

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



TRABAJO DE FIN DE GRADO

OBTENCIÓN DE LA INFORMACIÓN DEL ESTADO DEL CANAL(CSI) EN REDES WIFI HACIENDO USO DEL MICROCONTROLADOR ESP32

Titulación: Grado en Ingeniería en Tecnologías de la Telecomunicación

Mención: Telemática

Autor: Paula Casteres Báez

Tutor: Itziar Goretti Alonso González

Cotutor: David de la Cruz Sánchez Rodríguez

Fecha: Septiembre 2022

Resumen

Los sistemas de localización han proporcionado la posibilidad de ofrecer servicios a los usuarios a partir de sus posiciones. En exteriores, la localización está resuelta con el uso de GNSS (Global Navigation Satellite System), sin embargo, estos sistemas no funcionan en interior debido a que no hay visión directa con los satélites. Debido a esto, existe un interés en investigar en este tema. Para la localización en interiores se han aplicado diferentes estándares inalámbricos tales como WiFi, ZigBee, RFID, BLE, entre otros. Es conocido que hay gran despliegue de las redes WiFi en el interior de los edificios y por lo tanto resulta ser una tecnología para utilizar. Dentro de las diferentes técnicas que se utilizan en la localización en interiores, se encuentra los basados en *fingerprinting* los cuales están basados en el uso de la señal RSS (*Received Signal Strength*) o en la información del estado del canal (*CSI, Channel State Information*).

En este proyecto se propone la obtención del valor de la información del estado del canal (CSI) extraído utilizando un microcontrolador de la familia de ESP32. Posteriormente estos datos serán analizados mediante el empleo de redes neuronales como sistema de predicción de localización, con el objetivo de evaluar los datos obtenidos.

Abstract

Location systems have provided the possibility to offer services to users from their positions. Location outdoors is resolved with the use of GNSS (Global Navigation Satellite System), however, these systems do not work indoors because there is no direct line of sight with the satellites. That's the reason why there is an interest in researching this topic. For indoor location different wireless standards such as WiFi, ZigBee, RFID, BLE and others have been applied. It is known that there is great deployment of WiFi networks inside the buildings and therefore it turns out to be a technology to use. Among the different techniques used in indoor localization, there are those based on fingerprinting which are based on the use of the RSS signal (Received Signal Strength) or on channel status information (CSI, Channel State Information).

This project aims to obtain the value of canal status information (CSI) extracted using a microcontroller of the ESP32 family. Then, these data will be analyzed using neural networks as a location prediction system, with the aim of evaluating the data obtained.

Índice

| | |
|--|----|
| Resumen | 7 |
| Abstract | 8 |
| Índice de ilustraciones | 11 |
| Índice de tablas..... | 12 |
| Acrónimos | 13 |
| Capítulo 1. Introducción..... | 15 |
| 1.1 Antecedentes | 15 |
| 1.2 Objetivo | 17 |
| 1.3 Petitionarios | 18 |
| 1.4 Estructura del documento..... | 18 |
| Capítulo 2. Conceptos básicos | 21 |
| 2.1 Indicador de fuerza de la señal recibida | 21 |
| 2.2 Información del estado del canal..... | 22 |
| 2.3 IEEE 802.11 | 23 |
| 2.4 Multiple Input-Multiple Output(MIMO) | 24 |
| 2.5 OFDM | 25 |
| 2.6 Machine Learning | 25 |
| 2.7 Redes neuronales..... | 26 |
| Capítulo 3. Recursos del entorno de medidas | 29 |
| 3.1 Recursos hardware | 29 |
| 3.1.1 Microcontrolador ESP32..... | 29 |
| 3.1.2 Linksys | 32 |
| 3.2 Recursos software | 33 |
| 3.2.1 ESP-IDF | 33 |
| 3.2.2 GitHub..... | 34 |
| 3.2.3 Matlab..... | 39 |
| 3.2.4 Lenguaje de programación: Python..... | 39 |
| Capítulo 4. Toma de medidas..... | 43 |
| 4.1 Planteamiento | 43 |
| 4.2 Toma de medidas | 45 |
| Capítulo 5. Análisis de datos..... | 49 |
| 5.1 Introducción | 49 |
| 5.2 Análisis de los datos obtenidos en las medidas..... | 49 |
| 5.3 Librería de redes neuronales de Matlab | 54 |

| | |
|---|----|
| 5.4 Experimentos..... | 55 |
| Capítulo 6. Conclusiones y líneas futuras. | 63 |
| Bibliografía..... | 65 |
| Presupuesto..... | 69 |
| P.1 Recursos materiales..... | 69 |
| P.2 Tarifa por tiempo empleado. | 70 |
| P.3 Material Fungible. | 71 |
| P.4 Redacción de la documentación..... | 71 |
| P.5 Derechos de visado del COITT..... | 72 |
| P.6 Gastos de tramitación y envío. | 73 |
| P.7 Aplicación de impuestos y coste total..... | 73 |

Índice de ilustraciones

| | |
|---|----|
| Ilustración 1. Perceptrón multicapa [25] | 27 |
| Ilustración 2. Bloques funcionales SoC ESP32 | 31 |
| Ilustración 3. Punto de acceso Linksys WRT 1900 ACS..... | 32 |
| Ilustración 4. Menú de configuración passive..... | 34 |
| Ilustración 5. Acceder a menú de configuración..... | 35 |
| Ilustración 6. Pantalla de configuración ESP32 | 36 |
| Ilustración 7. Configuración de la tasa de baudios..... | 36 |
| Ilustración 8. Configuración del canal de salida | 37 |
| Ilustración 9. Establecer extracción de datos CSI | 37 |
| Ilustración 10. Establecer el canal..... | 38 |
| Ilustración 11. Modificación de la frecuencia..... | 38 |
| Ilustración 12. Datos tomados en pasillo | 40 |
| Ilustración 13. Vector CSI ejemplo en pasillo | 41 |
| Ilustración 14. Esquema de toma de medidas | 43 |
| Ilustración 15. Toma de medidas en laboratorio | 44 |
| Ilustración 16. Toma de medidas en el pasillo | 45 |
| Ilustración 17. Flashear ESP32 | 45 |
| Ilustración 18. Ejecutar scripts | 46 |
| Ilustración 19. Toma de medidas en laboratorio | 47 |
| Ilustración 20. Ubicación rúter pasillo | 47 |
| Ilustración 21. Toma de medidas en el pasillo | 48 |
| Ilustración 22. Fichero de datos capturados por el ESP32..... | 50 |
| Ilustración 23. Amplitud (a) y fase (b) de 100 medidas obtenidas..... | 52 |
| Ilustración 24. Ejemplo de 10 entradas ya normalizadas a la red neuronal | 53 |
| Ilustración 25. Estructura de la red neuronal del experimento 1 y 2..... | 55 |
| Ilustración 26. Estudio del número de nodos de la capa oculta experimento 1 | 56 |
| Ilustración 27. Estudio del número de nodos de la capa oculta experimento 2. | 57 |
| Ilustración 28. Comparación de aciertos del experimento 1 y experimento 2 | 58 |
| Ilustración 29. Estructura de la red neuronal para el experimento 3..... | 59 |
| Ilustración 30. Estudio del número de nodos de la capa oculta experimento 3 | 60 |
| Ilustración 31. Estudio del número de nodos de la capa oculta experimento 4 | 61 |

| | |
|---|----|
| Ilustración 32. Estudio del número de nodos de la capa oculta experimento 5 | 62 |
|---|----|

Índice de tablas

| | |
|--|----|
| Tabla 1. Equivalencias de cobertura RSSI [12] | 21 |
| Tabla 2. Características rúter WRT 1900 ACS [33] | 33 |
| Tabla 3. Estructura de datos según fabricante ESP32 | 51 |
| Tabla 4. Tabla de amortización. | 70 |
| Tabla 5. Cálculo del presupuesto para la redacción del documento | 71 |
| Tabla 6. Presupuesto <i>P1</i> | 72 |
| Tabla 7. Coste total..... | 73 |

Acrónimos

CSI: Channel State Information

COITT: Colegio Oficial de Ingenieros Técnicos de Telecomunicaciones

DSS: Direct Sequence Spread Spectrum

IEEE: Institute of Electrical and Electronics Engineers

GPS: Global Positioning System

MIMO: Multiple-Input Multiple-Output

NFC: Near-Field Communication

OFDM: Orthogonal Frequency Division Multiplexing

RSSI: Received Signal Strength Indicator.

SoC: System on a Chip

SDG: Stochastic Gradient Descent

TFG: Trabajo de Fin de Grado

TCP/IP: Transmission Control Protocol/Internet Protocol

UART: Universal Asynchronous Receiver-Transmitter

Capítulo 1. Introducción.

1.1 Antecedentes

En cuanto a tecnologías de localización es comúnmente conocido el GPS, un sistema de posicionamiento global que proporciona a sus usuarios información sobre navegación, posición y cronometría. Éste es empleado para sistemas de localización en exteriores y, se ha comprobado que, el GPS [1] no funciona correctamente para obtener geolocalización en espacios interiores debido a que no posee la capacidad de penetrar en los materiales de construcción.

Con el paso de los años, [2] los dispositivos móviles se usan cada vez más por lo que es posible afirmar que, nos acompañan en nuestro día a día. Aprovechando este hecho, ha surgido la necesidad de desarrollar e implantar sistemas de localización de objetos o de personas en interiores con la finalidad de obtener la ubicación en interiores. Esto abre un campo de nuevas aplicaciones ya que permite enviar mensajes personalizados, avisos e incluso es posible aumentar la productividad.

Estos sistemas que, actualmente se encuentran en pleno auge, utilizan tecnologías como el WiFi, NFC, beacons o Bluetooth. En este TFG nos centraremos en los sistemas que utilizan tecnologías WiFi. Dado el potencial que ofrecen las redes Wi-Fi y que su infraestructura ya está estandarizada, por ello cada vez es más común utilizarla para la localización en interiores. Se ha demostrado que, al aprovechar el funcionamiento de estos sistemas es posible analizar el comportamiento de la señal inalámbrica y establecer relación de ubicación según la potencia de la señal. Además de que es una de las opciones más baratas que brinda gran alcance y, a su vez, es sencillo de configurar. Una de sus características a destacar es la cantidad de datos que transmite, siendo mayor frente a otras tecnologías. [5][6].

Se han realizado numerosas investigaciones sobre la localización en interiores basada en radiofrecuencia, para las que se suele utilizar con mayor frecuencia, el indicador de intensidad de la señal recibida (RSSI), que representa el nivel de potencia recibida por el receptor de la transmisión, debido a su baja complejidad computacional y a su gran aplicabilidad

Otra de las líneas de trabajo recientes en el ámbito de la localización en interiores, es la de utilizar el valor CSI (Channel State Information) como dato de entrada ya que ofrece información sobre la atenuación de amplitud y del cambio de fase de la señal inalámbrica [5][6].

Se ha comprobado que existen varias razones por las que el valor de RSSI no es adecuado para determinar la localización en interiores. El RSSI se mide a partir de la señal a nivel de paquete, por lo que, obtener un valor preciso es bastante complicado. Además, a medida que se va aumentando la distancia desde la fuente, el valor de RSSI disminuye debido al multitrayecto. Esto se debe a que la señal recibida puede ser directa o indirecta, obteniendo así valores inexactos como resultado. En conclusión, no es viable instaurar una relación entre la potencia recibida y la distancia de separación, cuando se requiere mayor precisión en la localización.

El CSI ofrece características del enlace de comunicación entre transmisor y receptor usando las subportadoras de un sistema de comunicación OFDM y, además de representar la atenuación de la señal inalámbrica durante la propagación de esta, también describe la dispersión y la refracción con la que se encuentra la señal durante la transmisión. El valor de información del estado del canal (CSI) es medido en la capa física, dentro del modelo OSI de una red inalámbrica y, como este contiene valores de amplitud y fase del canal en cada subportadora, se puede utilizar como huella dactilar para mejorar la precisión de un sistema de localización en interiores.

Por tanto, el CSI es un valor detallado de las propiedades del canal físico que ayuda a estimar las propiedades en un enlace de comunicación inalámbrico. La información del estado del canal es una alternativa frente al uso del valor RSSI, debido a que proporciona información de fase y amplitud de la subportadora multicanal. De esta forma, se puede hacer una descripción más exacta de la ruta de propagación de la señal. Además, para cada paquete, se obtienen múltiples valores de CSI mientras que, de RSSI, se obtiene un valor único.

A partir de aquí, se puede construir un mapa general o huella en el que se relacione cada ubicación con la señal inalámbrica. Para ello, el uso de CSI es lo más recomendable para

mejorar el rendimiento del sistema. Y, así plantear un sistema de localización en interiores más preciso.

En los últimos años, debido al desarrollo de las IoT, se ha iniciado una revolución de mercado de bajo coste en el que se emplean microcontroladores en determinados proyectos de investigación. Estos dispositivos son capaces de captar señales e incluso transmitir las para su posterior procesamiento. Su característica que produce un mayor interés es que, permiten realizar proyectos sin la necesidad de emplear demasiados recursos, lo que los hace más económicos y permiten sensorizar espacios de interiores y exteriores. Por esta razón, en este proyecto se va a realizar el estudio de cómo captar valores CSI empleando un microcontrolador de la familia ESP32.

En este TFG también se utilizarán técnicas de Machine Learning para analizar los datos obtenidos de CSI, con el fin de evaluar un posible sistema de localización. Cabe destacar que, la precisión de la ubicación es posible mejorarla si se aumenta el número de receptores o puntos de medidas que definen el mapa y sus antenas.

1.2 Objetivo

El objetivo de este proyecto consiste en el procesamiento y la posterior evaluación de la captura de la señal CSI mediante un microcontrolador ESP32, un dispositivo de bajo coste, para su utilización en localización en interiores. Este objetivo general se desglosa en los siguientes objetivos específicos:

- O1: Estudio de la señal CSI
- O2: Estudio y configuración de ESP32 para la captura de este tipo de señales
- O3: Realización de mediciones
- O4: Evaluación de las medidas realizadas

1.3 Peticionarios

Como solicitante del proyecto está la Escuela de Ingeniería de Telecomunicación y Electrónica (EITE), para así cubrir los créditos y requerimientos de la asignatura Trabajo de Fin de Grado, contemplada en el plan de estudios del Grado en Ingenierías en Tecnologías de la Telecomunicación (GITT). La elaboración de este trabajo se realiza dentro de la División de Redes y Servicios Telemáticos del Instituto Universitario para el Desarrollo Tecnológico y la Innovación en Comunicaciones (IDeTIC), de la Universidad de Las Palmas de Gran Canaria (ULPGC).

Se desarrolla este trabajo fin de grado con el objetivo de analizar el estado de información del canal de redes inalámbricas basadas en el estándar IEEE 802.11 para ofrecer servicios basados en localización con el uso de un microcontrolador ESP32.

1.4 Estructura del documento

Además de este primer capítulo introductorio en el que se explican los antecedentes y los objetivos a cumplir en este proyecto el resto del informe se estructura de la siguiente manera:

Capítulo 2, “Conceptos básicos”. En este capítulo se describen los aspectos básicos teóricos de las tecnologías que serán empleadas, así como sus características y los conceptos teóricos del proyecto, como lo son el CSI, WiFi, machine learning, etc.

Capítulo 3, “Estudio del entorno de desarrollo”. En este tercer capítulo se describen las herramientas y recursos utilizados para la realización de este proyecto.

Capítulo 4, “Toma de medidas”. Aquí se explicará la manera en la que se ha configurado el entorno de trabajo, así como en qué lugares se tomaron las medidas deseadas y cómo se realizó esta toma de medidas.

Capítulo 5, “Análisis de datos”. En este capítulo se realizará un estudio de los datos obtenidos, así como, un análisis de estos.

Capítulo 6, “Conclusiones y líneas futuras”. En este último capítulo de este proyecto se recogerán las conclusiones a las que se han llegado a partir del capítulo anterior.

A continuación, se encuentra la “Bibliografía”, en esta se encuentran las referencias utilizadas para la realización de este proyecto.

Finalmente encontraremos el presupuesto estimado realizado para la realización de este TFG.

Capítulo 2. Conceptos básicos

Para poder comenzar con el desarrollo técnico de este proyecto, primero se describirán algunos conceptos básicos necesarios para poder comprender lo que se va a desarrollar finalmente. Se tratarán conceptos como son RSSI, CSI, el estándar IEEE, MIMO, el OFDM como técnica de modulación, el Machine Learning y las redes neuronales.

2.1 Indicador de fuerza de la señal recibida

El indicador de fuerza de la señal recibida, RSSI por sus siglas en inglés, consiste en un término que se utiliza para medir el nivel de potencia de las señales recibidas en las comunicaciones inalámbricas. Este indica la calidad con la que se recibe una señal en un dispositivo cualquiera. La interpretación de este valor [12] y, para analizar el nivel de cobertura de la señal, se utilizan los valores que aparecen en la tabla 1.

| | RSSI (dBm) |
|--------------------|-------------|
| Excelente | >-76 |
| Muy buena | -89 a -77 |
| Buena/Media | -97 a -90 |
| Baja cobertura | -103 a -98 |
| Muy baja cobertura | -112 a -104 |
| Muy poca cobertura | -113 a -132 |
| Sin cobertura | -135 |

Tabla 1. Equivalencias de cobertura RSSI [12]

Muchos de los trabajos de localización en interiores utilizan como señal de referencia para la localización, la señal RSSI. Sin embargo, en [13] los sistemas de localización basados en el indicador de fuerza de la señal recibida, se ha demostrado que se pueden ver afectados por la variación temporal y espacial. Esto se debe a que se ha observado que estos sistemas tienen un rendimiento de posicionamiento deficiente, ya que el RSSI varía debido a las fluctuaciones de potencia de la señal tanto en el tiempo como en el espacio.

Aunque se puede afirmar, según [14], que se puede establecer un modelo para estimar la ubicación de un objeto frente a la fuente, utilizando la potencia recibida, existen varias razones por las que se ha demostrado que un sistema de localización en interiores basado en este valor no es viable.

Estas son, según [14]: en primer lugar, el valor de RSSI se mide a partir de la señal de radiofrecuencia a nivel de paquete, por lo que obtener un valor que sea preciso es bastante complicado. En segundo lugar, este valor se ve afectado por el efecto de trayectos múltiples. Como sabemos, la propagación de una onda de radiofrecuencia se atenúa por la reflexión cuando esta choca con la superficie de cualquier obstáculo. Por ello, es posible afirmar que el efecto multitrayecto es más intenso en entornos interiores.

De esta forma según [14] se concluye que, debido a estas razones parece que no es posible establecer una relación entre el RSSI y la distancia de separación con la fuente.

Como se ha demostrado que el valor de CSI suprime las desventajas ofrecidas por el valor de RSSI, en este proyecto se ha decidido que se utilizará la información del estado del canal como señal de referencia para localización.

2.2 Información del estado del canal

En los sistemas OFDM, el ancho de banda disponible se divide en múltiples subportadoras ortogonales y los datos se transmiten en esas subportadoras, usando por tanto diferentes frecuencias. La amplitud y fase de cada subportadora se ven afectadas por efectos de multitrayecto, desvanecimientos y dispersión. La presencia de objetos, personas o movimientos afecta directamente en la amplitud y fase de cada subportadora. La información del estado del canal, [15] CSI por sus siglas en inglés, ayuda a realizar una estimación sobre las propiedades del canal de una señal inalámbrica, ya que recoge el comportamiento de la señal inalámbrica entre un transmisor y un receptor. Este es un valor detallado de la capa física que proporciona información sobre cómo se propagan las señales por medio de un canal con sus diversos efectos, el retardo, la atenuación de la amplitud y el cambio de fase para cada subportadora.

El CSI se puede extraer como vector o matriz, en función de la cantidad de datos que se obtengan. Cabe destacar que, la obtención de la información del estado del canal está limitada debido a que las condiciones del canal se alteran de manera dinámica.

La expresión matemática (1) que lo representa es la siguiente:

$$H_i = |H_i|e^{j\sin(\angle(H_i))} \quad (1)$$

Donde $|H_i|$ y $\angle H_i$ representa la amplitud y fase de la subportadora i , respectivamente.

2.3 IEEE 802.11

A principios de la década de 1990 el Instituto de Ingenieros Eléctricos y Electrónicos creó un comité ejecutivo que fue añadido al estándar 802 para redes de área local y metropolitanas, debido a que era necesaria una comunicación inalámbrica estándar de infraestructura con el fin de crear un nicho de mercado deseable. En este momento, desarrollar una red inalámbrica era algo muy costoso, lento, o carecía de disponibilidad. Por esta razón, el comité 802.11 se centró en proporcionar una solución económica, confiable, rápida y robusta para las comunicaciones inalámbricas que pudiera crecer dentro de un estándar y que tuviera mucha aceptación [17].

En sus inicios, únicamente se centraban en la interoperabilidad, ya que su objetivo principal consistía en proporcionar puentes entre varios medios, hasta encontrar nuevas técnicas y espectros de radiofrecuencia para mejorar la compatibilidad con nuevas aplicaciones.

Por ello, se han ido modificando a lo largo de los años con el fin de proporcionar nuevos servicios, así como las capacidades para expandir las necesidades inalámbricas y abordar las deficiencias del estándar original.

Dentro de este estándar, se han creado normas que han ido evolucionando y mejorando a lo largo de los años, a continuación, se describirán algunas de estas: [17]

- IEEE 802.11a: En este caso trabaja en un espacio de frecuencia de 5GHz y ofrece velocidades de transmisión de 54 Mbps. Su mayor inconveniente es que sufre cambios que son provocados por las interferencias que recibe, lo que provoca que la calidad de la red disminuya y, además lo vuelva incompatible con otros estándares. Requiere de una modulación OFDM.

- IEEE 802.11b: Este consiste en el estándar más ampliamente conocido como WiFi. Trabaja en la banda de frecuencias de 2.4 GHz y ofrece una velocidad de transmisión de hasta 11Mbps. Emplea modulación DSSS.
- IEEE 802.11g: Ofrece una transmisión de 2.4 GHz soportando velocidades de 20 hasta 54 Mbps. Cabe destacar que, es compatible con el estándar IEEE 802.11b y emplea modulación de DSS y OFDM.
- IEEE 802.11n: Se estableció en 2008, pero se empezó a utilizar en 2009. Este puede trabajar en un rango de frecuencias de 2,4GHz hasta 5Ghz y logra una velocidad de transmisión de hasta 600Mbps. Además, utiliza la tecnología de Múltiples Entradas y Múltiples Salidas (MIMO). Esto es una ventaja que posee, debido a que esta tecnología ofrece más confianza que la tecnología WiFi y, además es necesaria para la realización de este proyecto. Esto se debe a [18] que el valor de CSI se determina realizando un análisis de los paquetes recibidos con secuencias de entrenamiento en el encabezado de estos. Además, debido a que OFDM es capaz de habilitar la interfaz de múltiples entradas y múltiples salidas (MIMO), los dispositivos que actúan como receptores y además poseen múltiples antenas son capaces de extraer datos de CSI de alta dimensión para cada paquete de la red inalámbrica, utilizando 802.11n.

2.4 Multiple Input-Multiple Output (MIMO)

Entre el transmisor y el receptor se pueden utilizar múltiples antenas, este tipo de sistema es a lo que se le llama ‘MIMO’[19], un sistema que ofrece las ventajas proporcionadas por la diversidad espacial, dentro de un entorno con desvanecimiento de multitrayecto.

Las configuraciones utilizadas en la implementación de un sistema MIMO consiste en presentar ganancia de diversidad, con lo que se consigue combatir el desvanecimiento de la señal inalámbrica y, de esta forma obtener ganancia en capacidad del sistema.

2.5 OFDM

OFDM [21] es una técnica de modulación que emplea distintas portadoras ortogonales que se encuentran sobrepuestas. Esta, se ha convertido en un método muy popular para la transmisión de señales de banda ancha sobre canales inalámbricos. Esta técnica de modulación se utiliza como solución al desvanecimiento de la frecuencia, así como para las distorsiones multitrayecto que se pueden encontrar en canales de banda ancha.

Cabe destacar que, una base de datos que contiene información del estado del canal (CSI) que ha sido extraída de un paquete basado en un canal de multiplexación de diversidad de frecuencia ortogonal proporciona información para localizar a un dispositivo transmisor mediante el uso de técnicas de aprendizaje automático como pueden ser las redes neuronales o árboles de decisión, entre otras.

2.6 Machine Learning

Actualmente, los algoritmos de Machine Learning [22][23] son cada vez más utilizados para analizar conjuntos de datos y de esta forma crear sistemas para tomar decisiones ante problemas para los que proponer una solución algorítmica no es viable.

Los algoritmos de Machine Learning tienen como cometido aprender a tomar decisiones o, incluso, realizar predicciones basadas en datos. Estos se clasifican a razón de las siguientes características: retroalimentación, propósito y método.

La retroalimentación [24] se tiene en cuenta cuando se desea entrenar a un algoritmo y se necesita para que la calidad del modelo pueda ir mejorando. Existen varios tipos de retroalimentación:

- Aprendizaje supervisado [24]: en este caso se utilizan datos de entrenamiento que consisten en objetos de entrada y sus valores de salida correspondientes. Su objetivo consiste en encontrar una función que mapee los datos de entrada a la salida deseada para minimizar el sesgo y el error de varianza de los resultados previstos.

- Aprendizaje no supervisado [24]: en este caso se utilizan datos de entrenamiento como objetos de entrada, pero, en este caso, sin valores de salida. Su objetivo consiste en encontrar una función que sea capaz de describir la estructura de los datos y, de esta forma, agrupar los datos de entrada que no han sido ordenados.
- Aprendizaje semisupervisado [24]: en este tipo de algoritmo se utiliza una pequeña cantidad de datos que se encuentran etiquetados que se complementa con una gran cantidad de datos no etiquetados.
- Aprendizaje por refuerzo [24]: su objetivo consiste en entrenar a un agente que debe tomar decisiones en base a sus observaciones.

La finalidad de un algoritmo de Machine Learning puede ser muy dispar [24], desde cómo clasificar una imagen hasta predecir la probabilidad de que un evento suceda. En la mayoría de los casos se utilizan para la detección de anomalías, la clasificación de datos, la agrupación de clústeres, la detección de características, codificación, entre otras cosas.

El método lo que consigue es obligar al algoritmo a mejorar en función de nuevos datos de entrada con el fin de mejorar su precisión. Existen distintas técnicas de machine learning pero en este TFG se han utilizado las redes neuronales para evaluar la estimación de localización utilizando como datos de entrada la señal CSI, obtenidos con el ESP32. En el siguiente apartado se explicará brevemente el funcionamiento de las redes neuronales.

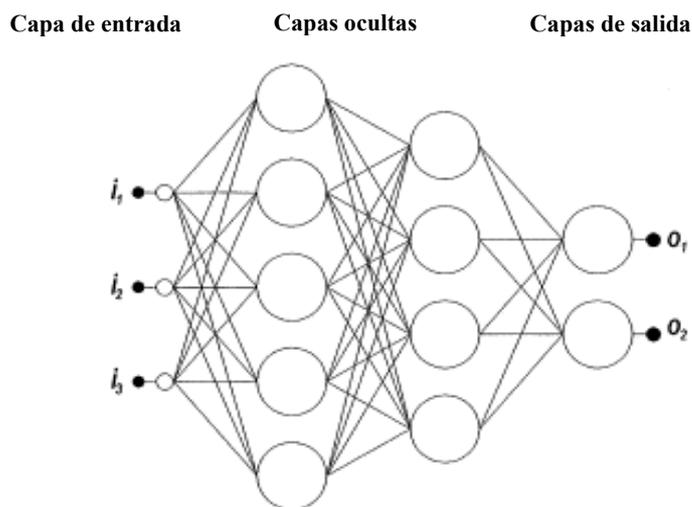
2.7 Redes neuronales

Las redes neuronales artificiales [25], pertenecen a la rama de la inteligencia artificial, intentan simular la forma que tiene el cerebro humano de procesar la información. Debido a su auge, en muchas disciplinas científicas y proyectos de investigación, las redes neuronales están sustituyendo a los métodos estadísticos tradicionales. Esto se debe a que se ha demostrado que, tienen una aplicación efectiva y eficiente, cuando se obtiene un conjunto de datos muy grande, con respecto al modelado estadístico tradicional.

Para el análisis de datos obtenidos en la realización de este proyecto, se ha utilizado el perceptrón multicapa. El perceptrón multicapa [25] consiste en un sistema de “neuronas” o nodos que se encuentran interconectados entre sí, mediante pesos y señales de salida. En el caso del perceptrón multicapa, la red posee una capa de entrada, cuya función únicamente es pasar el vector de entrada a la red, una o más capas ocultas y una capa de salida, como se muestra en la ilustración 1. La arquitectura de una red neuronal perceptrón multicapa se caracteriza por agrupar los nodos en diferentes niveles, tal y como puede verse en la ilustración 1:

- Capa de entrada: su función es la de propagar el vector de entrada al resto de la red. Su dimensión, es decir el número de nodos debe de coincidir con el vector de entrada.
- Capa de salida: su función es la de proporcionar la salida de la red neuronal.
- Capas ocultas: reciben las señales del nivel de entrada o nivel anterior. Cada nodo aplica una función no lineal donde intervienen los pesos y bias. Una red neuronal puede tener varias capas ocultas.

El vector de entrada y el vector de salida hacen referencia a las entradas y salidas del perceptrón multicapa. Ambos pueden ser representados como vectores únicos.



$I = [i_1, i_2, i_3] = \text{vector de entrada}$

$O = [o_1, o_2, o_3] = \text{vector de salida}$

Ilustración 1. Perceptrón multicapa [25]

Se ha demostrado que [25] este tipo de red neuronal es capaz de entrenarse para aproximarse a cualquier tipo de función medible. Además, en este caso no necesita realizar suposiciones previas, sino que, directamente modela funciones que no tienen que ser necesariamente lineales, y se entrena con precisión cuando recibe datos nuevos. Estas características hacen que el perceptrón multicapa sea una alternativa muy llamativa para elegir entre enfoques estadísticos.

Como se ha mencionado anteriormente, las redes neuronales deben de entrenarse. Para realizar este entrenamiento se necesita un conjunto de datos que posee vectores de entrada y de salida asociados. El perceptrón multicapa presenta repetidamente los datos de entrenamiento y los pesos de la red se ajustan hasta que se obtiene el mapeo de entrada-salida deseado [25].

A medida que se realiza el entrenamiento se podrá observar, como en ocasiones, la salida, para un vector de entrada dado, no es la salida deseada. De esta forma, se puede obtener una señal de error, que es la diferencia entre la salida deseada y la obtenida [25]. La magnitud de la señal de error es utilizada como referencia para ajustar los pesos de la red y, reducir el error general de la red neuronal. Tras haber finalizado el entrenamiento con datos de entrenamiento, el perceptrón multicapa puede ser generalizado a datos nuevos de entrada.

Capítulo 3. Recursos del entorno de medidas

En este capítulo se hará una descripción de las tecnologías requeridas y utilizadas para la realización de este proyecto. Primero, se hará una explicación de los recursos hardware utilizados, explicando las características técnicas del microcontrolador ESP32, necesario para realizar las pruebas pertinentes y las del punto de acceso utilizado para la obtención de estas. Luego, se describirán brevemente los recursos software empleadas como lo son los distintos lenguajes de programación y el repositorio que ha servido para la extracción de datos.

3.1 Recursos hardware

En este capítulo se describirán las características más relevantes de los recursos hardware utilizados para el desarrollo de este proyecto.

3.1.1 Microcontrolador ESP32

Un microcontrolador [27] consiste en un circuito integrado, digital síncrono que contiene una unidad central de procesamiento, una memoria que almacena el código de firmware, la memoria de programa, y una memoria de datos.

Hace unos años, era muy costoso incorporar conectividad Wi-Fi a un proyecto, y este hecho cambió tras el lanzamiento del SoC ESP8266, [28] por la empresa Espressif Systems, que integra un microcontrolador con una arquitectura de 32 bits, conexión WiFi, compatible con el protocolo TCP/IP.

Su primer módulo disponible fue el ESP-01 lanzado en agosto de 2014, por Ai-Thinker, una empresa de alta tecnología que integra diseño, desarrollo y producción en el campo de IoT. Este módulo integra memoria Flash, cristal oscilador y antena Wifi en PCB. Su configuración se realizaba mediante comandos AT y únicamente se utilizaba como módulo Wifi. A medida que se fue empleando en diferentes proyectos se percataron de su potencial como procesador independiente y no sólo como módulo.

Debido a esto, en el año 2016 se lanzó el sucesor del ESP8266 diseñado por Espressif Systems pero fabricado por la empresa TSM, llamado ESP32.

El ESP32 consiste en una familia de SoC [29] [30] y módulos de bajo costo y consumo de energía. En esta familia de chips podemos encontrar diferentes tipos, basados en el microprocesador Tensilica Xtensa LX6, de uno o dos núcleos, cuya frecuencia de operación es de hasta 240 MHz.

- ESP32-D0WDQ6
- ESP32-D0WD
- ESP32-D2WD
- ESP32-S0WD
- EP32-PICO-D4 (System in Package)

Cada uno de estos está formado por interruptores de antena, balun de RF, un amplificador de potencia, amplificadores de recepción de bajo ruido, y filtros y módulos de administración de energía. Una de sus características más destacadas es que logra un consumo de energía muy bajo que se consigue mediante el empleo de funciones de ahorro de energía como lo es la sincronización de reloj. Esta característica ha convertido a esta familia de chips en la elección perfecta, por ejemplo, para proyectos energizados con batería.

En la ilustración 2 [29] [30] se aprecian los diferentes bloques funcionales de un chip ESP32. Posee un bloque de conectividad inalámbrica en el que destacan la conectividad Wi-Fi y Bluetooth. A su derecha, se encuentra el bloque de radio que se encarga de recibir los datos digitales recogidos por los módulos Wi-Fi y Bluetooth, transformarlos en señales electromagnéticas que hacen posible la comunicación con teléfonos móviles o rúter.

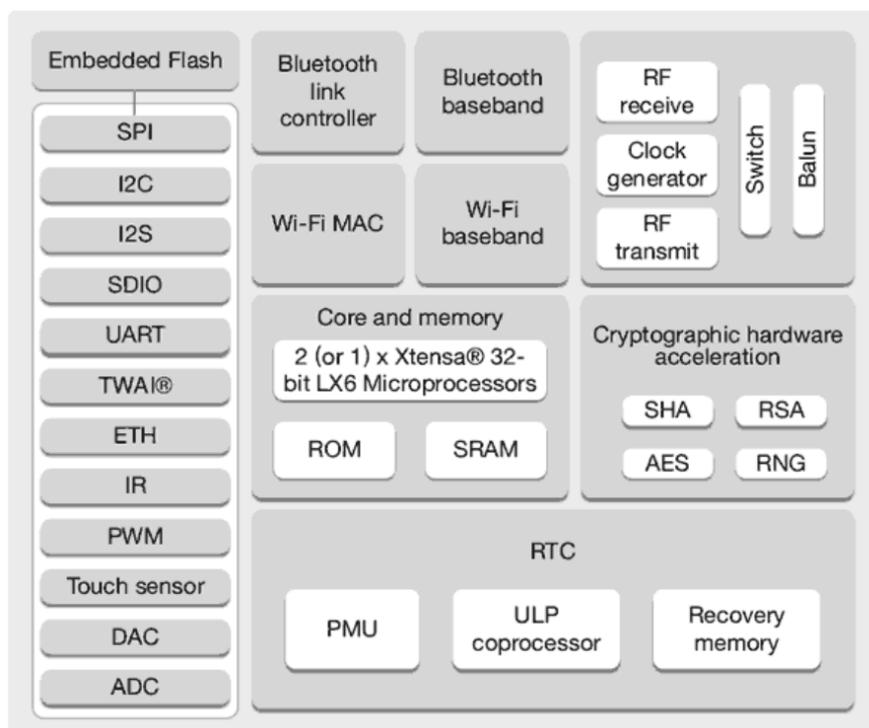


Ilustración 2. Bloques funcionales SoC ESP32

Además de contar con el microprocesador anteriormente mencionado [29] [30], también tienen un co-procesador de bajo consumo encargado de realizar conversiones analógico-digital mientras el ESP32 trabaja en el modo deep sleep.

Una característica propia de los ESP32 es que poseen una mayor cantidad de memorias internas y externas. Las memorias internas ya están incluidas en el SoC mientras que las externas se pueden añadir de manera adicional para incrementar la capacidad del sistema, de esta forma se obtienen placas con mejores prestaciones.

Posee 34 pines digitales que, entre otras cosas permiten añadir LEDs y botones. También tiene tres puertos UART compatibles con los protocolos RS-232, RS-485 e IrDA. Cabe añadir que, cuentan con dos interfaces I2C o TWI que permiten los modos maestro y esclavo. Si comparamos el ESP32 con su predecesor se destaca que posee un núcleo adicional y una mayor cantidad de pines de E/S, su WiFi es más rápida y, además soporta Bluetooth.

Cabe destacar que, el sistema operativo en tiempo real del ESP32 es el FreeRTOS, este es de código abierto, además de que ha sido diseñado para sistemas integrados. Sus funciones principales consisten en la gestión de memoria, de tareas y por último la sincronización de API.

Para programar un ESP32 es necesario utilizar el entorno de desarrollo ESP-IDF, Espressif Systems Internet of Things, que se puede encontrar en su repositorio de GitHub.

3.1.2 Linksys



Ilustración 3. Punto de acceso Linksys WRT 1900 ACS

El router que se ha utilizado para la realización de este proyecto ha sido el Linksys modelo WRT 1900 ACS [33], ya que soporta varios estándares de red, entre ellos el que nos interesa para extraer los valores de CSI, el IEEE 802.11n. Por esta razón, se ha realizado la configuración necesaria para que el router trabaje con dicho estándar ya que es el requerido para obtener los datos deseados de la red inalámbrica.

En la siguiente tabla 2, podemos observar sus características más relevantes:

| | |
|--------------------------------------|--|
| Estándares de red | IEEE 802.11a IEEE 802.11b IEEE 802.11g IEEE 802.11n IEEE 802.11ac |
| Bandas de radiofrecuencia | 2,4 GHz 5 GHz |
| Temperatura de funcionamiento | De 0° a 40° |
| Puertos | Alimentación Internet Ethernet (1-4) USB 3.0 USB 2.0/eSATA combinado |
| Tasa de enlace máxima | 1300 Mbps |

Tabla 2. Características rúter WRT 1900 ACS [33]

3.2 Recursos software

En segundo lugar, se hará una descripción de las características del software empleado para la realización de este TFG y, a su vez se analizarán las características pertinentes.

3.2.1 ESP-IDF

Para el desarrollo de dispositivos ESP32 su fábrica, Espressif Systems, proporciona de recursos de hardware y software, llamado ESP-IDF [34]. Este entorno de desarrollo realmente fue desarrollado para poder utilizarlo en Linux, por lo tanto, es necesario un terminal Linux para poder ejecutar los archivos bash. Aunque se ha demostrado que, es posible utilizarlo en otros sistemas operativos sin tener muchas dificultades para su ejecución y desarrollo.

El ESP-IDF, ofrece un menú de configuración ‘interactivo’ y accesible mediante el comando “idf.py menuconfig” [35], este se puede apreciar en la ilustración 4.

```
(Top)
Espressif IoT Development Framework Configuration
SDK tool configuration --->
Build type --->
Application manager --->
Bootloader config --->
Security features --->
Serial flasher config --->
Partition Table --->
ESP32 CSI Tool Config --->
Compiler options --->
Component config --->
Compatibility options --->

[Space/Enter] Toggle/enter [ESC] Leave menu [S] Save
[O] Load [?] Symbol info [/] Jump to symbol
[F] Toggle show-help mode [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```

Ilustración 4. Menú de configuración passive

El resto de las operaciones posibles a realizar, como compilar o flashear al ESP32, se llevan a cabo ejecutando comandos simples ofrecidos por el mismo software. El ESP-IDF se puede instalar entre las distintas herramientas del IDE Eclipse, de esta forma no sería necesario el uso del terminal ya que el mismo software ejecutaría todos los comandos, de cualquier operación, automáticamente.

Cabe destacar que, aunque el lenguaje de programación más utilizado para programar un ESP32 es C, y por ello la mayoría de las bibliotecas API se encuentran en este lenguaje, es posible programar un microcontrolador de este tipo en C++. Realmente, como se trata de un chip de código abierto cualquiera puede desarrollar un “sistema operativo” de trabajo para este, por lo que en la red se pueden encontrar múltiples soluciones para lograr programarlo en otros lenguajes de programación como JavaScript.

3.2.2 GitHub

Para el desarrollo de este proyecto se ha utilizado un repositorio, que contiene información sobre la programación para la extracción de datos CSI [35]. En este repositorio se encuentran los pasos y ejemplos que programan el ESP32 para obtener valores de CSI. Se decidió elegir este repositorio debido a que era sencillo de configurar

y más eficiente a la hora de extraer los datos requeridos para la realización de este proyecto.

Además, en este repositorio se hace hincapié de otras herramientas complementarias que son necesarias para la correcta ejecución de este programa, y, para que fuera posible extraer los datos CSI sin mayor problema. Por ello, se procedió a la instalación y configuración de estas: Java 11, Python 3.8, Git y las herramientas ESP-IDF 4.0, mencionadas anteriormente y, necesarias para la completa configuración del ESP32.

A continuación, se procederá a realizar una breve explicación sobre las configuraciones requeridas por el programa y de las herramientas de Espressif Systems para poder proceder a la toma de medidas. [35]

La instalación y la configuración de cada una de las herramientas se realizaron en un portátil con sistema operativo MacOs ya que se estableció que este se utilizaría para extraer los valores de CSI junto con el microcontrolador ESP32, con el apoyo del repositorio anteriormente mencionado.

Se seleccionó uno de los subproyectos ofrecidos por este programa para la toma de datos CSI, en este caso se eligió el 'passive' debido a que el ESP32 se encuentra capturando datos, no transmitiéndolos.

Con el fin de acceder a la configuración de este directorio se empleó la siguiente línea de comandos que podemos ver en la ilustración 5:

```
idf.py menuconfig
```

Ilustración 5. Acceder a menú de configuración

```

(Top)
Espressif IoT Development Framework Configuration
SDK tool configuration --->
Build type --->
Application manager --->
Bootloader config --->
Security features --->
Serial flasher config --->
Partition Table --->
ESP32 CSI Tool Config --->
Compiler options --->
Component config --->
Compatibility options --->

[Space/Enter] Toggle/enter [ESC] Leave menu [S] Save
[O] Load [?] Symbol info [/] Jump to symbol
[F] Toggle show-help mode [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)

```

Ilustración 6. Pantalla de configuración ESP32

Tras esto, se accede a la pantalla de configuración, ilustración 6, y se selecciona el siguiente tipo de configuración:

- Se personaliza la tasa de baudios a 155200, ver ilustración 7, ya que permite que se transmitan más datos en el puerto serie.

```

(Top) > Serial flasher config
Espressif IoT Development Framework Configuration
[ ] Disable download stub
Flash SPI mode (DIO) --->
Flash SPI speed (40 MHz) --->
Flash size (2 MB) --->
[*] Detect flash size when flashing bootloader
Before flashing (Reset to bootloader) --->
After flashing (Reset after flashing) --->
'idf.py monitor' baud rate (Custom baud rate) --->
(155200) Custom baud rate value

[Space/Enter] Toggle/enter [ESC] Leave menu [S] Save
[O] Load [?] Symbol info [/] Jump to symbol
[F] Toggle show-help mode [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)

```

Ilustración 7. Configuración de la tasa de baudios

- Se modifica el canal de salida de consola: UART, ilustración 8, y su tasa de baudios como la establecida en el apartado anterior.

```
(Top) > Component config > Common ESP-related
Espressif IoT Development Framework Configuration
[*] Enable lookup of error code strings
(32) System event queue size
(2304) Event loop task stack size
(3584) Main task stack size
(1024) Inter-Processor Call (IPC) task stack size
[*] IPC runs at caller's priority
(2048) Minimal allowed size for shared stack
Channel for console output (Custom UART) --->
  UART peripheral to use for console output (0-1) (UART0) --->
(1) UART TX on GPIO#
(3) UART RX on GPIO#
(1552000) UART console baud rate
[*] Interrupt watchdog
(300)  Interrupt watchdog timeout (ms)
[*]   Also watch CPU1 tick interrupt
[*] Initialize Task Watchdog Timer on startup
[ ]   Invoke panic handler on Task Watchdog timeout
|-----|
[Space/Enter] Toggle/enter  [ESC] Leave menu      [S] Save
[O] Load                    [?] Symbol info      [/] Jump to symbol
[F] Toggle show-help mode   [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```

Ilustración 8. Configuración del canal de salida

- Se establece que se quiere extraer datos CSI de la red inalámbrica, ilustración 9.

```
(Top) > Component config > Wi-Fi
Espressif IoT Development Framework Configuration
(10) Max number of WiFi static RX buffers
(32) Max number of WiFi dynamic RX buffers
  Type of WiFi TX buffers (Dynamic) --->
(32) Max number of WiFi dynamic TX buffers
[*] WiFi CSI(Channel State Information)
[*] WiFi AMPDU TX
(6)   WiFi AMPDU TX BA window size
[*] WiFi AMPDU RX
(6)   WiFi AMPDU RX BA window size
[*] WiFi NVS flash
  WiFi Task Core ID (Core 0) --->
(752) Max length of WiFi SoftAP Beacon
(32) WiFi mgmt short buffer number
  WiFi default log level (Info) --->
[*] WiFi IRAM speed optimization
[*] WiFi RX IRAM speed optimization
[*] Enable WPA3-Personal
|-----|
[Space/Enter] Toggle/enter  [ESC] Leave menu      [S] Save
[O] Load                    [?] Symbol info      [/] Jump to symbol
[F] Toggle show-help mode   [C] Toggle show-name mode [A] Toggle show-all mode
[Q] Quit (prompts for save) [D] Save minimal config (advanced)
```

Ilustración 9. Establecer extracción de datos CSI

Tras haber realizado las configuraciones pertinentes se procede a realizar la toma de medidas.

3.2.3 Matlab

Una vez configurado el microcontrolador ESP32 para la captura CSI, se obtienen medidas sobre diversos escenarios. Con las medidas obtenidas se pasa a la siguiente fase, el procesado de datos, y la evaluación de una red neuronal como sistema de predicción. Para ello, se ha utilizado la herramienta Matlab, utilizando la licencia de estudiante proporcionada por la universidad, debido a que ofrece las librerías correspondientes para la creación de estas. Y su uso es casi de inmediato.

La librería utilizada ha sido 'Deep Learning' [36] ya que sirve como apoyo para la implementación de redes neuronales. De esta forma es posible diseñar y entrenar diferentes redes para que lleven a cabo, por ejemplo, tareas de clasificación de datos que serían tediosas de hacer sin este tipo de tecnología.

Cabe resaltar que, el fin de este proyecto no consiste en la creación de una red neuronal sino en la extracción de datos CSI de una red inalámbrica. Estos valores son obtenidos a través de un microcontrolador ESP32, que se utiliza como elemento que puede extraer este tipo de medidas. Y, finalmente, realizar una evaluación rápida de estos valores, razón por la que se tomó la decisión de utilizar las redes neuronales.

3.2.4 Lenguaje de programación: Python

Como los datos que el ESP32 capturaba se imprimían en la consola, fue necesario elaborar una serie de scripts para capturar la consola, o el puerto del ESP32, con el fin de almacenarlos en el ordenador para su posterior procesado. Estos se desarrollaron con el lenguaje de programación Python, debido a que se trata de un lenguaje de programación de código abierto, además de que ofrece múltiples bibliotecas para múltiples funciones diferentes.

De esta forma, obtenemos ficheros CSV con la extracción de datos deseados para la realización de este proyecto. En las siguientes ilustraciones se puede observar un ejemplo de la toma de datos:

| | | | | | | | | | | | | | | | | | | | | | | | |
|----------|---------|-------------|-----|----|---|---|---|---|---|---|---|---|---|-----|---|----|---|----------|---|-----|---|---|---------|
| CSI_DATA | PASSIVE | C4-E9-84-1F | -44 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.91E+09 | 0 | 98 | 0 | 0 | 1908.36 |
| CSI_DATA | PASSIVE | S8-EF-68-0E | -42 | 11 | 1 | 7 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | -97 | 0 | 13 | 0 | 1.91E+09 | 0 | 98 | 0 | 0 | 1908.37 |
| CSI_DATA | PASSIVE | C4-E9-84-1F | -43 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.91E+09 | 0 | 98 | 0 | 0 | 1909.37 |
| CSI_DATA | PASSIVE | S8-EF-68-0E | -42 | 11 | 1 | 7 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | -97 | 0 | 13 | 0 | 1.91E+09 | 0 | 98 | 0 | 0 | 1909.37 |
| CSI_DATA | PASSIVE | C4-E9-84-1F | -43 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.91E+09 | 0 | 98 | 0 | 0 | 1910.38 |
| CSI_DATA | PASSIVE | S8-EF-68-0E | -43 | 11 | 1 | 7 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | -97 | 0 | 13 | 0 | 1.91E+09 | 0 | 98 | 0 | 0 | 1910.39 |
| CSI_DATA | PASSIVE | C4-E9-84-1F | -43 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.91E+09 | 0 | 98 | 0 | 0 | 1911.39 |
| CSI_DATA | PASSIVE | S8-EF-68-0E | -42 | 11 | 1 | 7 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | -97 | 0 | 13 | 0 | 1.91E+09 | 0 | 98 | 0 | 0 | 1911.4 |
| CSI_DATA | PASSIVE | C4-E9-84-1F | -43 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.91E+09 | 0 | 90 | 0 | 0 | 1912.17 |
| CSI_DATA | PASSIVE | S8-EF-68-0E | -42 | 11 | 1 | 7 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | -97 | 0 | 13 | 0 | 1.91E+09 | 0 | 118 | 0 | 0 | 1912.18 |
| CSI_DATA | PASSIVE | C4-E9-84-1F | -43 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.91E+09 | 0 | 98 | 0 | 0 | 1912.41 |
| CSI_DATA | PASSIVE | S8-EF-68-0E | -42 | 11 | 1 | 7 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | -97 | 0 | 13 | 0 | 1.91E+09 | 0 | 98 | 0 | 0 | 1912.41 |
| CSI_DATA | PASSIVE | C4-E9-84-1F | -43 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.91E+09 | 0 | 90 | 0 | 0 | 1913.17 |
| CSI_DATA | PASSIVE | S8-EF-68-0E | -42 | 11 | 1 | 7 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | -97 | 0 | 13 | 0 | 1.91E+09 | 0 | 118 | 0 | 0 | 1913.18 |
| CSI_DATA | PASSIVE | C4-E9-84-1F | -43 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.91E+09 | 0 | 98 | 0 | 0 | 1913.42 |
| CSI_DATA | PASSIVE | S8-EF-68-0E | -42 | 11 | 1 | 7 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | -97 | 0 | 13 | 0 | 1.91E+09 | 0 | 98 | 0 | 0 | 1913.43 |
| CSI_DATA | PASSIVE | C4-E9-84-1F | -43 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.91E+09 | 0 | 98 | 0 | 0 | 1914.44 |
| CSI_DATA | PASSIVE | S8-EF-68-0E | -42 | 11 | 1 | 7 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | -97 | 0 | 13 | 0 | 1.91E+09 | 0 | 98 | 0 | 0 | 1914.44 |
| CSI_DATA | PASSIVE | C4-E9-84-1F | -43 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 90 | 0 | 0 | 1915.18 |
| CSI_DATA | PASSIVE | S8-EF-68-0E | -42 | 11 | 1 | 7 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 118 | 0 | 0 | 1915.19 |
| CSI_DATA | PASSIVE | C4-E9-84-1F | -43 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 98 | 0 | 0 | 1915.44 |
| CSI_DATA | PASSIVE | S8-EF-68-0E | -42 | 11 | 1 | 7 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 98 | 0 | 0 | 1915.45 |
| CSI_DATA | PASSIVE | C4-E9-84-1F | -43 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 98 | 0 | 0 | 1916.46 |
| CSI_DATA | PASSIVE | S8-EF-68-0E | -42 | 11 | 1 | 7 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 98 | 0 | 0 | 1916.46 |
| CSI_DATA | PASSIVE | C4-E9-84-1F | -43 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 98 | 0 | 0 | 1917.47 |
| CSI_DATA | PASSIVE | S8-EF-68-0E | -42 | 11 | 1 | 7 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 98 | 0 | 0 | 1917.47 |
| CSI_DATA | PASSIVE | C4-E9-84-1F | -45 | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 66 | 0 | 0 | 1918.26 |
| CSI_DATA | PASSIVE | S8-EF-68-0E | -42 | 11 | 1 | 7 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 66 | 0 | 0 | 1918.26 |
| CSI_DATA | PASSIVE | C4-E9-84-1F | -43 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 98 | 0 | 0 | 1918.49 |
| CSI_DATA | PASSIVE | S8-EF-68-0E | -43 | 11 | 1 | 7 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 98 | 0 | 0 | 1918.49 |
| CSI_DATA | PASSIVE | C4-E9-84-1F | -43 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 90 | 0 | 0 | 1919.18 |
| CSI_DATA | PASSIVE | S8-EF-68-0E | -43 | 11 | 1 | 7 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 118 | 0 | 0 | 1919.19 |
| CSI_DATA | PASSIVE | C4-E9-84-1F | -43 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 98 | 0 | 0 | 1919.49 |
| CSI_DATA | PASSIVE | S8-EF-68-0E | -43 | 11 | 1 | 7 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 98 | 0 | 0 | 1919.5 |
| CSI_DATA | PASSIVE | C4-E9-84-1F | -43 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 98 | 0 | 0 | 1920.51 |
| CSI_DATA | PASSIVE | S8-EF-68-0E | -42 | 11 | 1 | 7 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 98 | 0 | 0 | 1920.51 |
| CSI_DATA | PASSIVE | C4-E9-84-1F | -43 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 240 | 0 | 0 | 1920.96 |
| CSI_DATA | PASSIVE | S8-EF-68-0E | -43 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 98 | 0 | 0 | 1921.52 |
| CSI_DATA | PASSIVE | C4-E9-84-1F | -42 | 11 | 1 | 7 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 98 | 0 | 0 | 1921.53 |
| CSI_DATA | PASSIVE | S8-EF-68-0E | -43 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 240 | 0 | 0 | 1921.96 |
| CSI_DATA | PASSIVE | C4-E9-84-1F | -43 | 11 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | -97 | 0 | 13 | 0 | 1.92E+09 | 0 | 98 | 0 | 0 | 1922.54 |

Ilustración 12. Datos tomados en pasillo

En la ilustración 12, se puede observar que cada fila representa una línea de datos completa, mientras que cada columna representa una línea de datos específica. A continuación, se explicará más detalladamente.

A su vez, se puede observar que, empleando el repositorio anteriormente mencionado, en la extracción de datos no obtenemos únicamente valores de CSI. Además de datos CSI se obtienen muchos otros datos sobre la red inalámbrica con la que se había establecido conexión. Pero los más destacables y útiles para poder realizar la investigación de este proyecto son: la dirección MAC siendo de interés la del punto de acceso, columna 3. Los datos de RSSI que se encuentran en la columna 4, el modo en el que está configurada la señal, sig_mode en la columna 6, ya que era necesario que estuviera utilizando el estándar IEEE 802.11n y finalmente, el vector de datos CSI, este se puede observar en la ilustración 13

Capítulo 4. Toma de medidas

En este capítulo se procederá a describir cómo y dónde han sido tomadas las muestras necesarias para su posterior análisis.

4.1 Planteamiento

Las muestras tomadas para el estudio de este TFG han sido extraídas primero, en el laboratorio de transmisión por línea y, luego, en el pasillo de despachos del departamento de ingeniería telemática, en el pabellón C. En ambos lugares, se establecieron diferentes puntos en los que se extraerían valores CSI. A su vez, se decidió tomar muestras en varios puntos definiendo dos *datasets* diferentes, con 30 y 15 puntos respectivamente, mientras que en el pasillo se colocaron 18 puntos para un *dataset*, al que denominaremos A y 36 para otro *dataset*, al que denominaremos B.

Se acordó que, en ambos casos se colocaría un ordenador portátil con sistema operativo Windows en un punto fijo, con el que se haría ping de manera automática, a la IP del punto de acceso del que queremos extraer los datos a procesar. Esto se debe a que, si no se generaba ningún tipo de tráfico no era posible que el programa extrajera ningún tipo de dato. En la ilustración 14, podemos observar el planteamiento descrito anteriormente, de manera gráfica, de cómo se iban a realizar la toma de medidas en este proyecto.

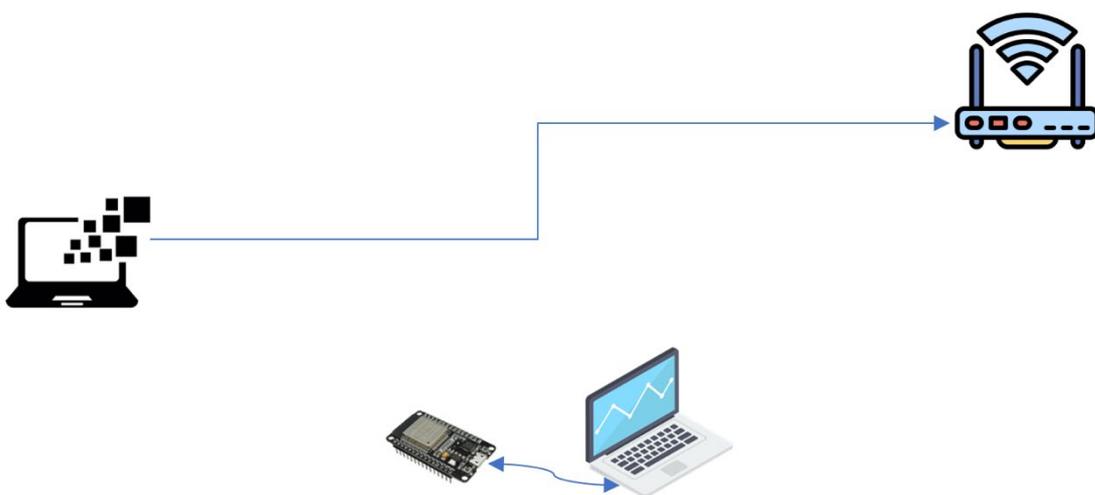


Ilustración 14. Esquema de toma de medidas

La idea principal para ambos lugares consistía en colocar el punto de acceso, con el que se querían tomar los datos, en un punto fijo y tomar los datos deseados de CSI. De esta forma lo que se iba modificando era la ubicación del portátil, así como la orientación del ESP32, con el que se tomaban las medidas, que variaba cada 90°. Esta variación de orientación se llevó a cabo debido a que se quiso estudiar si se producía algún tipo de modificación en las medidas al modificar la orientación del ESP32.

En la ilustración 15, se puede observar el planteamiento de cómo se iba a proceder a la toma de medidas en el laboratorio:

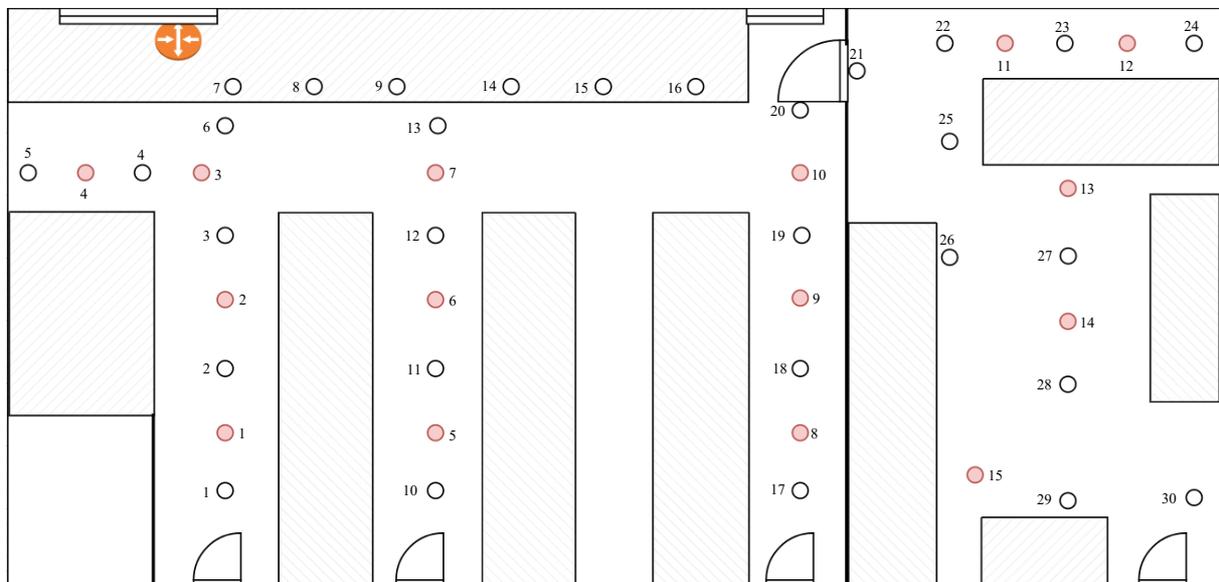


Ilustración 15. Toma de medidas en laboratorio

- Dataset A
- Dataset B

Se observa que el punto de acceso se ubicó debajo de la ventana, y algunos de los puntos de los que se deseaba extraer datos CSI. Además, se puede observar la enumeración utilizada para la toma de medidas deseada en este entorno. En el caso de los puntos blancos indicados en la imagen se trataban de puntos que definen el *dataset A*, o de entrenamiento, mientras que los rojos definirían otro *dataset*, de prueba, este mismo método se utilizó para las medidas tomadas en el pasillo.

A continuación, en la ilustración 16, veremos el planteamiento propuesto para realizar la extracción de muestras en el pasillo, así como la enumeración de las posiciones establecidas:

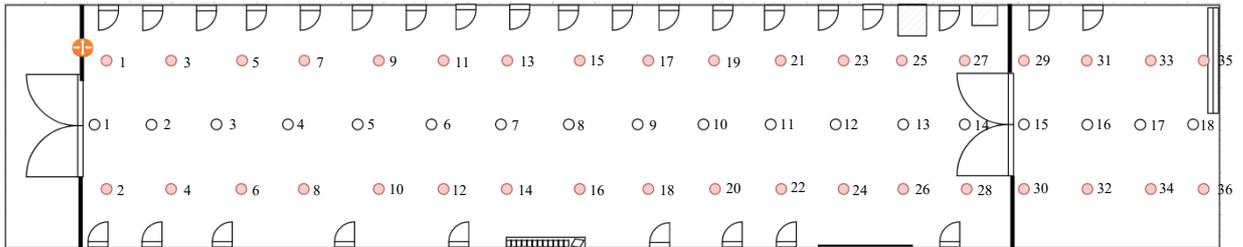


Ilustración 16. Toma de medidas en el pasillo

- Dataset A.
- Dataset B.

4.2 Toma de medidas

Una vez realizada la configuración de todas las herramientas requeridas, así como del programa ofrecido por el repositorio, este se instaló en el microcontrolador, para poder ejecutar el programa y proceder a la toma de medidas de este proyecto. Para su correcta instalación se utilizó el siguiente comando que se puede observar en la ilustración 17:

```
idf.py flash monitor
```

Ilustración 17. Flashear ESP32

Lo que se deseaba obtener era un número de muestras determinado, para luego realizar el procesamiento de estas correctamente, y que, a su vez para cada orientación del ESP32 se extrajera un archivo CSV distinto.

Por ello, se realizó un script de Python, que requería el valor del puerto serie, la tasa de baudios a utilizar y el nombre del archivo. De esta forma capturaba el tráfico de la dirección MAC del punto de acceso, y grababa las muestras a un archivo CSV.

A su vez como se ha comentado anteriormente, cada toma de medida se deseaba tomar en diferentes orientaciones del ESP32, y cada una de ellas con su archivo de muestras correspondiente, por lo que, se creó otro script de Python, con el fin de conseguir que cada orientación establecida en un mismo punto tuviera su propio archivo de datos. Este script se utilizó como complemento al anterior.

En el segundo script se solicita que se extraigan los datos, que el nombre del fichero contenga la orientación en la que se encuentra el ESP32, así como, la fecha del día en el que se toma cada muestra en concreto. De esta forma, cada vez que finaliza un proceso, este se para y solicita que se cambie de orientación para poder continuar con la toma de muestras y así, poder crear un nuevo archivo con sus nuevos datos correspondientes.

Como resultado, se obtuvo una línea de comandos, que se ejecutaba por consola, este último script descrito que, a su vez, contenía al anterior, por lo que ambos eran ejecutados de la misma manera y a la misma vez. Dicha línea de comandos era la siguiente:

```
sh /Users/paulacasteresbaez/Desktop/TFG/Scripts/tomar_muestras_orientacion.sh 192.168.1.1 Labpos 03
```

Ilustración 18. Ejecutar scripts

En la ilustración 18 se observa como, para la posible ejecución del script solicitaba los parámetros de entrada como es la IP del punto de acceso con el que se estuviera haciendo las pruebas, así como su ubicación, en ese caso era el punto tres de uno de los datasets realizados en el laboratorio y a este se le llamó ‘*Labpos 03*’.

En la ilustración 19, se puede observar un ejemplo de cómo se procedió a realizar la toma de medidas, con el ordenador que generaba el tráfico en un punto fijo, así como el punto de acceso, que se encuentra detrás de este y el portátil y el ESP32 tomando las medidas, con una orientación cualquiera.

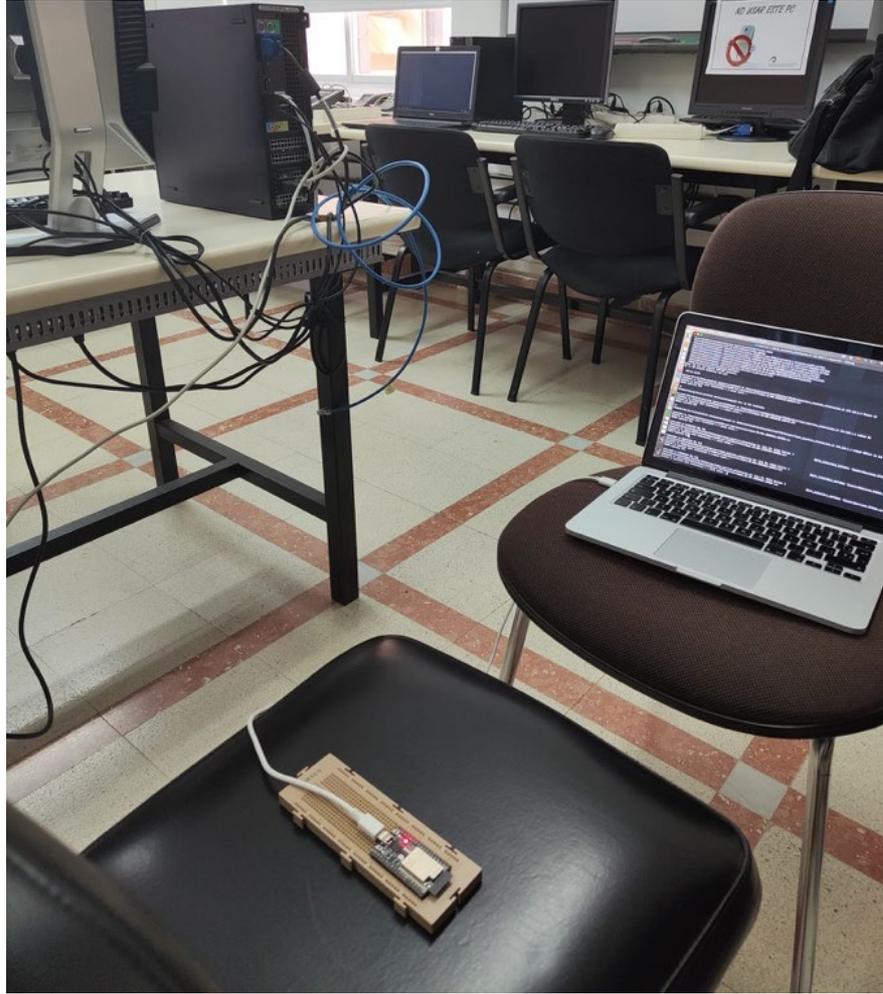


Ilustración 19. Toma de medidas en laboratorio

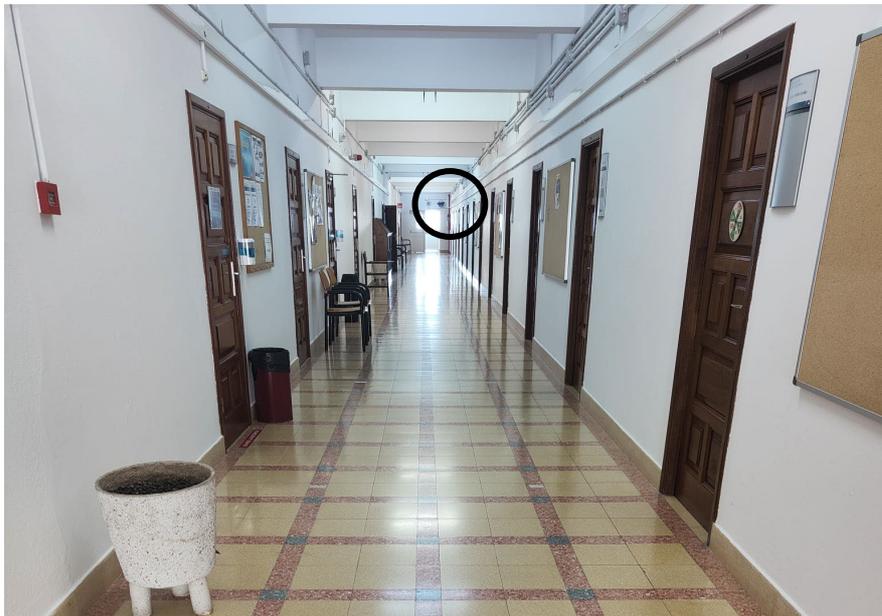


Ilustración 20. Ubicación rúter pasillo

Mientras que, en la ilustración 20 se observa en qué punto del pasillo se ubicó el punto de acceso con respecto al cual tomaríamos las medidas, así como algunos de los puntos marcados en el suelo con el mismo fin.

En la ilustración 21 se observa una de las tomas de las medidas realizadas en el pasillo.



Ilustración 21. Toma de medidas en el pasillo

Capítulo 5. Análisis de datos

En este capítulo se procederá a describir cómo se ha realizado el análisis de datos, así como el entorno de trabajo y las funciones utilizadas para este fin.

5.1 Introducción

Como sistema de predicción de localización se ha optado por experimentar con las redes neuronales. Para ello se han utilizado las librerías o toolboxes de redes neuronales de Matlab.

Matlab [37] es una herramienta de programación desarrollada por MathWorks, orientada al cálculo numérico y utilizada en el desarrollo de algoritmos, procesado de señales, de imágenes, análisis de datos, visión artificial, comunicaciones, finanzas, diseño de control, etc. Consiste en una herramienta orientada a la resolución de problemas científicos y de ingeniería

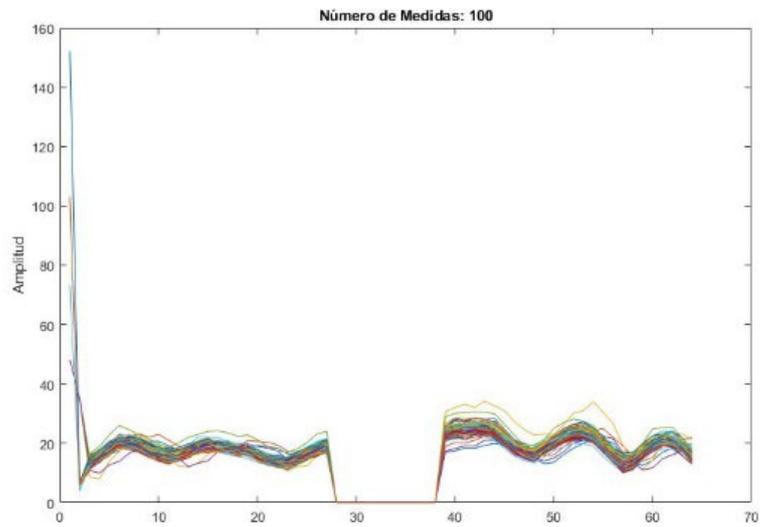
Esta se caracteriza por el uso de matrices, por sus herramientas de visualización y por la gran cantidad de toolboxes que facilitan abordar problemas en un tiempo relativamente corto. Este es el motivo por el que se ha optado por esta herramienta.

Aunque se realizaron medidas en el laboratorio de Transmisión por Línea y en el pasillo de la segunda planta del Pabellón C, los experimentos realizados se han llevado a cabo únicamente con las medidas tomadas en el pasillo.

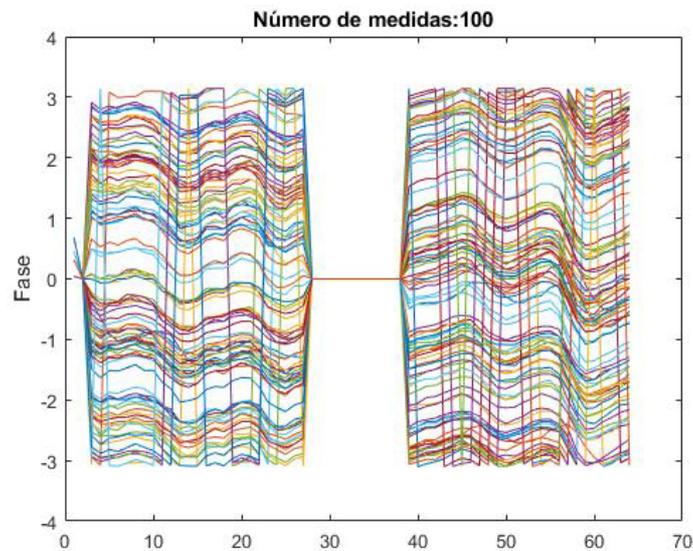
5.2 Análisis de los datos obtenidos en las medidas

Una vez realizadas las medidas en el espacio de interior del pasillo, el siguiente paso fue la de analizar los datos obtenidos.

Tal y como se introdujo en el capítulo 3, los datos capturados por el ESP32 aparecen en la ilustración 22. Este ejemplo es el que corresponde con el fichero de datos del punto 30 y con orientación del ESP32, la número 3. El total de medidas tomadas para este punto es de 492 medidas.



(a)



(b)

Ilustración 23. Amplitud (a) y fase (b) de 100 medidas obtenidas

En la ilustración 23 (a) y 23 (b), se observa amplitud y fase, determinados valores centrales están a 0, concretamente los valores que tienen las posiciones de la 28 a la 38. Esto se repite para todas las medidas obtenidas, en todos los puntos. En la citada ilustración 23 (a), en la amplitud, aparece además como la primera medida no está correlada con el resto de las medidas de las portadoras, esto puede deberse a un estado inicial del microcontrolador ESP32.

Se optó por eliminar este valor inicial además de las medidas encontradas a 0. El resultado es que el número de subportadoras se reduce de 64 valores iniciales a 52 valores. Esta reducción

se aplica a todas las medidas realizadas por punto y para todos los puntos medidos. Tal y como puede verse en la ilustración 24, para la amplitud. Pero el mismo proceso se aplicó también a la fase.

El siguiente paso que se siguió fue la de normalizar las entradas para asegurar que todas tienen el mismo rango y acelerar el proceso de entrenamiento.

La normalización aplicada es siguiendo la fórmula (1):

$$\hat{x}_{normal} = \frac{\hat{x} - \min(\hat{x})}{\max(\hat{x}) - \min(\hat{x})} \quad (1)$$

Donde \hat{x} es el vector de entrada, amplitudes CSI, $\min(\hat{x})$ es el valor mínimo del vector \hat{x} y $\max(\hat{x})$ es el valor máximo. De esta forma los vectores de entrada a la red neuronal estarán comprendidos entre el rango de $[-1,1]$.

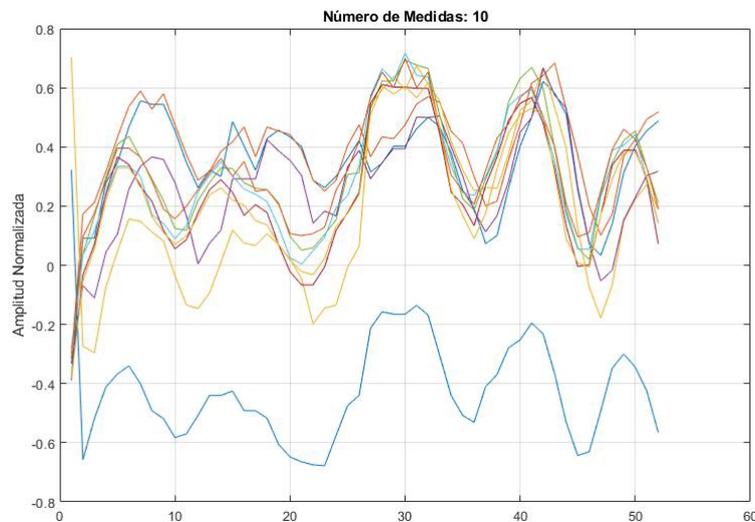


Ilustración 24. Ejemplo de 10 entradas ya normalizadas a la red neuronal

Para el entrenamiento de la red neuronal es necesario definir una salida, se recomienda que la salida de la red neuronal esté comprendida también entre los valores 0 y 1. La solución adoptada es definir tantos nodos de salida como puntos de medidas para entrenar. Es decir, en el capítulo anterior, en ilustración 16, se vio que en el pasillo se marcaron 18 puntos para el *dataset A*, centrales destinados al entrenamiento y unos 36 puntos laterales para el *dataset B*. Por lo tanto, se ha decidido que la capa de salida de la red neuronal sea de 18 nodos o 36 nodos según el

dataset con el que se trabaje. Cada nodo representa un punto de medida. Todos aquellos vectores de medidas realizadas en un punto P_i , tendrán su correspondiente nodo i de salida a 1, y el resto de los nodos estarán a 0.

5.3 Librería de redes neuronales de Matlab

La red neuronal utilizada para llevar a cabo este experimento es una red neuronal sencilla, se trata de un modelo con un solo nivel oculto, la conocida como perceptrón multicapa. La estructura de la red neuronal es de 52 nodos de entrada, correspondiente a las medidas de las subportadoras normalizadas sea amplitud o fase según sea la señal de entrada y 18 o 36 nodos de salida dependiendo del *dataset* de trabajo.

Para los experimentos realizados, se utilizó el algoritmo de entrenamiento *Levenber-Marquardt*. Es el algoritmo que Matlab usa por defecto ya que entrena a gran velocidad, aunque requiere de un mayor esfuerzo computacional.

Lo primero que se decidió fue probar con la señal de amplitud y entrenar una red neuronal con 10 nodos en la capa oculta y comprobar si el modelo elegido satisfacía el funcionamiento del sistema. Se eligió una red neuronal MLP de 3 capas cuya estructura es de [52 10 18], es decir 52 nodos en la capa de entrada, 10 nodos en la capa oculta, y 18 nodos en la capa de salida. La tasa de acierto para el conjunto de entrenamiento es 67.86, lo que invitaba a realizar más pruebas.

Para el entrenamiento de las redes neuronales y evaluar su uso como modelo de predicción de localización, se procedió a realizar los siguientes experimentos:

- Experimento 1: Dataset A y amplitudes como vector de entrada. Estudio del número de nodos óptimo de la capa oculta eligiendo el 70% de medidas para entrenar y el 30% para validar.
- Experimento 2: Dataset A y amplitudes como vector de entrada. Estudio del número de nodos óptimo de la capa oculta eligiendo el 50% de medidas para entrenar y el 30% para validar y 20% de test.
- Experimento 3: Dataset B y amplitudes como vector de entrada. Estudio del número de nodos óptimo de la capa oculta eligiendo el 50% de medidas para entrenar y el 30% para validar y 20% de test.

- Experimento 4: Dataset A y fases como vector de entrada. Estudio del número de nodos óptimo de la capa oculta eligiendo el 50% de medidas para entrenar y el 30% para validar y 20% de test.
- Experimento 5: Dataset B y fases como vector de entrada. Estudio del número de nodos óptimo de la capa oculta eligiendo el 50% de medidas para entrenar y el 30% para validar y 20% de test.

5.4 Experimentos

- **Experimento 1: Dataset A y amplitudes como vector de entrada.** Estudio del número de nodos óptimo de la capa oculta eligiendo el 70% de medidas para entrenar y el 30% para validar.

Tanto el experimento 1 y 2, utilizan la estructura de red neuronal que aparece en la ilustración 25. La capa de entrada está formada por 52 valores de amplitud de CSI, y la capa de salida la forman 18 nodos, correspondiente a los puntos de medida. Los nodos de la capa oculta en estos experimentos es lo que se va modificando para encontrar un valor lo más óptimo posible.

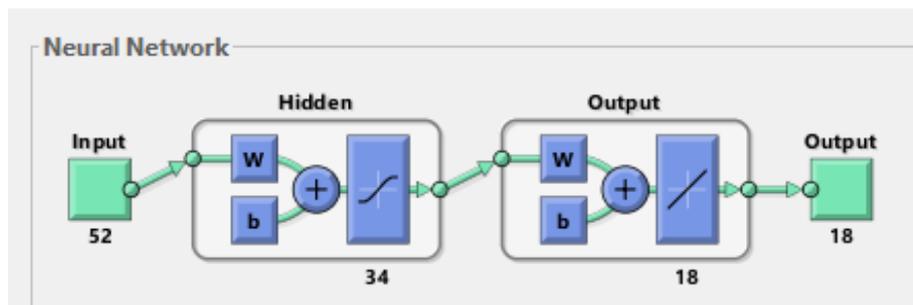
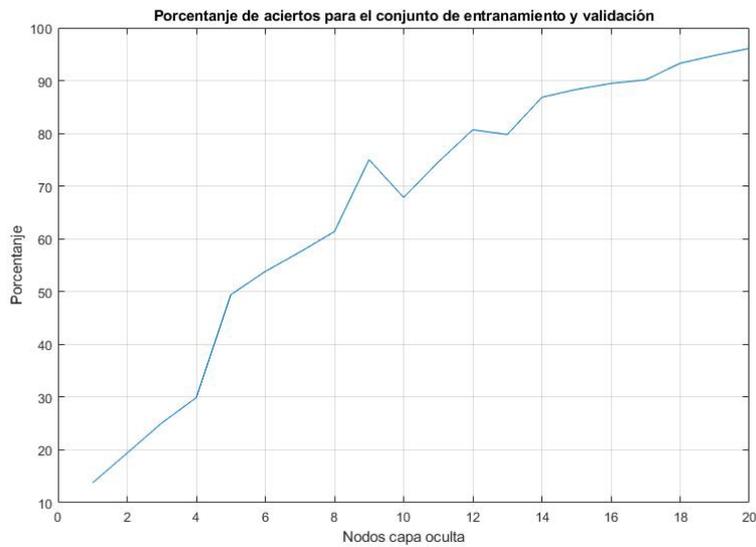
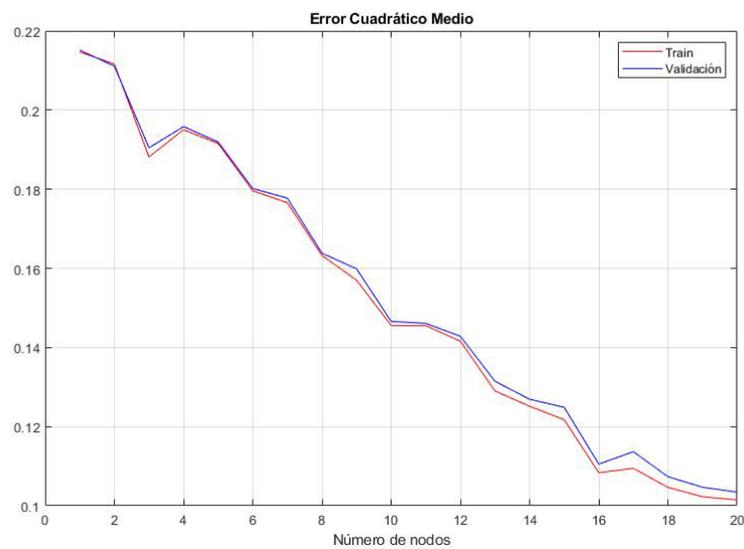


Ilustración 25. Estructura de la red neuronal del experimento 1 y 2

Para estudiar cuál es el número de nodos de la capa oculta, se entrenó 20 redes neuronales desde 1 a 20 nodos.



(a) Tasa de aciertos en %



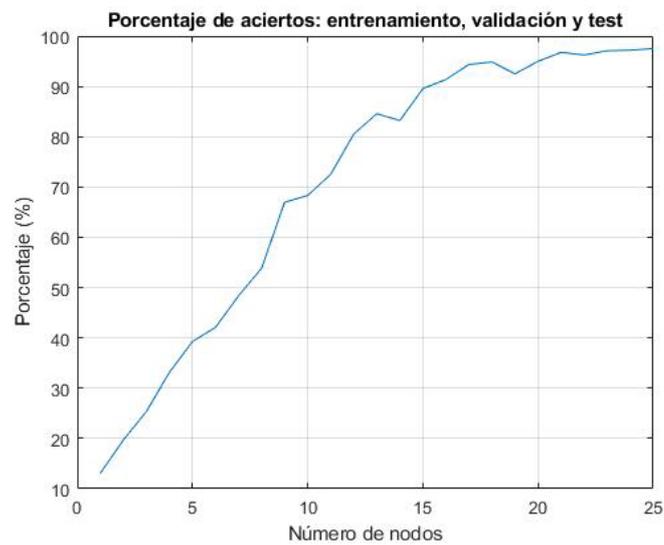
(b) Error cuadrático medio

Ilustración 26. Estudio del número de nodos de la capa oculta experimento 1

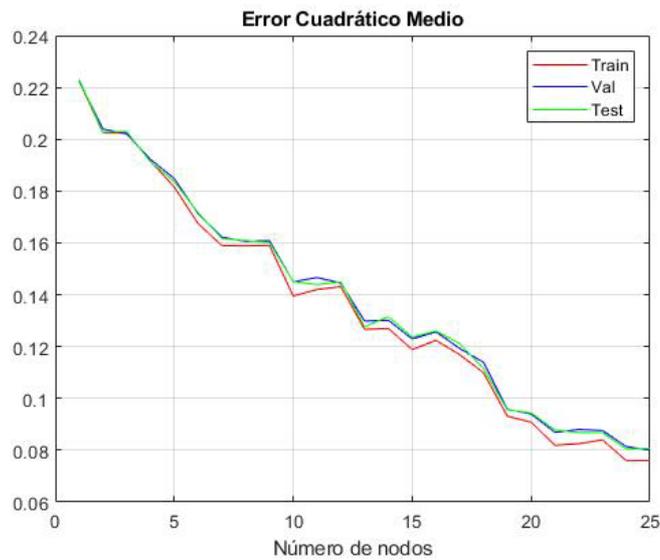
En la ilustración 26(a) se representa el porcentaje de aciertos para todo el conjunto de entrenamiento. Como se puede observar el porcentaje obtenido está por encima del 96%. En la ilustración 26(b) se representa el error cuadrático medio para los conjuntos de entrenamiento y de validación. Tal y como aparece en la ilustración 26, tanto el porcentaje de aciertos sube como el error cuadrático medio disminuye al aumentar el número de nodos de la capa oculta.

Pero a medida que el número de nodos aumenta en la capa oculta, aumenta el coste computacional y por tanto el tiempo empleado para entrenar la red neuronal pudiendo estar para una red de 20 nodos entorno a 1 hora y media.

- **Experimento 2: Dataset A y amplitudes como vector de entrada.** Estudio del número de nodos óptimo de la capa oculta eligiendo el 50% de medidas para entrenar y el 30% para validar y 20% de test.



(a) Tasa de aciertos en %



(b) Error cuadrático medio

Ilustración 27. Estudio del número de nodos de la capa oculta experimento 2.

En la ilustración 27(a) se representa el porcentaje de aciertos para todo el conjunto de entrenamiento, validación y test. Como se puede observar el porcentaje obtenido está por encima del 97%, sin embargo, para llegar a ese rendimiento fue necesario aumentar el número de nodos de la capa oculta y en consecuencia el tiempo empleado en el entrenamiento aumenta. En la ilustración 27(b) se representa el error cuadrático medio para los conjuntos de entrenamiento, validación y de test. Tal y como se observa el error cuadrático medio para el conjunto de validación y de test es algo superior al de entrenamiento, pero van en sintonía con el del entrenamiento.

Comparando el experimento 1 con el 2 en tasa de aciertos, ver ilustración 28, observamos que cuando el número de nodos es bajo, el porcentaje de aciertos es menor para el experimento 2, debido que el conjunto de entrenamiento es inferior, el aprendizaje está limitado por el número de nodos de la red, así como las muestras de entrenamiento. A medida que el número de nodos de la red aumenta parece que la tasa de aciertos es, aproximadamente, de ambos experimentos muy parecida, con lo que se puede concluir que al aumentar el número de nodos la red es posible conseguir tasas de aciertos mejores.

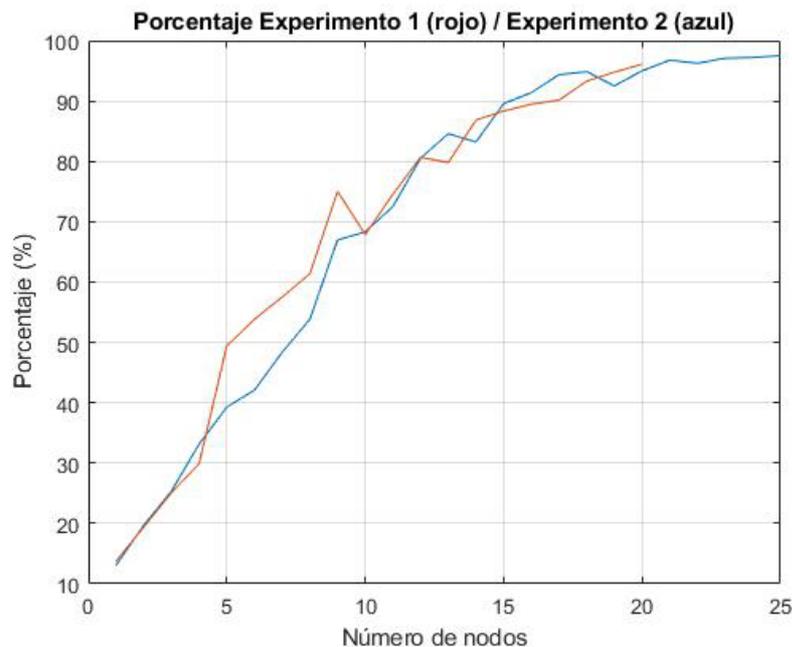


Ilustración 28. Comparación de aciertos del experimento 1 y experimento 2

- **Experimento 3: Dataset B y amplitudes como vector de entrada.** Estudio del número de nodos óptimo de la capa oculta eligiendo el 50% de medidas para entrenar y el 30% para validar y 20% de test.

Para este experimento se partió del dataset B, que consistía en 36 puntos medidos del pasillo.

Hay que recordar que, para cada punto, se realizaron medidas colocando el ESP32 en cuatro orientaciones, cada 90°.

La red neuronal en este caso tiene una estructura tal y como aparece en la ilustración 29, donde la capa de entrada contiene 52 nodos (amplitudes CSI) y la etapa de salida es de 36 nodos correspondientes a cada punto de medida. La capa oculta es la que va a ir variando hasta encontrar un número de nodos aceptable.

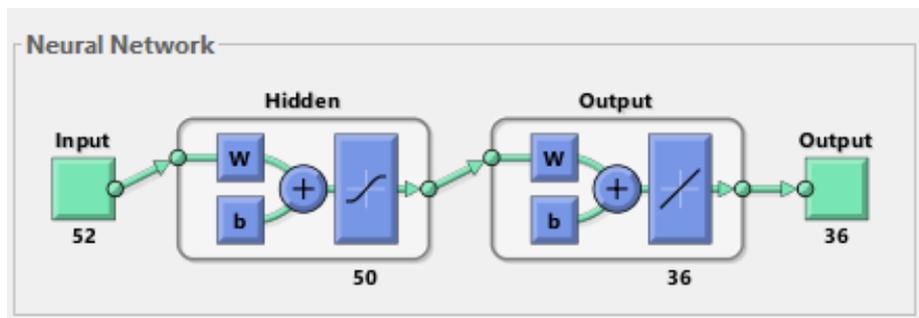
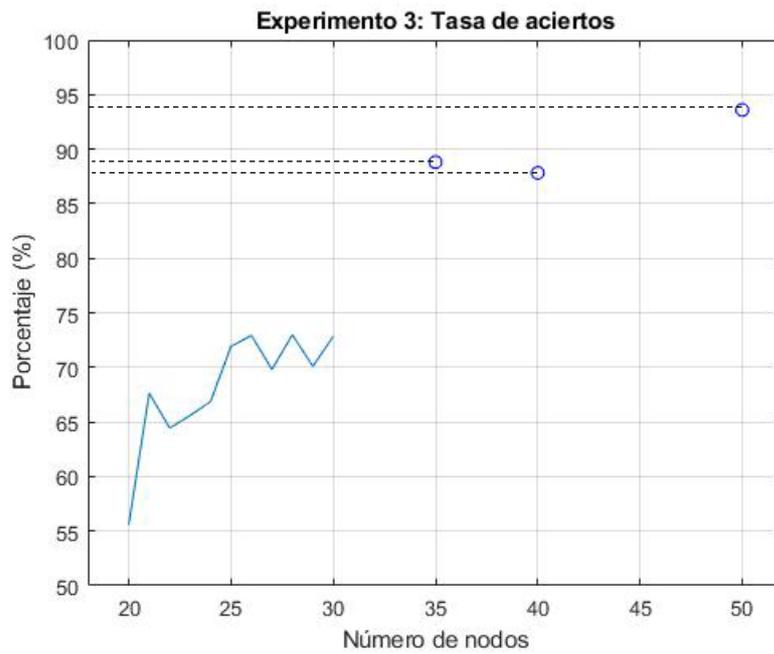
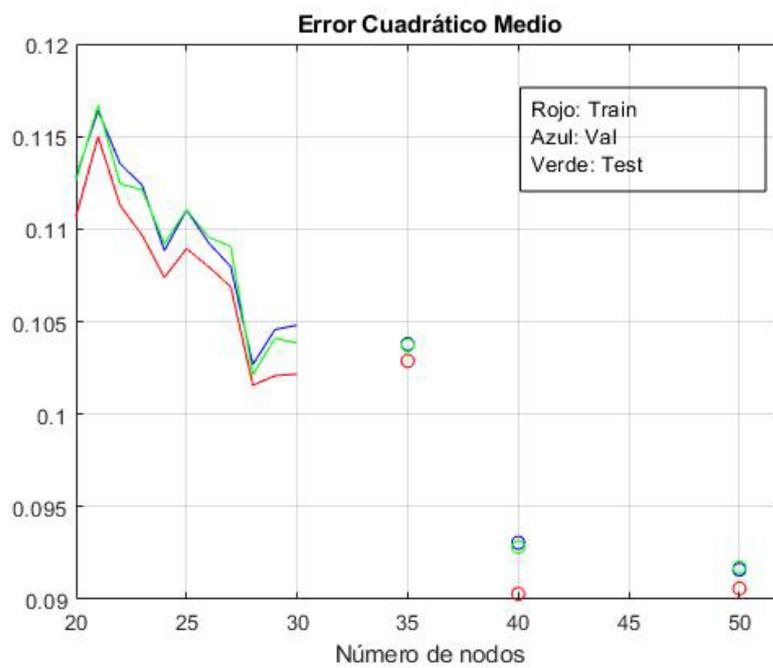


Ilustración 29. Estructura de la red neuronal para el experimento 3

Este experimento al tener un mayor número de puntos, el fichero de entrenamiento se duplica, por lo que el entrenamiento es más pesado, ya que se requiere de un mayor número de nodos en la capa oculta, por tanto, el coste computacional y el tiempo de entrenamiento se eleva considerablemente pasando a 16 horas cuando se tiene 50 nodos en la capa oculta. Como en los experimentos anteriores se alcanzaba tasas por encima del 90% a partir de 20 nodos en la capa oculta, se decidió comenzar con este número en la capa oculta. Las pruebas realizadas fueron la de entrenar 11 redes neuronales desde 20 a 30 nodos en la capa oculta. Al observarse que la tasa de aciertos está por debajo del 75%, se optó por entrenar redes con 35 y 40 nodos consiguiendo tasas entorno al 88%.



(a) Porcentaje de aciertos



(b) Error cuadrático medio

Ilustración 30. Estudio del número de nodos de la capa oculta experimento 3

Según la ilustración 30, se observa la necesidad de aumentar el número de nodos hasta 50 nodos de la capa oculta para obtener resultados por encima del 90%

- **Experimento 4: Dataset A y fases como vector de entrada.** Estudio del número de nodos óptimo de la capa oculta eligiendo el 50% de medidas para entrenar y el 30% para validar y 20% de test.

En este experimento 4 y el siguiente 5, se trabajó en analizar el número de nodos de la capa oculta utilizando la fase como vector de entrada.

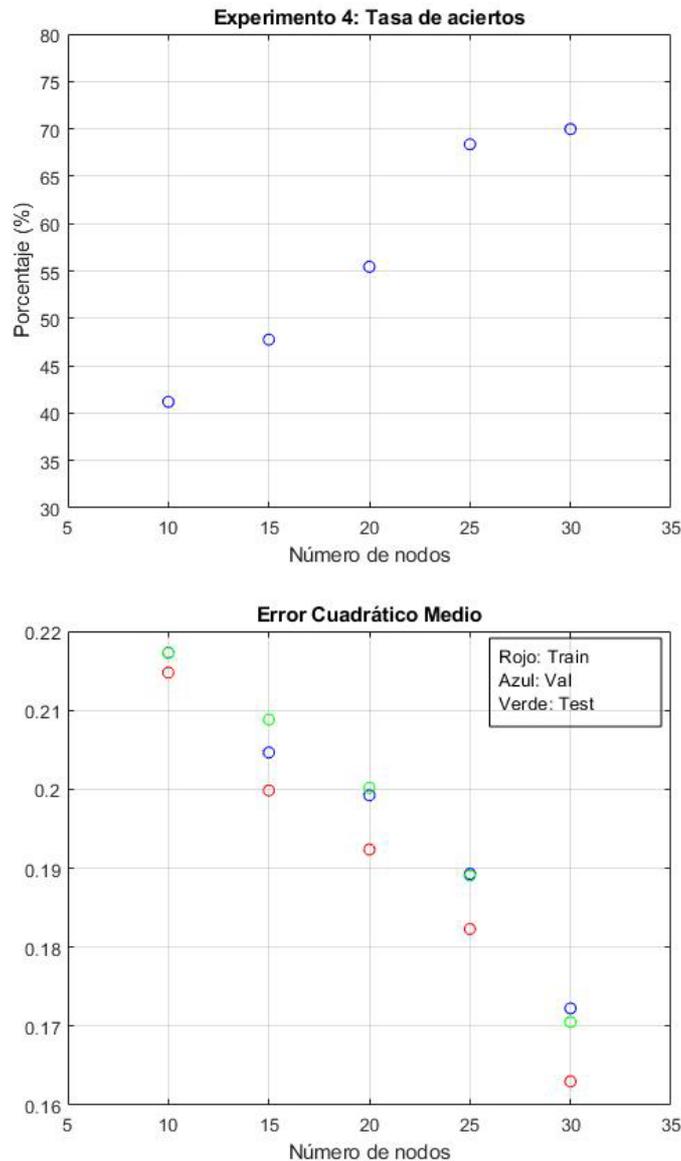


Ilustración 31. Estudio del número de nodos de la capa oculta experimento 4

Como se observa en la ilustración 31, la estructura de la red neuronal cuando se trabaja con la fase como vector de entrada debe ser más compleja para obtener tasas más altas.

- **Experimento 5: Dataset B y fases como vector de entrada.** Estudio del número de nodos óptimo de la capa oculta eligiendo el 50% de medidas para entrenar y el 30% para validar y 20% de test.

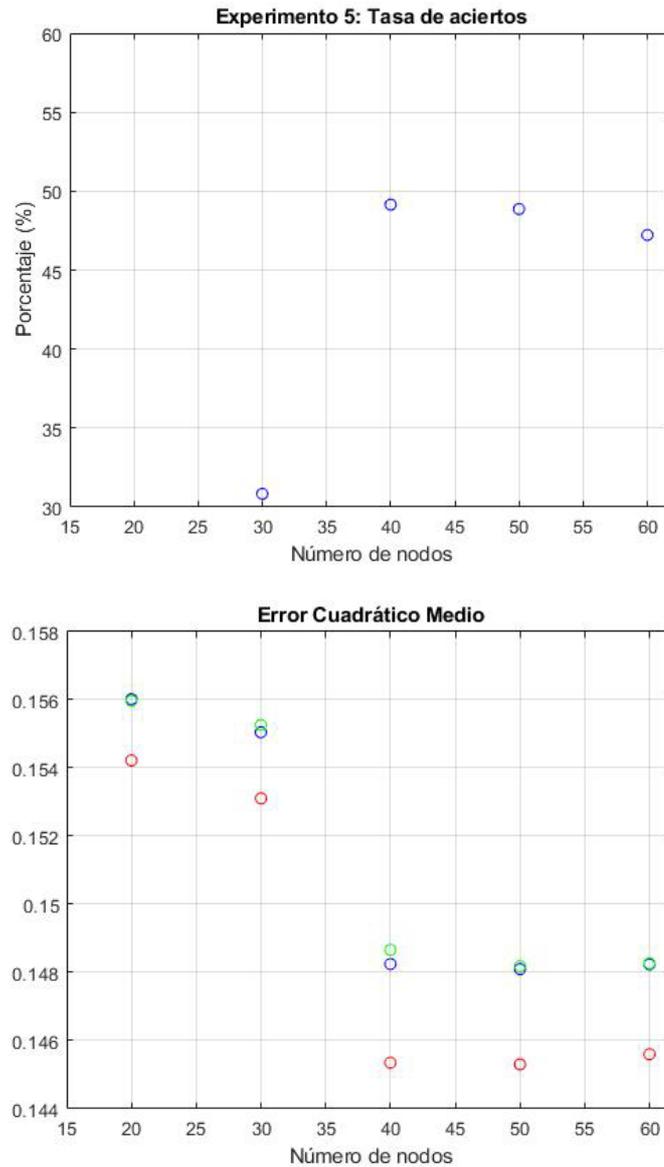


Ilustración 32. Estudio del número de nodos de la capa oculta experimento 5

Según se observa en la ilustración 32, al aumentar el fichero de entrenamiento, el aprendizaje de la red es más costoso. Y la tasa de aciertos se encuentran por debajo del 50%. Probablemente el que exista por cada punto cuatro orientaciones medidas a 0° , 90° , 180° y 270° pueda influir en la fase.

Capítulo 6. Conclusiones y líneas futuras.

Tras el desarrollo de este proyecto y en relación con los objetivos establecidos desde su inicio se ha podido concluir lo siguiente:

- Con respecto al primer objetivo “Estudio de la señal CSI” se consultaron diferentes referencias bibliográficas para obtener información sobre la información del estado del canal.
- En cuanto al estudio y la configuración del ESP32 para la captura de este tipo de señales, O2, se estudió y analizó la documentación ofrecida por el fabricante. Además, se realizó una investigación sobre diferentes repositorios, en GitHub, con el fin de obtener información acerca de la captura de CSI utilizando este microcontrolador. Finalmente se logró configurar el entorno de desarrollo del ESP32 partiendo de los tutoriales ofrecidos por el fabricante y con la ayuda del repositorio seleccionado.
- En lo que respecta al tercer objetivo, “Realización de mediciones”, se desarrollaron los distintos scripts, se llevó a cabo el montaje y la correspondiente valoración de los equipos requeridos para este objetivo. Tras esto, se planificaron los escenarios para la realización de medidas, el pasillo y el laboratorio. Además, se estableció que, de cada punto de medida se tomarían medidas en cuatro orientaciones distintas del ESP32.
- Finalmente, para la evaluación de medidas realizadas, se llevó a cabo un estudio de la aplicación directa y a nivel básico de la librería de las redes neuronales utilizando Matlab y su aplicación al problema de predicción de localización. También se realizó una evaluación del número de nodos de las redes neuronales como modelo de predicción de localización. Como conclusión se obtiene que, a medida que el conjunto de entrenamiento es mayor, la estructura de la red neuronal es más complicada.
- Se concluye que se obtienen mejores resultados en cuanto estructura de red neuronal y coste computacional cuando se trabaja con la señal de amplitud como vector de entrada frente a la fase. Ya en [39], se presenta un esquema de entrenamiento de *deep learning*, utilizando una red neuronal profunda de 4 capas ocultas centrándose en el uso de las amplitudes de las subportadoras.
- Cuando se utiliza la fase como vector de entrada se ha observado que habría que buscar estructuras de red neuronal más complejas para obtener mejores resultados. La

orientación del ESP32 y su influencia en la fase dentro del mismo punto, es uno de los puntos que se tendría que analizar.

Podemos afirmar que los objetivos planteados desde un principio han quedado definidos en este trabajo. Se inició partiendo de un análisis de múltiples referencias bibliográficas para el estudio de CSI, de informes o repositorios donde se hacía uso del microcontrolador ESP32 como sensor de la señal CSI.

Cabe destacar que, durante el desarrollo de este proyecto, surgieron algunos contratiempos que retrasaron la toma de medidas, por ejemplo, la ambigüedad de algunas referencias bibliográficas en el tipo de microcontrolador ESP32 a emplear. Además, fue necesario, para generar tráfico en la red, solicitar un préstamo de un portátil en la biblioteca. Por último, para la realización de medidas fue necesario un espacio vacío de personas, lo que, en algunos casos, supuso la repetición de medidas.

Cabe añadir que, debido a que al aumentar tiempo para entrenar la red neuronal a medida que aumenta el número de nodos y el fichero de medidas, no se realizaron más experimentos y éstos se limitaron al pasillo.

En cuanto a líneas futuras del proyecto es destacable que, este trabajo se podría complementar con la aplicación de distintas técnicas de machine learning, como pueden ser los árboles de decisión, SVM, redes neuronales más complejas, deep learning, entre otros. Además, sería posible contrastar las medidas con otros sistemas de medición CSI y, de esta forma, conocer la fiabilidad de la captura realizada por el microcontrolador ESP32. Las medidas tomadas podrían ser evaluadas con otras versiones de EPS32. También, analizar cómo influye la orientación del ESP32 en las medidas realizadas en un mismo punto. Por último, sería posible realizar medidas con otros puntos de acceso.

Bibliografía

- [1] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, y L. M. Ni, «CSI-Based Indoor Localization», *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, n.º 7, pp. 1300-1309, jul. 2013, doi: [10.1109/TPDS.2012.214](https://doi.org/10.1109/TPDS.2012.214).
- [2] «¿Qué son los sistemas de localización en interiores?», *ABAMobile*, 21 de julio de 2021. <https://abamobile.com/web/que-son-sistemas-localizacion-en-interiores/> (accedido 10 de julio de 2022).
- [3] W. Liu *et al.*, «Survey on CSI-based Indoor Positioning Systems and Recent Advances», en *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, sep. 2019, pp. 1-8. doi: [10.1109/IPIN.2019.8911774](https://doi.org/10.1109/IPIN.2019.8911774).
- [4] SITUM, «Sistemas de localización en interiores», *Situm*, 24 de febrero de 2022. <https://situm.com/es/blog/posicionamiento-en-interiores/sistemas-de-localizacion-en-interiores/> (accedido 10 de julio de 2022).
- [5] «Wi-ESP—A tool for CSI-based Device-Free Wi-Fi Sensing (DFWS) | Journal of Computational Design and Engineering | Oxford Academic». <https://academic.oup.com/jcde/article/7/5/644/5837600#208232291> (accedido 10 de julio de 2022).
- [6] Z. Wang, B. Guo, Z. Yu, y X. Zhou, «Wi-Fi CSI-Based Behavior Recognition: From Signals and Actions to Activities», *IEEE Communications Magazine*, vol. 56, n.º 5, pp. 109-115, may 2018, doi: [10.1109/MCOM.2018.1700144](https://doi.org/10.1109/MCOM.2018.1700144).
- [7] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, y L. M. Ni, «CSI-Based Indoor Localization», *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, n.º 7, pp. 1300-1309, jul. 2013, doi: [10.1109/TPDS.2012.214](https://doi.org/10.1109/TPDS.2012.214).
- [8] «A novel method of Wi-Fi indoor localization based on channel state information | IEEE Conference Publication | IEEE Xplore». https://ieeexplore.ieee.org/abstract/document/7752710?casa_token=VC14jODKNpIAAAAA:j_CwtK49x7rRYMQUMmLouvVKOOxKUKHtYgn5e8DE-TVxSbVHZ049GPbCb67QCrUCgp6KQg (accedido 10 de julio de 2022).
- [9] S. Abdul Samadh, Q. Liu, X. Liu, N. Ghourchian, y M. Allegue, «Indoor Localization Based on Channel State Information», en *2019 IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet)*, ene. 2019, pp. 1-4. doi: [10.1109/WISNET.2019.8711803](https://doi.org/10.1109/WISNET.2019.8711803).
- [10] Z. Yang, Z. Zhou, y Y. Liu, «From RSSI to CSI: Indoor localization via channel response», *ACM Comput. Surv.*, vol. 46, n.º 2, pp. 1-32, nov. 2013, doi: [10.1145/2543581.2543592](https://doi.org/10.1145/2543581.2543592).
- [11] X. Tian, S. Zhu, S. Xiong, B. Jiang, Y. Yang, y X. Wang, «Performance Analysis of Wi-Fi Indoor Localization with Channel State Information», *IEEE Transactions on Mobile Computing*, vol. 18, n.º 8, pp. 1870-1884, ago. 2019, doi: [10.1109/TMC.2018.2868680](https://doi.org/10.1109/TMC.2018.2868680).

- [12] «Indicador de fuerza de la señal recibida», *Wikipedia, la enciclopedia libre*. 18 de febrero de 2021. Accedido: 10 de julio de 2022. [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=Indicador_de_fuerza_de_la_se%C3%B1al_recibida&oldid=133320567
- [13] A. A. Ghany, B. Uguen, y D. Lemur, «A Robustness Comparison of Measured Narrowband CSI vs RSSI for IoT Localization», en 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), nov. 2020, pp. 1-5. doi: 10.1109/VTC2020-Fall49728.2020.9348854.
- [14] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, y L. M. Ni, «CSI-Based Indoor Localization», *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, n.o 7, pp. 1300-1309, jul. 2013, doi: [10.1109/TPDS.2012.214](https://doi.org/10.1109/TPDS.2012.214).
- [15] «Información del estado del canal (Channel state information) - les.wiki». https://www.pecestopica.les.wiki/detial/Channel_state_information (accedido 10 de julio de 2022).
- [16] «Información del estado del canal Diferentes tipos de información sobre el estado del canal.y Descripción matemática». https://hmong.es/wiki/Channel_state_information (accedido 10 de julio de 2022)
- [17] <https://ece3115.elimu.net/Books/WiMAX/History%20of%20IEEE%20801-11%20Standardization.pdf>
- [18] R. Crepaldi, J. Lee, R. Etkin, S.-J. Lee, y R. Kravets, «CSI-SF: Estimating wireless channel state using CSI sampling & fusion», en 2012 Proceedings IEEE INFOCOM, mar. 2012, pp. 154-162. doi: 10.1109/INFCOM.2012.6195523.
- [19] C. V. Vargas y W. E. Lopez, «Sistemas de Comunicación Inalámbrica», vol. 3, p. 17, 2007.
- [20] «Single-site indoor fingerprint localization based on MIMO-CSI | IEEE Journals & Magazine | IEEE Xplore». <https://ieeexplore.ieee.org/document/9521175> (accedido 10 de julio de 2022).
- [21] X. Sun, X. Gao, G. Y. Li, y W. Han, «Fingerprint Based Single-Site Localization for Massive MIMO-OFDM Systems», en *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, dic. 2017, pp. 1-7. doi: [10.1109/GLOCOM.2017.8254873](https://doi.org/10.1109/GLOCOM.2017.8254873)
- [22] M. Kim, D. Han, y J.-K. K. Rhee, «Machine Learning for Practical Localization System Using Multiview CSI», *IEEE Access*, vol. 8, pp. 184575-184584, 2020, doi: [10.1109/ACCESS.2020.3029598](https://doi.org/10.1109/ACCESS.2020.3029598).
- [23] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, y J. S. Rellermeyer, «A Survey on Distributed Machine Learning», *ACM Comput. Surv.*, vol. 53, n.º 2, pp. 1-33, mar. 2021, doi: [10.1145/3377454](https://doi.org/10.1145/3377454).

- [24] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, y J. S. Rellermeier, «A Survey on Distributed Machine Learning», *ACM Comput. Surv.*, vol. 53, n.º 2, pp. 1-33, mar. 2021, doi: [10.1145/3377454](https://doi.org/10.1145/3377454).
- [25] M. W. Gardner y S. R. Dorling, «Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences», *Atmospheric Environment*, vol. 32, n.º14, pp. 2627-2636, ago. 1998, doi: [10.1016/S1352-2310\(97\)00447-0](https://doi.org/10.1016/S1352-2310(97)00447-0)
- [26] «IBM Docs», 17 de agosto de 2021. <https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/es/spss-modeler/SaaS?topic=networks-neural-model> (accedido 15 de julio de 2022).
- [27] D. E. Bolanakis, «A Survey of Research in Microcontroller Education», *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, vol. 14, n.º 2, pp. 50-57, may 2019, doi: [10.1109/RITA.2019.2922856](https://doi.org/10.1109/RITA.2019.2922856).
- [28] «ESP8266 todo lo que necesitas saber del módulo WiFi para Arduino». https://programarfacil.com/podcast/esp8266-wifi-coste-arduino/#Que_es_el_ESP8266 (accedido 10 de julio de 2022).
- [29] «ESP32 Wifi + Bluetooth en un solo lugar», 16 de febrero de 2021. <https://programarfacil.com/esp8266/esp32/> (accedido 10 de julio de 2022).
- [30] «Probamos el nuevo ESP32-C3, el sustituto del ESP8266», *Luis Llamas*. <https://www.luisllamas.es/probamos-el-nuevo-esp32-c3-el-sustituto-del-esp8266/> (accedido 10 de julio de 2022).
- [31] https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- [32] A. Maier, A. Sharp, y Y. Vagapov, «Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things», en *2017 Internet Technologies and Applications (ITA)*, sep. 2017, pp. 143-148. doi: [10.1109/ITECHA.2017.8101926](https://doi.org/10.1109/ITECHA.2017.8101926).
- [33] «Router inalámbrico Smart Wi-Fi de doble banda AC1900 Linksys WRT1900AC», *Linksys*. <http://www.linksys.com/es/wireless-routers/wrt-wireless-routers/linksys-wrt1900ac-ac1900-dual-band-wifi-router/p/p-wrt1900ac/> (accedido 12 de julio de 2022).
- [34] «Get Started - ESP32 - — ESP-IDF Programming Guide release-v4.3 documentation». <https://docs.espressif.com/projects/esp-idf/en/release-v4.3/esp32/get-started/index.html> (accedido 10 de julio de 2022).
- [35] «GitHub - StevenMHernandez/ESP32-CSI-Tool: Extract Channel State Information from WiFi-enabled ESP32 Microcontroller. Active and Passive modes available. (https://stevenmhernandez.github.io/ESP32-CSI-Tool/)». <https://github.com/StevenMHernandez/ESP32-CSI-Tool> (accedido 10 de julio de 2022).

[36] «Deep Learning Toolbox». <https://es.mathworks.com/products/deep-learning.html> (accedido 27 de julio de 2022).

[37] «Descripción del producto MATLAB - MATLAB & Simulink - MathWorks España». https://es.mathworks.com/help/matlab/learn_matlab/product-description.html (accedido 18 de agosto de 2022).

[38] *ESP-IDF Eclipse Plugin*. Espressif Systems, 2022. Accedido: 10 de julio de 2022. [En línea]. Disponible en: <https://github.com/espressif/idf-eclipse-plugin>

[39] Wang, X.; Gao, L.; Mao, S.; Pandey, S. CSI-based fingerprinting for indoor localization: A deep learning approach. *IEEE Trans. Veh. Technol.* 2017, 66, 763–776.

Presupuesto

A continuación, se realizará un presupuesto de lo que sería el coste total de la realización de este proyecto. Para ello, se analizarán los siguientes puntos, se les aplicarán los impuestos vigentes a cada uno de ellos y, finalmente se calculará el coste total del Trabajo de Fin de Grado.

- Recursos materiales.
- Tarifa por tiempo empleado.
- Material fungible.
- Redacción de documentación.
- Derechos de visado del COITT.
- Gastos de tramitación y envío.

P.1 Recursos materiales.

Para el posible desarrollo de este proyecto han sido requeridos ciertos recursos hardware y software, que, dependiendo de si se necesita de licencia o no, pueden tener costes asociados.

En cuanto a los recursos hardware se pueden destacar los dos portátiles utilizados, uno que se utilizó para la generación de tráfico con la fuente que se querían tomar las medidas, mientras que el otro se empleó para la toma de medidas y la realización de la documentación de la memoria. Con respecto al software, cabe destacar la licencia necesaria para la utilización de Matlab y el paquete de Microsoft Office requerido para la realización de la memoria. En ambos casos, al poseer la licencia de estudiante ofrecida por la universidad no suponen ningún coste añadido.

A la hora de realizar el cálculo de la amortización, se supone el sistema de amortización como lineal, se toma que el inmovilizado material se desprecia de manera constante a lo largo de su vida útil. Para calcular la cuota de amortización anual se utiliza la siguiente fórmula:

$$Cuota\ anual = \frac{(Valor\ de\ adquisición - Valor\ residual)}{Años\ de\ vida\ útil}$$

Teniendo en cuenta que, el valor residual consiste en el valor estimado que podría tener el material una vez haya finalizado su vida útil y que, la vida útil estimada del material empleado en este proyecto es de cuatro años. Con estos datos se procederá a realizar los cálculos de la cuota anual.

Además, el tiempo empleado para el desarrollo de este proyecto es de cinco meses, de esta forma se calculará el valor de adquisición y su amortización para esos cinco meses.

| Elemento | Valor de adquisición (€) | Amortización (€) |
|--|--------------------------|------------------|
| MacBook Pro ,2,7 GHz Intel Core i5 de doble núcleo, 8 GB RAM | 1490,00 | 237,50 |
| Rúter Linksys WRT 1900 ACS | 387,00 | 50,50 |
| Dell, Intel Core i3 7 ^a generación, 8 GB RAM | 480,00 | 85,00 |
| Office 365 (licencia ULPGC) | 0,00 | 0,00 |
| MATLAB (licencia ULPGC) | 0,00 | 0,00 |
| Total | | 377,50 |

Tabla 4. Tabla de amortización.

De esta forma se concluye que el coste total de gastos en recursos materiales asciende a un valor de trescientos setenta y siete euros con cincuenta céntimos.

P.2 Tarifa por tiempo empleado.

En este apartado se tienen en cuenta los gastos relacionados con la mano de obra, los cálculos se harán por hora de trabajo de un Ingeniero Técnico de Telecomunicación.

Según la tabla de retribuciones de personal contratado para proyectos de investigación establecida por la ULPGC en el año 2021, con una dedicación de 20 horas semanales, el salario

sería de 1078,68 € mensuales. Por lo que se estima una retribución de 13,48 €/h. Debido a que las horas dedicadas para la completa realización de este TFG es de 300 horas, se calcula el coste total por tiempo empleado.

$$13,48 \cdot 300 = 4044,00 \text{ €}$$

Finalmente, se obtiene que el trabajo tarifado por tiempo empleado asciende a un valor de cuatro mil cuarenta y cuatro euros.

P.3 Material Fungible.

En este proyecto no se contempla ningún tipo de gasto por edición de documento ni por material de oficina, por lo tanto, el costo relacionado con el material fungible es de cero euros.

P.4 Redacción de la documentación.

Con respecto al coste de la redacción del documento se utiliza la siguiente fórmula:

$$R = 0,07 * P * C_n$$

Teniendo en cuenta que:

- R, son los honorarios por la redacción del documento.
- P, es el presupuesto.
- C_n , es el coeficiente de ponderación en función del presupuesto.

El coste final es igual a la suma de los costes de trabajo tarifado por tiempo empleado, calculado anteriormente, y de la amortización del inmovilizado material. En la Tabla 4 se puede observar el cálculo de este presupuesto.

| Concepto | Coste (€) |
|---|-----------|
| Tarificación por tiempo empleado | 4.044,00 |
| Amortización del inmovilizado material | 377,50 |
| Total | 4.421,50 |

Tabla 5. Cálculo del presupuesto para la redacción del documento

Debido a que el coeficiente de ponderación para presupuestos menores de 30.050,00€ está definido por el COITT con un valor de 1.00, el coste de la redacción de documento del TFG es de:

$$R = 0,07 \cdot 4421,5 \cdot 1 = 309,50 \text{ €}$$

Finalmente, el coste de la redacción del proyecto se queda con un valor de trescientos nueve euros con cincuenta céntimos.

P.5 Derechos de visado del COITT.

El COITT tiene establecido que, para proyectos técnicos que son de carácter general, los derechos de visado para 2020 se calculan mediante la siguiente fórmula:

$$V = 0,006 * P_1 * C_1 + 0,003 * P_2 * C_2$$

Teniendo en cuenta que:

- V , coste de visado de trabajo.
- P_1 , presupuesto del proyecto.
- C_1 , coeficiente reductor en función del presupuesto.
- P_2 , presupuesto de ejecución material correspondiente a la obra civil.
- C_2 , coeficiente reductor en función a P_2 .

El valor del presupuesto del proyecto, P_1 , se calcula sumando los costes del trabajo tarifado por tiempo empleado, la amortización del inmovilizado material y a la redacción del documento. Mientras que, de manera similar al apartado anterior, se obtiene que el coeficiente reductor en función del presupuesto tiene un valor de 1,00 debido a que el proyecto tiene un presupuesto inferior a 30.050,00 €. Como no se requiere ningún tipo de obra civil el valor de P_2 se corresponde a 0,00 €. En la tabla 5 se puede observar este cálculo.

| Concepto | Coste (€) |
|---|-----------|
| Tarificación por tiempo empleado | 4.044,00 |
| Amortización del inmovilizado material | 377,50 |
| Redacción del trabajo | 309,50 |
| Total = P1 | 4.731,00 |

Tabla 6. Presupuesto P_1

$$V = 0,006 \cdot 4731 \cdot 1 = 28,38 \text{ €}$$

Con esto se obtiene los costes de visado de este TFG son de veintiocho euros con treinta y ocho céntimos.

P.6 Gastos de tramitación y envío.

Según el COITT, cada documento visado por vía telemática posee un coste de seis euros.

P.7 Aplicación de impuestos y coste total.

Además, al presupuesto del desarrollo de este proyecto se le debe aplicar el Impuesto General Indirecto Canario (I.G.I.C.) el cual posee un valor de un 7%. En la tabla 6 se puede observar como afecta el valor de este impuesto al presupuesto total del presente TFG.

| Concepto | Coste (€) |
|---|-----------|
| Tarifificación por tiempo empleado | 4.044,00 |
| Amortización del inmovilizado material | 377,50 |
| Redacción del trabajo | 309,50 |
| Derechos de visado del COITT | 28,38 |
| Gastos de tramitación y envío | 6,00 |
| Subtotal | 4.765,38 |
| I.G.I.C (7%) | 333,57 |
| Total | 5.098,95 |

Tabla 7. Coste total

Finalmente se ha obtenido que el coste total para el desarrollo del TFG obtención de la información del estado del canal (CSI) en redes Wi-Fi haciendo uso del microcontrolador ESP32 asciende a un valor de: cinco mil noventa y ocho euros con noventa y cinco céntimos.