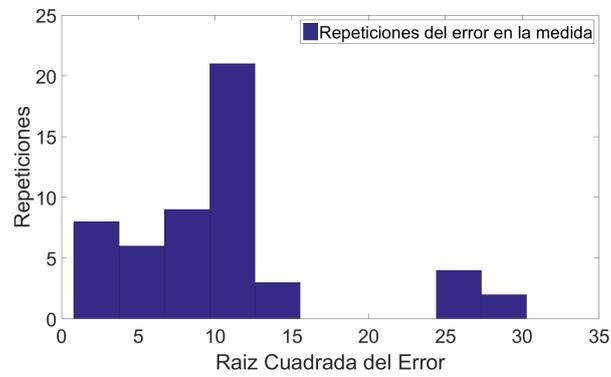


Figura 5.12: Histograma de los errores en las medidas con trilateración usando 4 nodos medidores.

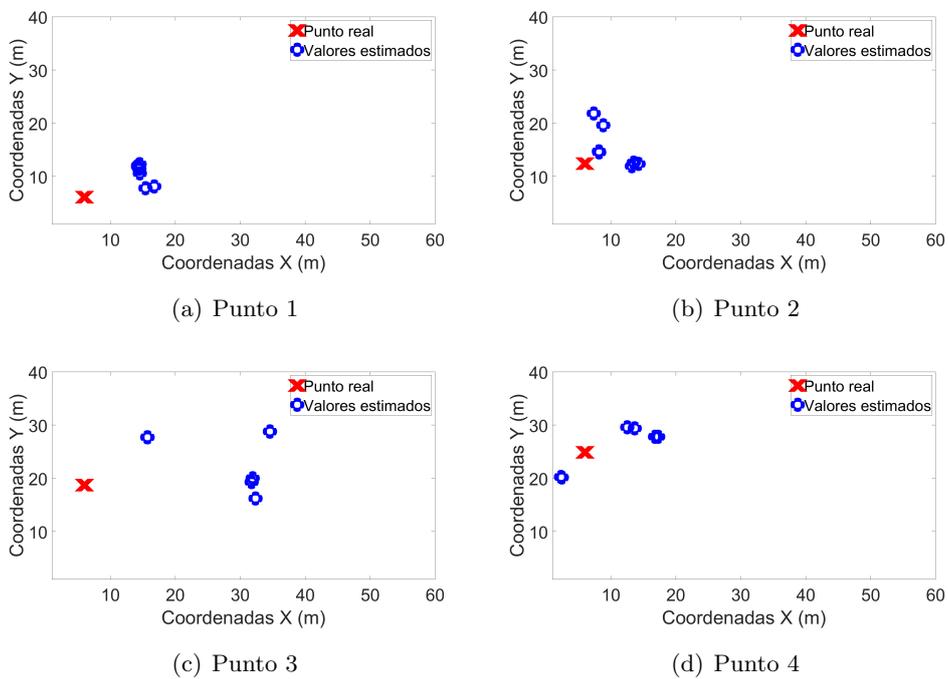


Figura 5.13: Puntos reales frente a puntos estimados usando trilateración con 4 nodos medidores.

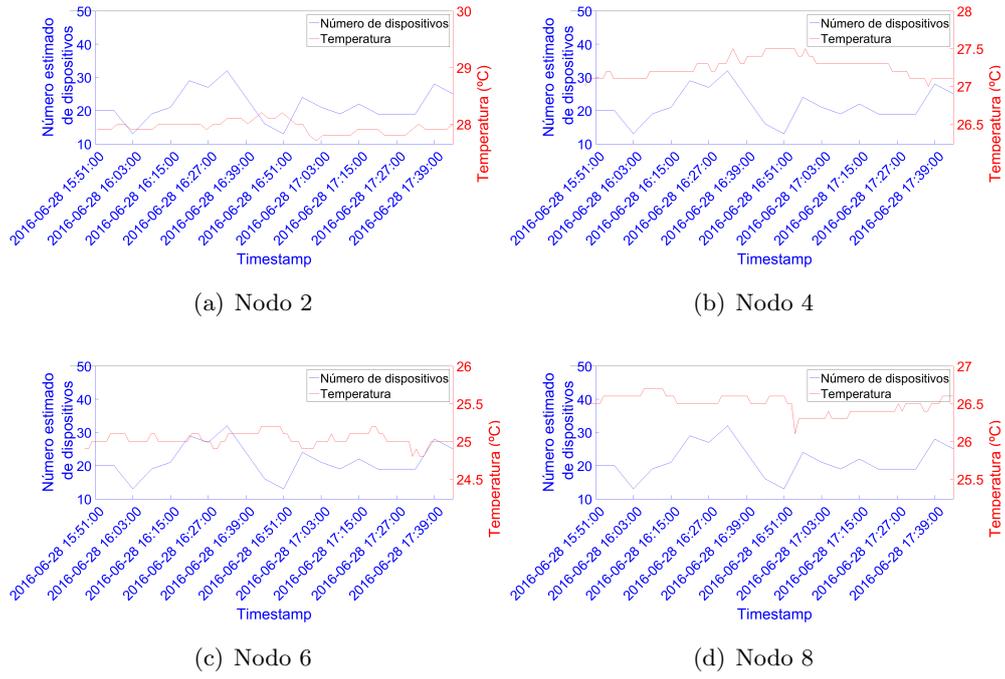


Figura 5.14: Comparación entre número de dispositivos y temperatura.

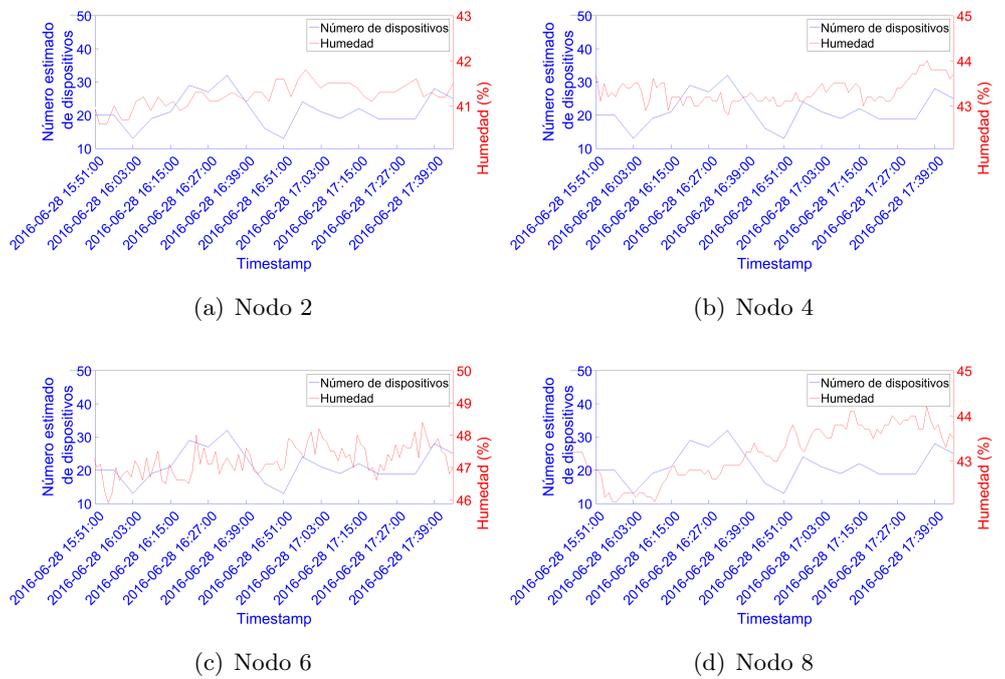
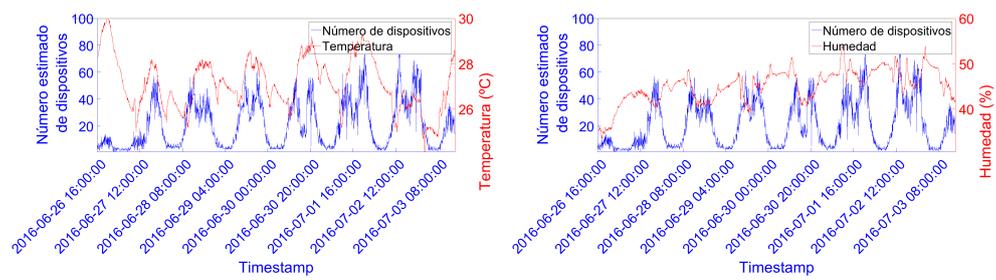


Figura 5.15: Comparación entre número de dispositivos y humedad.



(a) Comparación entre dispositivos y temperatura a lo largo de una semana (b) Comparación entre dispositivos y humedad a lo largo de una semana

Figura 5.16: Medidas de temperatura y humedad a lo largo de una semana.

Conclusiones y líneas futuras

Tras el desarrollo de este proyecto, se ha logrado confeccionar una herramienta para estimar la localización de dispositivos móviles en un recinto cerrado. A pesar de que las estimas generadas por el sistema presentan un error de algunos metros en las medidas, se considera que el resultado es satisfactorio. Para la realización de estos cálculos se han utilizado dos algoritmos distintos y se ha podido comparar el rendimiento de ambos. Tras analizar los resultados obtenidos, se ha podido comprobar que el algoritmo de centroides tiene un rendimiento mucho mejor que el algoritmo de trilateración. Además de tener una precisión mayor, se ha observado que el algoritmo de trilateración es mucho más lento debido a su funcionamiento. Por lo tanto, se concluye que para un funcionamiento en tiempo real, resultaría mucho más útil una implementación con el algoritmo de centroides.

Con respecto a la parte del conteo de personas, se ha podido comprobar que no es útil para buscar una solución a corto plazo, dado que hay que asumir que existe una relación determinista entre el número de dispositivos con el WiFi activado y el número de personas.. Sin embargo, se ha podido comprobar en base a los resultados que si se observan las tendencias a largo plazo se obtienen resultados coherentes. Esto se puede traducir en una estimación por parte de los comercios, observando las horas de mayor afluencia del Mercado y pudiendo considerar este conocimiento como base para modificar políticas o estrategias comerciales, por ejemplo, variando las ofertas ofrecidas o los horarios de trabajo.

Las conclusiones obtenidas a partir de los estudios de temperatura y humedad no son concluyentes. Aparentemente no existe una relación directa entre las medidas de temperatura y humedad y la cantidad y localización de los dispositivos detectados a través del sistema desarrollado. Sin embargo, cuando se realizan las medidas en un periodo largo, aparece una relación entre ambos valores. Es posible que esta relación no venga dada por la influencia del número de dispositivos sobre las condiciones ambientales, sino que una mayor

temperatura atraiga a más gente a pasear, o incluso que las horas de mayor temperatura coincidan con las horas de mayor afluencia de gente. Serían necesarias más medidas para poder comprobar esta relación.

Por último, se han analizado los resultados de la representación de un momento concreto. Esta representación se apoya en los dos algoritmos utilizados y, por lo tanto, los resultados que aparezcan aquí arrastrarán el error que introducen estos algoritmos. En las pruebas que se han realizado, los resultados guardan similitud con el entorno real en ese momento en el caso del algoritmo de centroides, no apareciendo la referencia requerida en el caso de la trilateración. Por tanto, se puede volver a observar que el funcionamiento usando el algoritmo de centroides es mejor que cuando se usa trilateración.

Como posible continuación de los desarrollos realizados y los resultados obtenidos durante este trabajo, se presentan soluciones y mejoras alternativas que se podrían aplicar y que no se han implementado porque extenderían demasiado el alcance del proyecto tal y como se concibió en un principio

- En primer lugar, se podría hacer uso de la tecnología Bluetooth, que está instalada pero no se ha implementado su funcionamiento. Esto presenta un problema añadido, ya que no es posible relacionar la tarjeta WiFi y Bluetooth presentes en un mismo dispositivo. Como solución a explorar, se propone que un dispositivo WiFi y uno Bluetooth se consideren como el mismo dispositivo físico cuando se encuentren durante un tiempo en posiciones próximas y con tiempos de permanencia similares. Esto, evidentemente, implica un análisis de las observaciones recogidas no exento de complejidad.
- Durante el desarrollo de esta herramienta ha aparecido un nuevo servicio de Google, un concepto conocido como la Physical Web [7]. Este sistema se basa en el envío de mensajes Bluetooth con comandos, que anuncian una determinada página web a los dispositivos cercanos. Estos dispositivos, si tienen la opción activada, reciben una notificación para poder acceder a la página anunciada. Esto tiene una aplicación clara de publicidad para los comercios presentes en el Mercado, que podrían anunciar productos u ofertas a los clientes. Una vez completado el objetivo del punto anterior sería interesante integrar este servicio (u otro análogo) en el sistema con el fin de incrementar sus funcionalidades.
- Otra idea que apareció durante el desarrollo del sistema fue el análisis de más tipos de paquetes que los de tipo Probe Request. De esta manera se podrían analizar, por ejemplo, los mensajes de tipo Beacon enviados por los routers o los paquetes de datos, lo que nos daría información adicional sobre la carga de la red. Debido a la limitación computacional de los dispositivos medidores empleados, en nuestro caso, este tipo de análisis

no se pudo llevar a cabo. El uso de dispositivos con mayor capacidad habilitaría este y otro tipo de análisis más complejo.

- Se ha planteado también la posibilidad de aplicar una estrategia de autocalibración de los nodos, en la que uno de ellos abandonaría el modo monitor para pasar a enviar paquetes de tipo Probe Request. De este modo, los otros nodos podrían captar estos paquetes y procesar la potencia recibida. Si esta potencia no fuera la esperada del nodo que emite, se estimaría que ha aparecido un obstáculo en el medio y se procedería a recalcular los modelos de propagación utilizados.
- No se han podido realizar aplicaciones de móvil o escritorio para este sistema por falta de tiempo. Sería interesante tener una manera de presentar los datos de una forma visual e interactiva para poder dar un uso sencillo a la plataforma desarrollada.
- Se ha estudiado la migración del sistema a placas Raspberry Pi 3 [9]. Esta nueva versión tiene características superiores a la que se usa actualmente, resultando en un aumento de la capacidad de procesamiento de los nodos. Aunque esta placa incluye una interfaz WiFi y otra Bluetooth, la interfaz WiFi no soporta el uso de modo monitor, por lo que seguiría siendo necesario una tarjeta de red adicional. Sin embargo, la interfaz embebida se podría usar en la estrategia de autocalibración descrita en uno de los puntos anteriores de manera continua.
- Por último, se propone que al calcular la localización de un dispositivo se tengan en cuenta los obstáculos presentes en el mapa y los comercios. Así, no sería posible situar un dispositivo sobre una pared o un comercio cerrado, teniendo en cuenta también los horarios de apertura y cierre de comercios.

Bibliografía

- [1] Accuware. URL: <https://www.accuware.com/>.
- [2] Aircrack-ng. URL: <https://www.aircrack-ng.org/>.
- [3] Dht-22. URL: <https://www.adafruit.com/product/385>.
- [4] Ekahau. URL: <http://www.ekahau.com/>.
- [5] Google location system. URL: <https://developer.android.com/guide/topics/location/index.html?hl=es>.
- [6] Página web de python. URL: <https://www.python.org/>.
- [7] Physical web. URL: <https://google.github.io/physical-web/>.
- [8] Raspberry pi 2. URL: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>.
- [9] Raspberry pi 3. URL: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>.
- [10] Scapy. URL: <http://www.secdev.org/projects/scapy/>.
- [11] Tp-link tl-wn722n. URL: <http://www.tp-link.com/ar/products/details/TL-WN722N.html>.
- [12] Tshark. URL: <https://www.wireshark.org/docs/man-pages/tshark.html>.
- [13] Zmq. URL: <http://zeromq.org/>.
- [14] Chema Alonso. La huella digital de tu conexión, jul 2016. URL: <http://www.elladodelmal.com/2011/09/la-huella-digital-de-tu-conexion.html>.

- [15] Yiu-Tong Chan ;Wing-Yue Tsui ; Hing-Cheung So ; Pak chung Ching. Time-of-arrival based localization under nlos conditions. Technical report, Dept. of Electron. Eng., City Univ. of Hong Kong, China, 2006.
- [16] Ditrendia. Informe mobile en españa y en el mundo 2015. Technical report, Ditrendia, 2015.
- [17] Alberto Gozalo Madrazo. Aplicación de sistemas de georreferenciación en edificios inteligentes. Technical report, Universidad de Cantabria, 2015.
- [18] Gomes Goncalo ;Sarmiento Helena. Indoor location system using zigbee technology. Technical report, Inst. Super. Tecnico, Inesc-ID, Lisbon, Portugal ;, 2009.
- [19] Martina Higuera Fuentes. Sistemas de geolocalización wifi. Technical report, Universidad de Cantabria, 2010.
- [20] IEEE. Standard ieee 802.11, 2012. URL: <http://standards.ieee.org/about/get/802/802.11.html>.
- [21] ITU-R. Propagation data and prediction models for the planning of indoor radiocommunication systems and radio local area networks in the frequency range 900 mhz to 100 ghz. Technical report, ITU Radiocommunication Assembly, 1997.
- [22] D. E. Manolakis. Efficient solution and performance analysis of 3-d position estimation by trilateration. Technical report, Hellenic Air Force Acad., Athens, Greece, 1996.
- [23] F. Lassabe ; P. Canalda ; P. Chatonnay ; F. Spies more authors. A friis-based calibrated model for wifi terminals positioning. Technical report, Lab. d'Informatique, Univ. de Franche, Montbeliard, France, 2005.
- [24] Rong P; Mihail L. Sichitiu. Angle of arrival localization for wireless sensor networks. Technical report, Dept. of Electr. & Comput. Eng., North Carolina State Univ., Raleigh, NC, 2006.
- [25] ZEBRA. Analysis of ios 8 mac randomization on loca tioning. Technical report, ZEBRA TECHNOLOGIES, 2015.