



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Desarrollo de un dispositivo de seguimiento de obras de
arte para el campus de la UPV

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Alcuña Alepuz, Jose

Tutor/a: Sánchez López, Miguel

CURSO ACADÉMICO: 2021/2022

Resumen

Este proyecto consiste en el desarrollo de un sistema que permita la localización de un dispositivo usando las tecnologías disponibles por el entorno, en este caso se busca utilizar una red eduroam para realizar el desarrollo.

El proyecto se ha centrado en la captura de las señales de los puntos de acceso para determinar la localización exacta del dispositivo que escanea estos puntos. También se tienen en cuenta los distintos algoritmos para el cálculo de esta posición y las técnicas más utilizadas en otros proyectos antecedentes a este.

En el documento se describirán los detalles del proceso que se ha seguido para obtener el resultado deseado, así como las opciones de herramientas disponibles para obtenerlo y los obstáculos que se han presentado. También se incluye una comparativa de los dispositivos más óptimos que se han analizado y son económicamente asequibles en el mercado, nombrando sus características más notables e interesantes para este proyecto.

Finalmente se sacan conclusiones respecto al proyecto hecho, así como indagar en las características de implementar un sistema parecido pero a gran escala dentro de unas instalaciones universitarias.

Palabras clave: localización, eduroam, punto de acceso, instalaciones universitarias.

Resum

Aquest projecte consisteix en el desenvolupament d'un sistema que permeti la localització d'un dispositiu usant les tecnologies disponibles per l'entorn, en aquest cas es busca utilitzar una xarxa eduroam per realitzar el desenvolupament.

El projecte s'ha centrat en capturar els senyals dels punts d'accés per determinar la localització exacta del dispositiu que escaneja aquests punts. També es tenen en compte els diferents algorismes per al càlcul d'aquesta posició i les tècniques més utilitzades en altres projectes anteriors a aquest.

Al document es descriuran els detalls del procés que s'ha seguit per obtenir el resultat desitjat, així com les opcions d'eines disponibles per obtenir-lo i els obstacles que s'han presentat. També s'inclou una comparativa dels dispositius més òptims que s'han analitzat i són econòmicament assequibles al mercat, anomenant-ne les característiques més notables i interessants per a aquest projecte.

Finalment se'n treuen conclusions respecte al projecte fet, així com indagar en les característiques d'implementar un sistema semblant però a gran escala dins unes instal·lacions universitàries.

Paraules clau: localització, eduroam, punt d'accés, instal·lacions universitàries.

Abstract

This project consists of the development of a system that allows the location of a device using the technologies available in the environment, in this case it seeks to use an eduroam network to carry out the development.

The project has focused on capturing signals from access points to determine the exact location of the device that scans these points. The different algorithms for calculating this position and the techniques most used in other projects that preceded this one are also taken into account.

The document will describe the details of the process that has been followed to obtain the desired result, as well as the tool options available to obtain it and the obstacles that have arisen. A comparison of the most optimal devices that have been analyzed and are economically affordable in the market is also included, naming their most notable and interesting features for this project.

Finally, conclusions are drawn regarding the project done, as well as investigating the characteristics of implementing a similar system but on a large scale within university facilities.

Keywords: location, eduroam, access point, university facilities.

Tabla de contenidos

Resumen	2
Resum	3
Abstract	4
Tabla de contenidos	5
Tabla de figuras	7
1. Introducción	8
1.1 Motivación	8
1.2 Objetivos	8
1.3 Estructura de la memoria	9
2. Estado del arte	10
2.1 Soluciones similares	10
2.1.1 UBICAPP	10
2.1.2 Keep Track	12
2.2 Microcontroladores	14
2.3 Arduino	14
3. Estudio previo	16
3.1 WiFi y eduroam	16
3.2 Access Point (AP) o punto de acceso	17
3.3 Dispositivos disponibles	18
3.3.1 ESP8266	18
3.3.2 ESP32	19
3.3.3 Raspberry Pi Pico W	20
3.4 Algoritmos de localización	21
3.4.1 TOA/TDOA	21
3.4.2 RSSI (Received Signal Strength Indication)	22
3.4.3 Media aritmética	23
3.4.4 AOA (Angle of arrival)	23
3.5 Técnicas de localización	24
3.5.1 SignPost	24



3.5.2 Triangulación	24
3.5.3 Trilateración	24
3.5.4 Multilateración	24
3.5.5 Fingerprint	24
3.5.6 Probabilidad	24
3.6 Lenguajes de programación	24
3.6.1 IDE Arduino	25
3.6.2 PHP	27
3.7 Servidor	27
3.7.1 Plesk	27
4. Descripción del sistema	29
4.1 Escaneo de los AP	33
4.2 Inserción y cálculo de datos	38
5. Resultados obtenidos	42
5.1 Propuestas de mejora	45
6. Conclusión	48
7. Bibliografía	50
8. Anexos	51
8.1 OBJETIVOS DE DESARROLLO SOSTENIBLE	51

Tabla de figuras

Figura 1. Imagen de checador en la aplicación UBICAPP	12
Figura 2. Mapa de la localización en UBICAPP	13
Figura 3. Mis localizadores	14
Figura 4. Detalles de un localizador en el mapa	14
Figura 5. Figura 5. Arduino UNO	16
Figura 6. Esquema de dispositivos de una red doméstica con Router sin AP incorporado	18
Figura 7. Placa ESP32	21
Figura 8. Ejemplo algoritmo TOA/TDOA	23
Figura 9. Ejemplo algoritmo RSSI	24
Figura 10. Ejemplo algoritmo AOA	25
Figura 11. IDE Arduino	27
Figura 12. Diseño general del proyecto	30
Figura 13. phpMyAdmin.	31
Figura 14. Diagrama de código Arduino a seguir	33
Figura 15. Diagrama del código de la página web a seguir junto al servidor	34
Figura 16. Importación de las librerías y definición de variables estáticas	35
Figura 17. Programa setup para escaneo y envío de datos	39
Figura 18. Código fuente del sitio web con método GET	42
Figura 19. Código fuente página web con tablas de posición	45
Figura 20. Representación gráfica de las tablas con el recorrido de un dispositivo	45
Figura 21. Comparación en Google Maps de la localización real tomada y la calculada	46
Figura 22. Ejemplo de solución con plano svg	47



1. Introducción

1.1 Motivación

En la actualidad se dispone de una gran cantidad de dispositivos que permiten la geolocalización por satélite orientadas a ser utilizadas por los usuarios en muchas de las acciones que se realizan diariamente.

Es bastante común utilizar este tipo de herramientas en dispositivos móviles, sin embargo cuando se diseña un dispositivo concreto para el uso de esta tecnología el precio de venta al usuario suele ser elevado, como por ejemplo, un dispositivo GPS para la conducción o dispositivos de seguimiento para deportistas de montaña.

Ya que generalmente las aplicaciones que utilizan este tipo de tecnologías vienen dadas para dispositivos móviles, GPS o relojes inteligentes, no se pueden atribuir a grandes proyectos donde se necesite una gran cantidad de dispositivos localizadores. Debido a esta carencia en el mercado se ha propuesto el desarrollo del proyecto, el cual va enfocado a la creación de un dispositivo que contemple estas características.

1.2 Objetivos

El objetivo de este proyecto es desarrollar un dispositivo de seguimiento de obras de arte para el campus de la UPV.

Posteriormente, con la explotación del dispositivo de seguimiento, se podría elaborar una aplicación web para mostrar gráficamente las posiciones de los distintos dispositivos dentro del campus e incluso mostrar el recorrido que ha llevado a cabo durante un periodo de tiempo.

Teniendo en cuenta las ventajas que ofrece un dispositivo de seguimiento para obras de arte, será necesario optimizar este mismo para brindar una experiencia duradera y con unos datos resultantes precisos. Estos datos deben poder gestionarse dentro de una base de datos alojada en un servidor para realizar informes o crear una solución donde mostrar los resultados obtenidos.

Para lograr los objetivos, es necesario establecer un punto de partida y seguir una ruta correspondiente a los siguientes subobjetivos:

- La búsqueda de un dispositivo con los componentes hardware necesarios que se adapte a las necesidades del proyecto.
- Analizar las diferentes herramientas que ofrecen los dispositivos y optimizar el uso de estas para obtener mejores resultados.
- Comparar las diferentes soluciones software para el cálculo de la posición.
- Analizar el comportamiento del dispositivo en el campus de la UPV.
- Diseñar la lógica del servidor junto con la base de datos.

1.3 Estructura de la memoria

El proyecto se divide en ocho partes. La primera parte es una introducción que cubre los objetivos generales y la estructura del informe. En cuanto al texto del segundo párrafo, se trata de un documento anterior que detalla aplicaciones similares existentes y la investigación que se está realizando sobre posibles tecnologías para la determinación de la ubicación de dispositivos. De lo contrario, el tercer párrafo enumera todo lo que se necesitará para completar el proyecto y otros posibles métodos que se pueden utilizar. La última parte de la descripción del sistema explica el diseño de la aplicación en sí, tanto en términos de arquitectura del servidor como de diseño de la base de datos, con el correspondiente diagrama entidad-relación.

Al finalizar la fase de diseño, se realizará una descripción de todas las pruebas realizadas para asegurar que se cumplen correctamente los requisitos previamente definidos y se dan resultados. Después de probar la corrección de la operación y el cumplimiento de todos los casos de uso de la aplicación desarrollada, se ejecutará. Termina con una sección de conclusión que resalta los problemas potenciales encontrados durante el desarrollo del proyecto y la vista respecto a lo diseñado y la propuesta. Además, hay un directorio que se utilizó para obtener la información necesaria para este proyecto.



2. Estado del arte

En esta parte del trabajo describiremos la investigación realizada en la cual se recopiló información sobre la existencia de tecnologías, herramientas y aplicaciones comercialmente disponibles que puedan ser similares en funcionalidad a las implementadas en este proyecto.

Esta sección se divide en dos partes, primero, discutiremos aplicaciones similares en el mercado con sus respectivas características. En segundo lugar, se describirán en detalle las diversas tecnologías que se han estudiado para permitir el reconocimiento de escritura a mano en la solución en desarrollo.

2.1 Soluciones similares

A continuación, se presentan dos soluciones desarrolladas por distintos ingenieros cuyas propuestas obtienen unos resultados satisfactorios mediante la utilización de herramientas similares a la propuesta por este proyecto.

La primera solución se trata de un sistema de localización utilizando una interfaz de aplicación móvil para vehículos en aparcamientos. La segunda solución trata de una aplicación la cual indica la ubicación de un dispositivo concreto en interiores conociendo los puntos de acceso de las proximidades. Estas dos soluciones muestran diferentes aplicaciones atribuidas al uso de herramientas que permiten la localización mediante distintos métodos y cómo se han aplicado en cada caso concreto, mostrando similitudes en todos los ámbitos que se van a mostrar en el siguiente trabajo.

2.1.1 UBICAPP

UBICAPP es una aplicación para móviles que se engendró mediante el desarrollo de un sistema de geolocalización de vehículos u objetos sirviéndose de un sistema de posicionamiento global mediante satélites.

Esta se construyó basándose en un sistema formado por un dispositivo dotado de la capacidad de realizar la geolocalización obteniendo las coordenadas precisas de su posición y comunicándose con la aplicación móvil como canal de comunicación del sistema con el usuario.

En el desarrollo de UBICAPP, la placa Raspberry Pi 3 se utiliza como unidad de procesamiento para comunicarse con el módulo de posicionamiento y el módulo de comunicación remota a través del sistema de comunicación móvil global, que a su vez se comunica a través de satélites. La programación del dispositivo se realizó en lenguaje Python. La aplicación para teléfonos móviles Android se desarrolló en lenguaje Java utilizando el software oficial y gratuito ofrecido por Google, Android Studio.

El uso de la aplicación permite, utilizando una cuenta con usuario y contraseña, registrar tu ubicación en unos registros de forma que son guardados de forma segura y pueden ser consultados por medio de una página web.

Este resultado de la app muestra un sistema de fácil utilización para el usuario con una aplicación con interfaz sencilla y con una curva de aprendizaje corta. Estas características la hacen destacar en el mercado por su poca complejidad en su usabilidad.

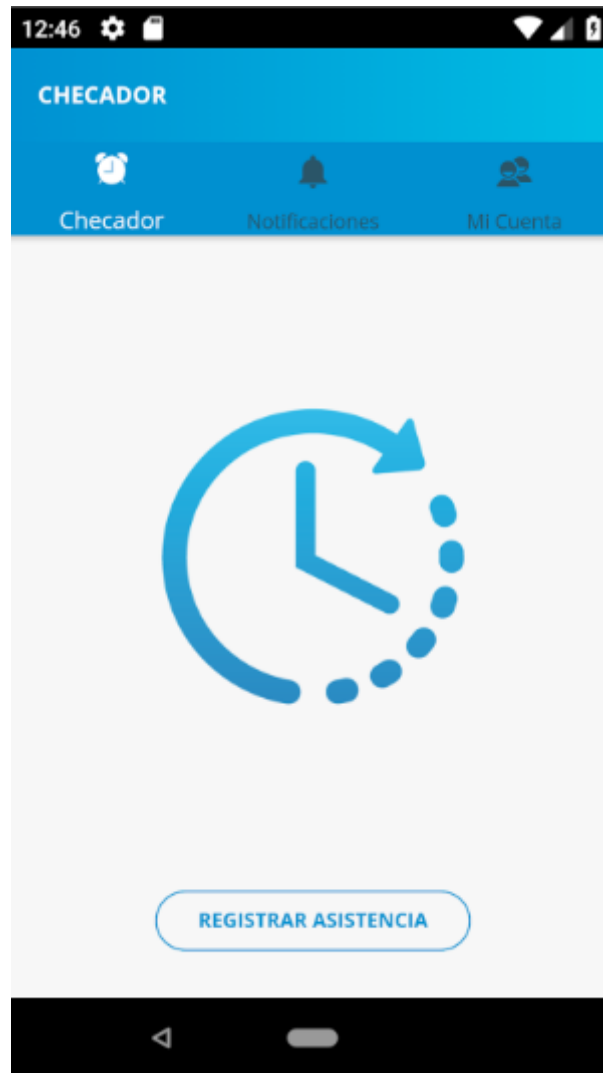


Figura 1. Imagen de checador en la aplicación UBICAPP



Figura 2. Mapa de la localización en UBICAPP

2.1.2 KeepTrack

KeepTrack es un sistema que permite a los usuarios localizar sus vehículos mediante GPS y visualizar el historial de las localizaciones de sus vehículos sobre un mapa a través de una página web. Este sistema se creó con el objetivo de que los usuarios puedan recuperar un vehículo en caso de robo, o ayudar a la policía a encontrarlo.

KeepTrack funciona gracias a dispositivos de seguimiento y aplicaciones web. El dispositivo se instala en el vehículo y utiliza tecnología GPS para determinar su ubicación. El sistema envía esta ubicación con una solicitud HTTP al servidor a través de GPRS.

El dispositivo que se utilizó es un módulo Adafruit FONA 808. Este módulo es capaz de realizar comunicaciones de datos a través de redes de telefonía de segunda generación con GPRS y utilizar el sistema de localización por satélite GPS para obtener su posición. Para controlar el módulo de Adafruit se utilizó la placa de desarrollo Arduino UNO. Para la aplicación del servidor se utilizó una arquitectura MVC en lenguaje PHP. Se puede observar como en el mapa de la aplicación aparecen los localizadores disponibles y la localización correspondiente de cada marcador en el mapa.

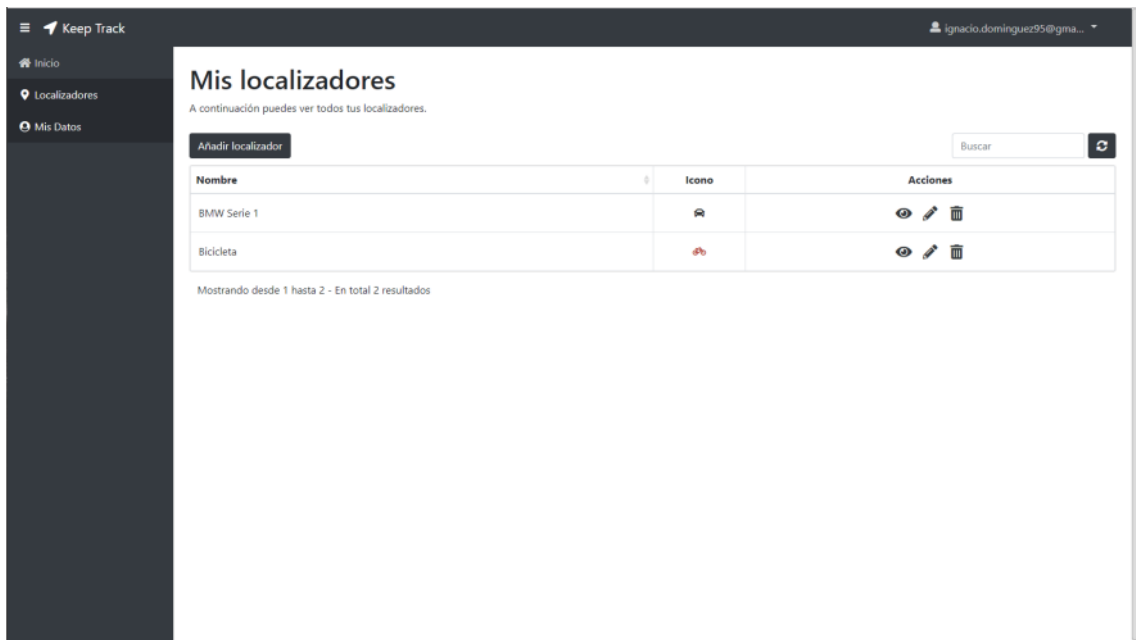


Figura 3. Mis localizadores

Se puede observar como en el mapa de la aplicación aparecen los localizadores disponibles y la localización correspondiente de cada marcador en el mapa.

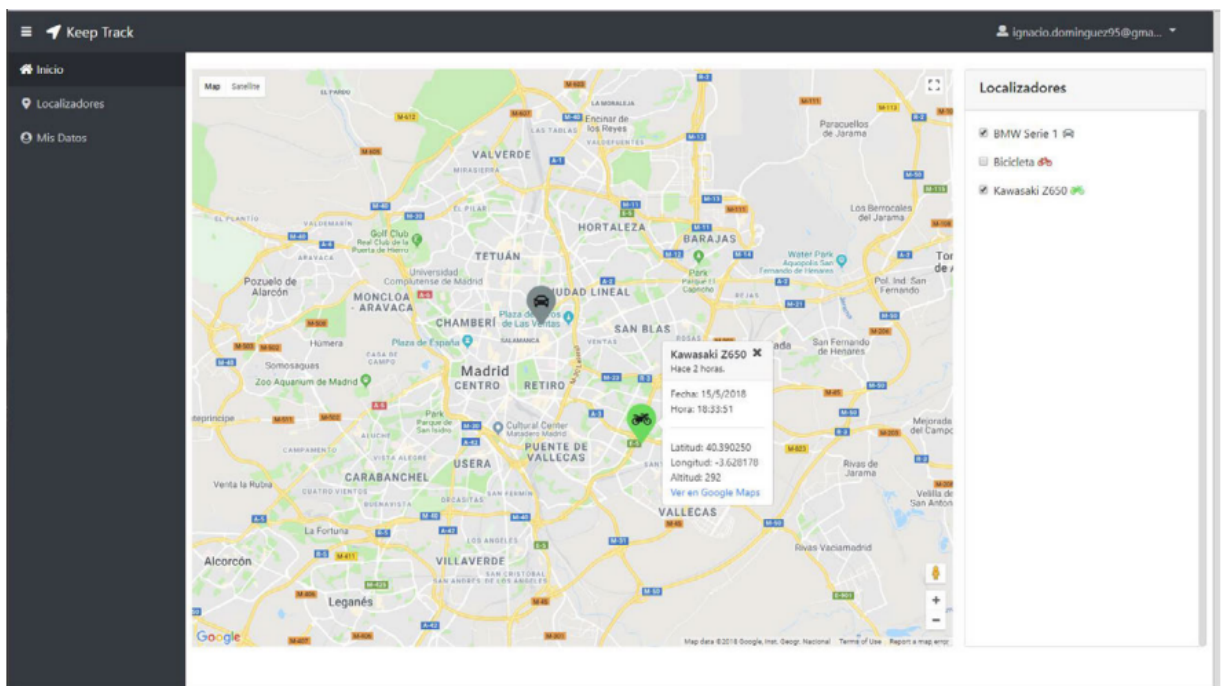


Figura 4. Detalles de un localizador en el mapa

2.2 Microcontroladores

El primer microcontrolador, según el instituto Smithsonian, fue desarrollado en el año 1971 y fue llevado a cabo por ingenieros de Texas Instruments. Sin embargo, no fue hasta 1974 que apareció por primera vez en el mercado. TMS 1000 fue el nombre que este microcontrolador recibió e incorporaba un microprocesador, memoria ROM, memoria RAM y reloj en un mismo circuito integrado. Más tarde, Intel se unió al desarrollo de microcontroladores, porque tales circuitos crearon una gran demanda para el desarrollo de sistemas integrados y sistemas de control. El desarrollo del Intel 8048 fue un éxito para la empresa y resultó en un mayor presupuesto para el desarrollo de microcontroladores.

En estas primeras etapas del desarrollo de microcontroladores, había dos variedades de memoria de programa. La memoria PROM (Programmable Read-Only Memory) que se puede programar sólo una vez, es decir, una vez ingresado el programa no se puede cambiar. Y la memoria EPROM (Erasable Programmable Read-Only Memory) que se puede reprogramar exponiendo el chip a la luz ultravioleta a través de una ventana de cuarzo.

En 1993, Microchip Corporation lanzó el microcontrolador PIC16x84 con EEPROM (memoria de solo lectura programable borrable eléctricamente). Este tipo de memoria permite una reprogramación rápida sin el uso de herramientas costosas. Son un gran paso adelante en el desarrollo de sistemas integrados, ya que permiten una creación de prototipos mucho más rápida y abren la posibilidad de lanzar actualizaciones de software después del lanzamiento del producto. También en 1993, Atmel introdujo el primer microcontrolador flash que costaba más que la EEPROM pero era más rápido.

Algunos de los microcontroladores más populares en la actualidad son productos de empresas como Microchip y Atmel (ahora propiedad de Microchip) entre otras empresas. Microchip fabrica microcontroladores de 8, 16 y 32 bits y proporciona herramientas de hardware y software para facilitar la programación y depuración de programas.

2.3 Arduino

Una de las plataformas de desarrollo más utilizadas en la actualidad es Arduino, un proyecto de código abierto que desarrolla "microcontroladores de placa única" como Arduino UNO, Arduino NANO o Arduino MEGA.

Estas placas se basan en microcontroladores Atmel. Su popularidad se debe a su bajo costo y creación de prototipos muy rápida, así como a una comunidad de usuarios muy grande donde puede compartir diseños y cumplir con los pedidos. Dado que en este proyecto se utilizó la placa de desarrollo Arduino UNO, a continuación se explican sus funciones principales.

La placa Arduino Uno se basa en el microcontrolador Atmel ATmega328P con 32 KB de Flash y 1 KB de EEPROM. Como se muestra en la figura. 2-2, la placa proporciona fácil acceso a catorce pines de entrada y salida, seis de los cuales están dedicados a salidas PWM (modulación de ancho de pulso) y seis a entradas analógicas. Disponible para microcontroladores.



Figura 5. Arduino UNO

3. Estudio previo

Hay muchas maneras de realizar la localización de un dispositivo. La manera más común es usando GPS, pero en este proyecto de desarrollo la localización se realizará en unas condiciones concretas donde existen espacios cerrados y abiertos dentro de una zona delimitada por la conexión a la red eduroam.

Si se desea saber la posición del dispositivo, el primer pensamiento debe ser tener 3 o más puntos, a partir de estos puntos se calculará la distancia de cada punto al dispositivo y se triangulará su posición.

Pero se debe de barajar qué tecnología utilizar para el dispositivo que se utilizará en este proyecto.

Se pueden encontrar diferentes tipos de tecnologías como: Wi-Fi, Zigbee, Infrarrojos, RFID (Identificación por Radio Frecuencia), entre otras opciones. Entre estas tecnologías se utilizará la conocida tecnología Wi-Fi. La tecnología está presente en el día a día, por lo que es fácil de usar y gratuita.

Si se tiene en cuenta que se utilizará la tecnología Wifi, ahora se debe considerar cómo se capturarán los datos. Es decir, ¿Qué algoritmo de localización se aplicará? Esto es importante porque el nivel de calibración variará según el método. Es por ello que, tras investigar un poco, se han encontrado 3 tipos de algoritmos que son eficientes:

- TOA/TDOA (Time [difference] of Arrival)
- RSSI (Received Signal Strength Indication)
- Media aritmética de la longitud y latitud (para grandes cantidades de datos)
- AOA (Angle of arrival)

Debido a la tecnología Wi-Fi utilizada, se recomiendan puntos de acceso (AP) completos. Los puntos de acceso no son lo mismo que los enrutadores; Esto se explicará con más detalle más adelante.

Una vez que haya definido sus algoritmos y métodos de localización, debe definir el lenguaje de programación que está utilizando. Dado que uno de los objetivos finales es mostrar los resultados en la página web, el código que ejecuta las pruebas y el código que se ejecuta en la aplicación web deben ser diferentes.

Esto último también incluye la creación o uso de un servidor que será utilizado para procesar información por ambas partes. Después de haber encontrado todo lo necesario para este proyecto, cada elemento se definirá más en detalle.

3.1 WiFi y eduroam

Wifi es una tecnología para WLAN (Wireless Local Area Network) cuyos dispositivos se basan en el estándar IEEE 802.11. La tecnología surge de la necesidad de crear un mecanismo de conexión inalámbrica que sea compatible entre diferentes dispositivos y permite conectarse a Internet a través de un punto de acceso.

Esta tecnología es la más utilizada, por lo que se aprovecha su libre disponibilidad para promover el avance del proyecto; aunque una de sus desventajas puede ser el alto consumo de batería.

Eduroam (Education Roaming) es un servicio que proporciona una conexión segura a internet, principalmente para la movilidad de estudiantes. A través de esta iniciativa, estudiantes, investigadores y docentes pueden disfrutar de conexiones tanto dentro de sus propios campus como en otros campus participantes. Consiste en una infraestructura basada en un sistema proxy RADIUS, que permite a cualquier usuario de Eduroam acceder a la red de cualquier institución asociada al programa.

En el caso de España, eduroam EU forma parte del proyecto RedIRIS, una red académica y de investigación española, que se encarga de prestar servicios de comunicación a la comunidad científica y universitaria, financiado por el Ministerio de Ciencia, Innovación y Universidades.

3.2 Access Point (AP) o punto de acceso

Primero, se debe hacer mención del de AP (punto de acceso) y del Router. Estos dos términos no tienen el mismo significado.

Un Router es un dispositivo de red que permite enrutar paquetes entre redes y brinda acceso a Internet en el hogar, mientras que un punto de acceso es un dispositivo conectado a un router para crear una red inalámbrica (WLAN) en un área específica. Permite que los dispositivos inalámbricos se conecten a esta red al permitir que los paquetes se enruten al enrutador. Un enrutador puede integrar un punto de acceso, por lo que los dos términos a menudo se confunden.

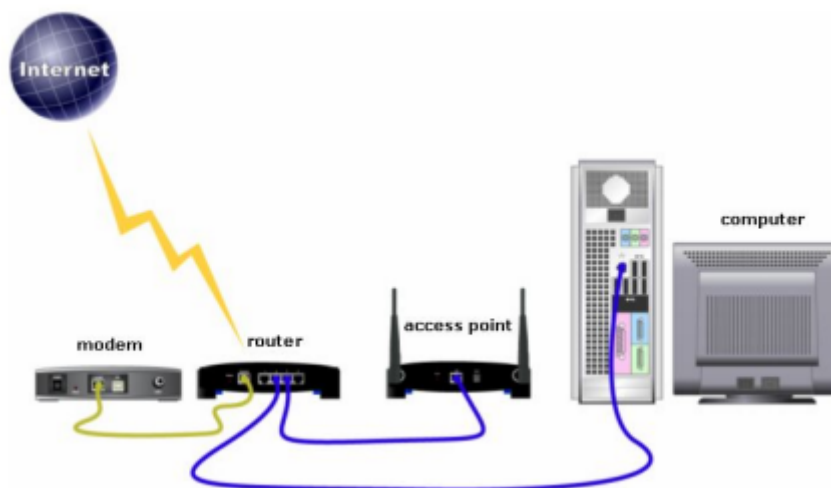


Figura 6. Esquema de dispositivos de una red doméstica con Router sin AP incorporado

3.3 Dispositivos disponibles

Los microprocesadores utilizan diferentes tipos de memoria para procesar datos. Los dispositivos de memoria y de entrada/salida están integrados en el mismo circuito. La memoria de acceso aleatorio (RAM) almacena datos antes de enviarlos, mientras que la memoria de solo lectura (ROM) almacena el sistema operativo de los nodos sensores. Los microprocesadores de los nodos sensores también se conocen como pequeñas CPU que se preocupan por la velocidad, el voltaje y el consumo de energía de la CPU. Las operaciones de los sensores se ejecutan a baja velocidad de la CPU. La mayor parte del tiempo, los sensores permanecen en modo de suspensión. Cuando el procesador está en modo de suspensión, esto no significa que no está consumiendo energía. En el modo de suspensión, participa en otras actividades como la sincronización del tiempo.

En el mercado actual se puede encontrar un gran catálogo de dispositivos con características relevantes para el desarrollo del proyecto como soporte WiFi, funciones de bajo consumo (sleep mode), de pequeñas dimensiones y de bajo coste.

Por ello se va a analizar tres dispositivos que destacan por estas características, los cuales son:

- ESP8266
- ESP32
- Raspberry Pi Pico W

3.3.1 ESP8266

Se trata de un chip compatible con el protocolo TCP/IP e integrado con conexión WiFi, creado por Espressif Systems. El objetivo principal de este chip es permitir que cualquier microcontrolador acceda a la red.

La principal ventaja del ESP8266 es su bajo consumo de energía. Ideal para dispositivos portátiles y de IoT, los expertos dicen que puede tener un gran impacto en las nuevas tendencias en muchos campos. El único inconveniente que figura es que toda la documentación está en chino, sin embargo, la gran cantidad de gente que seguía el producto comenzó a traducir la documentación y crear firmwares para el ESP8266.

En cuanto a el consumo eléctrico del dispositivo dependerá de diferentes factores como por ejemplo: el modo en el que esté trabajando el ESP8266, los protocolos que estemos utilizando,

Debido a que el ESP8266 va enfocado a los sectores wearables, dispositivos del IoT y móviles, este dispositivo requiere de una gestión eficaz de energía. Tiene una arquitectura de bajo consumo de energía que opera en 3 modos.

Modo activo o active mode: a pleno rendimiento.

Modo dormido o sleep mode: solo el RTC (Real Time Clock) funciona para mantenerse sincronizado. Permanece en modo alerta de los posibles eventos que le

hagan activarse. Mantiene en memoria los datos de conexión y para que no debas de restablecer la conexión con la WiFi. Consume desde 0,6 mA hasta 1 mA.

Deep sleep o sueño profundo: el reloj en tiempo real está encendido pero no activo. Debe pasar por la hibernación antes de despertar y se debe prestar especial atención a los datos, porque en este estado de alguna manera están apagados y se perderán todos los datos no guardados. Consume alrededor de 20 μ A.

Para su programación se requiere primero un buen conocimiento de los comandos AT, e incluso del idioma de donde proviene, ya que la documentación no estaba traducida. Esto último se resolvió gracias a la comunidad de internet.

Posteriormente, aparecieron diferentes versiones del software y traducciones de la documentación oficial. Por lo tanto, es relativamente fácil programar el ESP8266 en estos días. En cuanto al hardware, dependerá de si el modelo que estás usando incluye un adaptador serie/USB. Sin embargo, se puede programar usando un Arduino.

La versión de software predeterminada admite comandos AT (conjunto de comandos estándar). Ahora es posible programar a través del IDE oficial de Arduino con la misma sintaxis gracias al software proporcionado por NodeMCU.

Cabe añadir que GitHub cuenta con una gran cantidad de librerías que facilitarán la programación del ESP8266.

3.3.2 ESP32

ESP32 es una serie de SoC (System on Chip) y módulos, creado por Espressif Systems, caracterizado por su costo y consumo de energía bajos. Esta nueva familia se caracteriza notablemente porque además de Wi-Fi, también soporta Bluetooth y además, es la sucesora del conocido ESP8266. Se basa básicamente en microprocesadores Tensilica Xtensa LX6 (single o dual core) con una frecuencia de operación de hasta 240 MHz.

Esto permite que el dispositivo logre un consumo de energía muy bajo gracias a sus funciones de ahorro de energía, incluida la sincronización del reloj y múltiples modos de funcionamiento. Todo esto lo convierte en una herramienta ideal para proyectos alimentados por batería o aplicaciones IoT.

Este chip tiene capacidad Wi-Fi y es compatible con 802.11 b/g/n en la banda de 2,4 GHz, por lo que puede alcanzar velocidades de hasta 150 Mb/s. Además, incluye conectividad Bluetooth que es compatible con Bluetooth v4.2 y Bluetooth Low Energy (BLE). De hecho, es una unidad de radio, estrechamente vinculada con módulos de comunicación inalámbrica, que transmite y recibe información directamente. En otras palabras, recibe y convierte los datos digitales de los módulos Wi-Fi y Bluetooth en señales electromagnéticas que se envían de forma inalámbrica para comunicarse con un teléfono móvil o un router. Sin embargo, también hace lo contrario de convertir a datos digitales, que los módulos Wi-Fi y Bluetooth pueden interpretar las ondas electromagnéticas generadas por otros dispositivos.



ESP32 tiene un alto grado de integración. Su pequeño pack incluye:

- Amplificador de potencia
- Balun de RF
- Filtros y módulos de administración de energía
- Amplificador de recepción de bajo ruido
- Interruptores de antena

El ESP32 está equipado con dos procesadores de bajo consumo Tensilica Xtensa de 32 bits, así como un coprocesador de consumo ultrabajo, que se utiliza cuando el dispositivo está en modo de suspensión profunda para realizar operaciones similares a conversiones analógico-digital y otras operaciones.

Las memorias que constan en un dispositivo ESP32 son diferentes a la mayoría de las placas Arduino, aunque se puede utilizar su IDE para la parte de programación, en estas se encuentran, clasificadas en internas y externas, más tipos de memorias. La memoria interna es la memoria que ya está incluida en el SoC y la memoria externa es la memoria que se puede agregar para ampliar la capacidad del sistema. Muchas placas de desarrollo basadas en ESP32 agregan memoria externa para un mejor rendimiento del sistema.

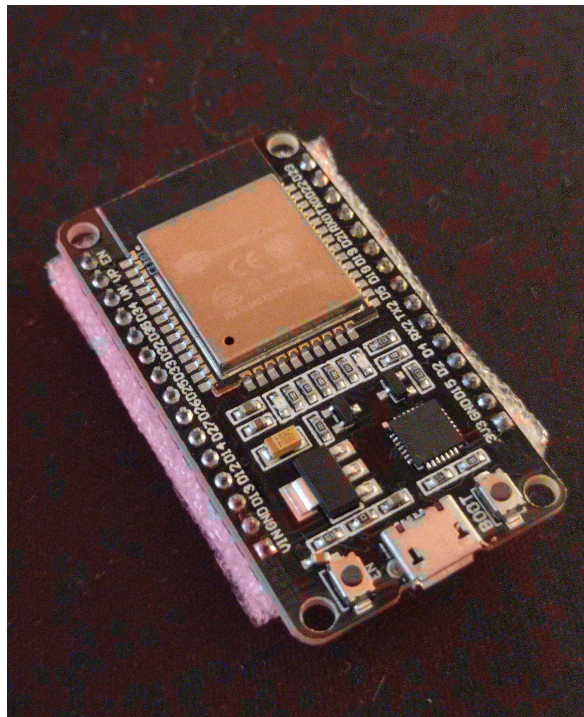


Figura 7. Placa ESP32

3.3.3 Raspberry Pi Pico W

El Raspberry Pi Pico W es un pequeño microcontrolador habilitado para Wi-Fi que, si se programa correctamente, será capaz de potenciar todo tipo de sistemas. Con este tipo de controlador, la integración con dispositivos IoT se vuelve más fácil y

versátil. Aunque lo que más destaca de este dispositivo es su económico precio, que resulta asequible para todo el mundo.

A pesar de no ser tan potente como sus hermanos mayores, el Raspberry Pi Pico W está más enfocado en la integración de IoT, porque es un microcontrolador habilitado para WiFi, pequeño y de bajo consumo. Suficiente para que cualquiera que quiera trastear pueda comprarlo y tenerlo conectado en algún lado. El nuevo Raspberry Pi Pico W ofrece conectividad 802.11n y retrocompatibilidad completa de pines con los modelos de gama alta.

Según Raspberry Pi, Pico ha vendido casi 2 millones de unidades desde su lanzamiento. Y ahora que esta versión tiene Wi-Fi actualizado, es probable que ese número crezca aún más.

Además del chip WiFi 4, la Raspberry Pi Pico W ofrece 2 MB de memoria flash y un puerto micro USB. Sus dimensiones oficiales son 51 milímetros de altura por 21 milímetros de ancho. A nivel de programación, el funcionamiento es el mismo que los modelos anteriores. El nuevo Raspberry Pi es compatible con C y tiene instalado Micro Python.

Cabe destacar que este microcontrolador tiene soporte de modo a bajo consumo "sleep".

3.4 Algoritmos de localización

Los algoritmos de ubicación son un punto importante a considerar, ya que nos esforzaremos por maximizar la precisión en función de los datos disponibles. En consecuencia, se determinarán los algoritmos a utilizar, los cuales también indicarán las ventajas y desventajas de cada algoritmo.

3.4.1 TOA/TDOA

Algoritmo que estima la posición relativa del receptor calculando la distancia entre el sensor y el terminal, utilizando la diferencia entre el tiempo de recepción (Rx) y el tiempo de transmisión (Tx) (el tiempo de viaje entre el transmisor y el receptor) para calcular la distancia. y entre sensor y terminal. El tiempo de propagación es el tiempo que tarda la señal en viajar desde el transmisor hasta el receptor y se puede calcular como (es decir, transmisor - receptor). Por tanto, sabiendo que la velocidad de propagación es aproximadamente igual a la velocidad de la luz, se puede determinar la distancia entre el emisor y el receptor.

Este método tiene dos problemas principales: primero, no hay una ruta directa entre el transmisor y el receptor, segundo, los relojes de ambos dispositivos deben estar sincronizados para obtener la distancia correcta que afecta a la bolsa.



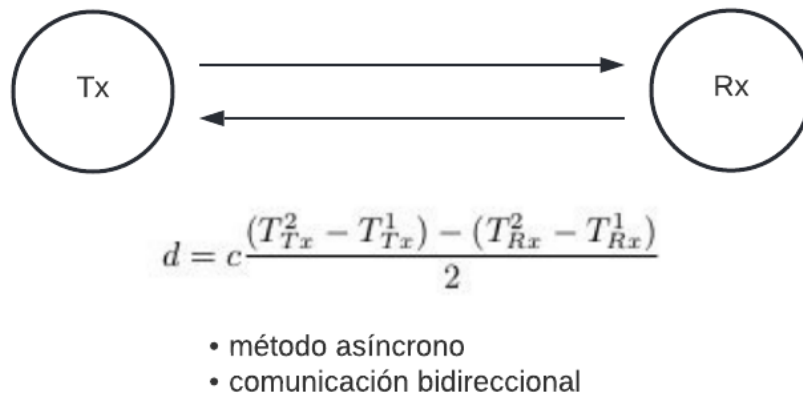


Imagen 8. Ejemplo algoritmo TOA/TDOA

3.4.2 RSSI (Received Signal Strength Indication)

Este algoritmo obtiene una estimación de la distancia en función de la potencia de la señal recibida. La relación entre potencia y distancia puede variar mucho debido a problemas como la reflexión, la refracción, la difracción y la dispersión de los objetos. Por lo tanto, se utilizarán técnicas de posicionamiento para mejorar la relación entre distancia y fuerza.

La desventaja de este algoritmo es que cada fabricante define el rango de sensibilidad del dispositivo que se utilizará para representar su rango de valores RSSI, y la correspondencia entre el valor RSSI y el valor en dbm significa que es probable que cada dispositivo proporcione un valor de salida, aunque está en la misma ubicación, por lo que es difícil conseguir que cualquier dispositivo con una tarjeta 802.11 utilice este mismo sistema para ubicar su sistema global.

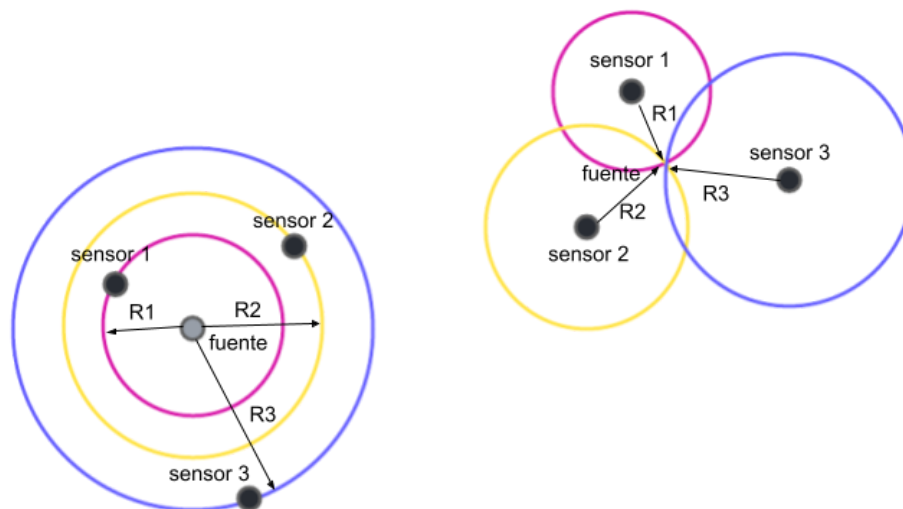


Figura 9. Ejemplo algoritmo RSSI

3.4.3 Media aritmética

Este algoritmo realiza la media aritmética de cada atributo que contiene una localización 2D, la suma total de longitudes dividido la cantidad de longitudes que hay forman el resultado de la longitud final, y la suma total de latitudes dividido la cantidad de latitudes disponibles forman el resultado de la latitud final.

La utilización de este algoritmo carece de precisión en sistemas en los que se reciben pocos datos de las localizaciones de los AP (Puntos de Acceso) al sistema de referencia, el dispositivo que recibe la información del entorno. Sin embargo, esto cambia cuando la cantidad de datos de los AP es abundante y fiable ya que la precisión del resultado final no suele tener tanta diferencia con el resultado real.

Cabe destacar que este algoritmo facilita el cálculo dejando de lado posibles errores cuadráticos.

3.4.4 AOA (Angle of arrival)

El algoritmo se basa en el ángulo de llegada de la señal desde el punto de vista del terminal utilizado. El ángulo de incidencia se calcula trazando una línea desde el receptor hasta el transmisor y otra línea desde el receptor en una dirección conocida (por ejemplo, el punto de origen). Si hay varios receptores con ubicaciones conocidas, la ubicación del transmisor se puede determinar mediante triangulación.

Su precisión aumenta con el número de antenas utilizadas y el costo. No es una buena opción en redes Wi-Fi porque el algoritmo funciona sin multicamino, porque las señales reflejadas provocan desviaciones en el ángulo de llegada.

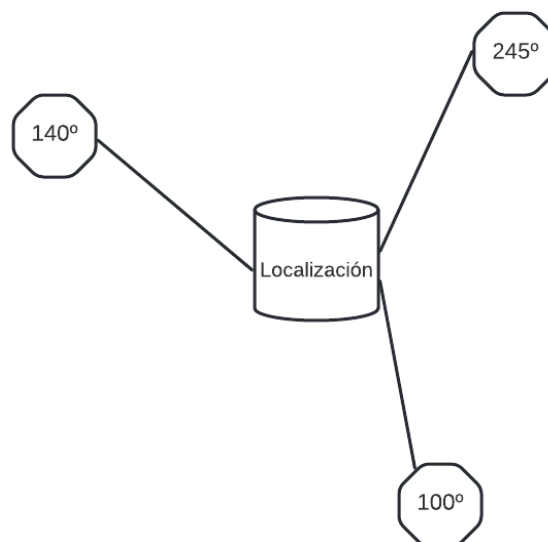


Figura 10. Ejemplo algoritmo AOA

3.5 Técnicas de localización

Este es un punto fundamental a tener en cuenta porque estos métodos combaten el efecto Multipath. Por ello, a continuación se definirá brevemente el significado de cada técnica.

3.5.1 SignPost

Este método tiene como objetivo obtener la ubicación del dispositivo en función del sensor más cercano. Esto es similar a usar el algoritmo RSSI simple. Este método puede ayudar a obtener las probables secciones siguientes dentro de un gran plano.

3.5.2 Triangulación

Este método se basa en el ángulo de incidencia (AOA), pero requiere múltiples mediciones utilizando múltiples sensores. Este método se usa a menudo en los algoritmos AOA porque más de un dispositivo puede triangular la posición de las unidades por ángulo.

3.5.3 Trilateración

La técnica se basa en la estimación de la distancia. Suele utilizarse junto con los algoritmos RSSI y TOA, en cuyo caso los elementos pertenecientes a la red deben estar sincronizados.

3.5.4 Multilateración

Este es un método más preciso que el método de trilateración, pero los cálculos son más complicados. También necesita estar sincronizado con la red porque funciona con el algoritmo TDOA (TDOA tiene mayor precisión que TOA).

3.5.5 Fingerprint

Este método requiere una calibración previa del entorno de instalación de la red de sensores como puntos de acceso, raspberrys y más, para crear un almacén de datos de huellas dactilares y utilizar estos datos como base para cálculos posteriores.

3.5.6 Probabilidad

Este método requiere pruebas preliminares. Esta solución elude la calibración utilizando heurísticas que consideran el historial de ubicación. Esto significa que si ya tenemos un historial de trabajo, se puede utilizar.

3.6 Lenguajes de programación

Será importante la parte de programación, debido a que se extraerán para su uso posterior los datos necesarios. Por consiguiente, se debe decidir qué tipo de lenguaje de programación se desea utilizar.

Dada la poca experiencia que se ha ganado durante la carrera en experimentar programación sobre microprocesadores, se indagó en las posibles soluciones disponibles para el diseño y la programación sobre el dispositivo localizador. Ya que los dispositivos descritos en este trabajo son compatibles con el IDE de Arduino y la programación en este entorno de desarrollo integrado va acompañada de una extensa documentación en la red, se decidió la utilización de esta.

También hay que tener en cuenta el tratamiento de datos que deberá realizar la parte del servidor, en este caso se tomó la decisión de realizar una página web que recabe los datos del dispositivo y realice el cálculo pertinente para poder insertarlo en una base de datos. Con dichas premisas se utilizará el lenguaje de programación PHP, aunque también se podría utilizar JavaScript.

Se va a utilizar PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) en la programación de la página web.

3.6.1 IDE Arduino

Un IDE (Entorno de Desarrollo Integrado) es un programa informático que incluye un conjunto de herramientas de programación. Puede ser específico para un solo lenguaje de programación o puede usarse para múltiples lenguajes de programación.

Un IDE es un entorno de desarrollo empaquetado como un programa de aplicación; es decir, incluye un editor de código, un compilador, un depurador y un generador de interfaz gráfica de usuario (GUI). Además incluye herramientas para cargar un programa compilado en la memoria flash del hardware, en el caso de Arduino.

Este es el aspecto del IDE:



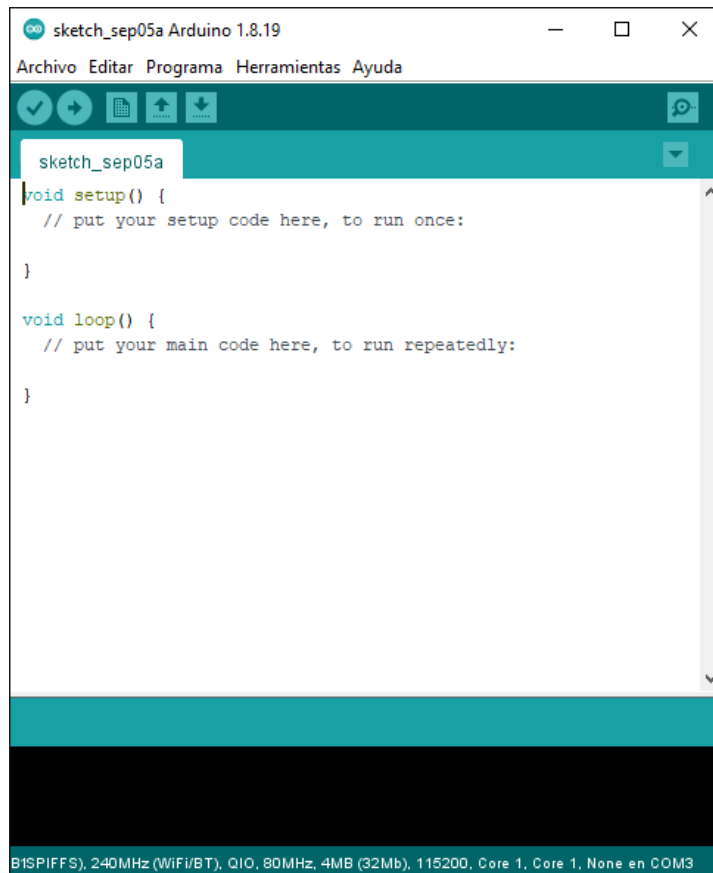


Figura 11. IDE Arduino

El IDE de Arduino es multiplataforma, lo que facilita su instalación y uso en los distintos sistemas operativos como Windows, Linux y Mac OSX. Cada uno de los menús y opciones que tiene son muy intuitivos en cuanto a su función aunque hay una descripción detallada de su funcionamiento en la documentación oficial del IDE, los más importantes ahora son:

- Botones de Verificar y Subir
- Botón Monitor Serie
- Consola de Error
- Menú herramientas Placa y Puerto
- Puerto y placa seleccionada
- Menú preferencias
- Proyecto/Sketch
- Resaltado de palabras clave

Lo que destaca de este IDE es que es un software básico ideal tanto para principiantes como expertos, por ello se ha utilizado este IDE para el desarrollo del código del manejo del microprocesador.

3.6.2 PHP

PHP es un lenguaje de código abierto muy popular que es particularmente adecuado para el desarrollo web y se puede incrustar en HTML. En lugar de usar muchos comandos para mostrar HTML (como C o Perl), las páginas PHP contienen HTML con código incrustado que realiza una función o un conjunto de funciones. La diferencia entre PHP y el lado del cliente, como JavaScript, es que su código se ejecuta en el servidor, genera HTML y lo envía al cliente. El cliente obtendrá el resultado de ejecutar el script incluso si no sabe cuál es el código. El servidor web incluso se puede configurar para usar PHP para manejar todos los archivos HTML, por lo que los usuarios no podrán ver el contenido por debajo del nivel HTML.

3.7 Servidor

La gestión de los datos entre el dispositivo localizador, la base de datos y la aplicación web debe hacerse mediante una máquina servidor que administre tanto los sitios web como las posibles soluciones para el proyecto.

En el ámbito de los sistemas TIC de la Universidad Politécnica de Valencia, existe un servicio de publicación de sitios web personales tanto para el personal docente como para los estudiantes con la tarea de diseño y mantenimiento, y también se encuentra el Servicio de Publicación Web Avanzado que permite al usuario utilizar un panel de control para gestionar su página web. Este servicio que proporciona la Universidad Politécnica de Valencia (UPV) aloja estas páginas web en su servidor interno.

El Servicio de Publicación Web Avanzado permite crear dos tipos de páginas web que se pueden solicitar, estas son o personales o institucionales, la diferencia con cada una de estas es el espacio permitido en disco, de 6GB en una personal a 10GB en una institucional. Este servicio será el que se utilizará para alojar la página web y administrar la base de datos para el presente proyecto de desarrollo, este a su vez utiliza Plesk.

3.7.1 Plesk

Plesk es la plataforma de alojamiento WebOps líder para ejecutar, automatizar y hacer crecer aplicaciones, sitios web y empresas de alojamiento. Siendo la única plataforma agnóstica del sistema operativo, Plesk se ejecuta en más de 384.000 servidores, automatizando más de 11M sitios web y más de 15M de buzones de correo.

La plataforma de alojamiento de Plesk permite de manera efectiva a los desarrolladores de aplicaciones al proporcionar acceso a una infraestructura web simple y más segura administrada por profesionales de la web y empresas de alojamiento. La visión de Plesk es elevar constantemente la rentabilidad de los clientes y socios proporcionándoles una plataforma en la nube que otorga a los desarrolladores de aplicaciones un entorno listo para codificar.

Además de simplificar la complejidad, Plesk aumenta sus esfuerzos para permitir que tanto los clientes como los socios amplíen y personalicen Plesk como una plataforma de alojamiento abierta. El rico ecosistema de extensiones de Plesk proporciona acceso a características aún más relevantes dirigidas a audiencias específicas.

4. Descripción del sistema

Para la realización del proyecto se necesita una estructura de diseño clara y sencilla.

Ya que es una demo de un sistema que se podría producir a gran escala, se ha decidido un diseño en el que se obtienen dos módulos y cada uno de ellos se encarga de una tarea de la solución final.

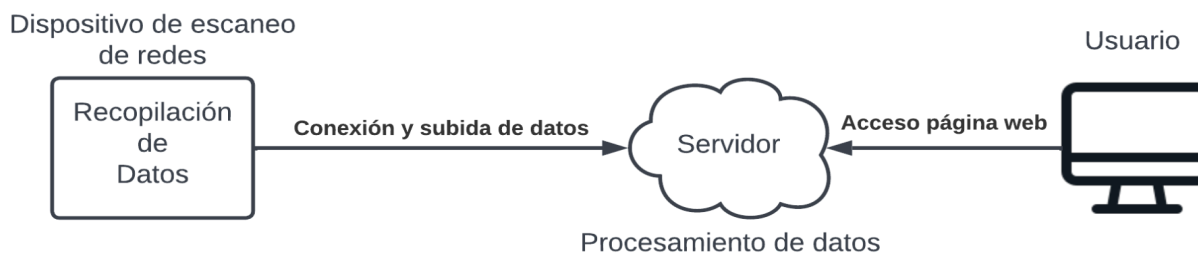


Figura 12. Diseño general del proyecto

Desde el punto de vista de los datos pretratados antes de que el sitio web esté disponible, se considera que los datos recopilados previamente se envían al servidor para su uso posterior en el sitio web bajo petición.

Como puede ver en investigaciones anteriores, se trabajará específicamente con puntos de acceso debido a su disponibilidad y la falta de implementación adicional en el lugar de trabajo. Asimismo, cabe señalar que estos Puntos de Acceso (APs) deberán ser entregados en el marco del plan desarrollado para el proyecto, en este caso, el área de cobertura de las instalaciones de la UPV. El primer módulo tratará, de escanear los AP disponibles para tratar la información que se recibe de ellos, esta recepción de datos y el tratamiento del mismo, incluyendo la conexión con el servidor, la realizará el microprocesador elegido.

Para este desarrollo se ha decidido utilizar la placa ESP32. Se ha escogido el ESP32 ya que es una placa accesible fácilmente en páginas de compra/venta como Amazon, además de ser accesible su precio es muy económico y para realizar el proyecto a gran escala es un dispositivo que podría obtener muy buenos resultados.

El segundo módulo abarca el servidor, donde se contempla la página web de recepción, tratamiento de datos y envío de estos a la siguiente parte del mismo, la base de datos. Se va a utilizar una base de datos mySql ya que Plesk proporciona una herramienta de administrador de base de datos de fácil accesibilidad la cual se trata de phpMyAdmin. Este administrador de base de datos MySQL tiene un uso sencillo de la aplicación web con una interfaz amistosa, además que es un software muy popular desarrollado en PHP.

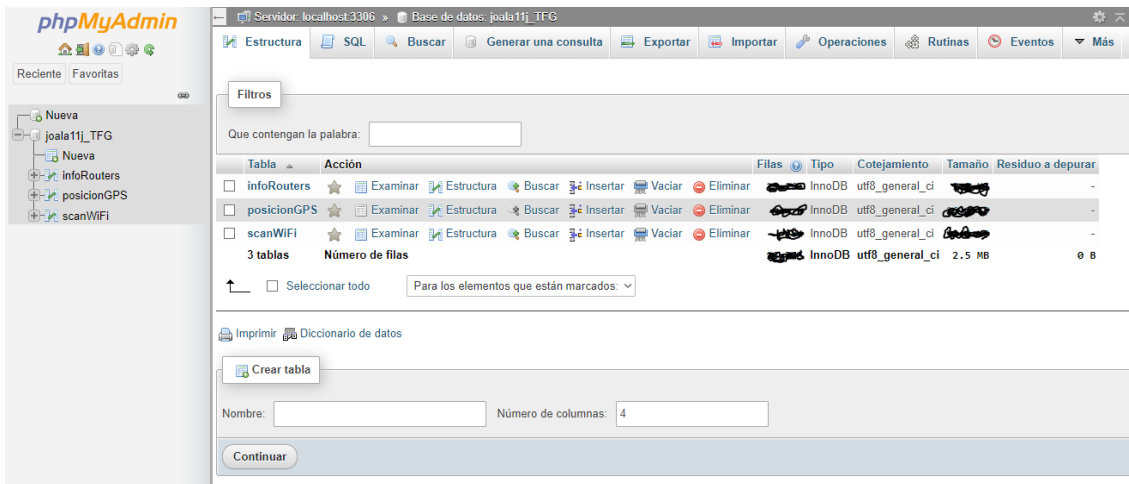


Figura 13. phpMyAdmin

Después de estudiar el algoritmo propuesto y los métodos de posicionamiento disponibles, se decidió utilizar el algoritmo de ancho completo y medio largo, que es un método probabilístico. Esto se debe a que la implementación del algoritmo TOA/TDOA requiere una alta precisión y depende del tiempo del equipo, mientras que el algoritmo AOA requiere más equipo para lograr la precisión y aumenta el costo. Por otro lado, el algoritmo RSSI es fácil de implementar porque los datos están fácilmente disponibles gracias a la investigación en informática. Sin embargo, debido a la alta precisión del resultado final, se decidió utilizar el promedio.

Dada la técnica de la probabilidad y tras el estudio de abundantes pruebas de recopilación de datos de estudios realizados en proyectos externos, se llega a la conclusión de que la utilización de los algoritmos como TOA, AOA y RSSI funcionan correctamente en ambientes donde los AP son escasos a comparación del espacio en el que se desarrolla este mismo proyecto. Por eso y tras realizar pruebas teóricas con datos variados, se ha verificado que en el 90% de resultados da un margen de error menor a un metro de diferencia con la posición real de escaneo.

Antes de realizar las configuraciones adecuadas, se debe seleccionar la opción de calcular la ubicación del dispositivo para recopilar datos del entorno antes de iniciar la aplicación desde el dispositivo con una porción mínima de código escaneado de AP. Por lo tanto, se debe filtrar la lista de puntos de acceso con direcciones MAC. También se debe seleccionar el tipo de datos a almacenar, lo que resulta en la selección de información durante el procesamiento.

En la base de datos se ha importado toda la información de los puntos de acceso que proporcionó la UPV para el procesamiento de datos, esta tabla tiene el nombre de infoRouters y cada registro contiene la información de un AP, estos atributos de la tabla son:

- nombre, el cual hace referencia al SSID del AP.
- espacio, donde se encuentra dentro del campus.
- latitud.
- longitud.
- mac.

Por otro lado se tiene la tabla scanWiFi, en esta tabla se insertarán los AP escaneados por el dispositivo. Cada registro de la tabla contempla los siguientes atributos:

- id, creado automáticamente e incremental para cada inserción en BBDD.
- idDispositivo, que es igual a un nombre definido en el código interno de cada dispositivo.
- bssid, este atributo es un nombre de identificación único de todos los paquetes de una red inalámbrica para identificarlos como parte de esa red. Este se forma con la dirección MAC por tanto esta se podrá averiguar del siguiente atributo.
- intensidad, este atributo hace referencia a la intensidad de señal con la que el dispositivo localizador detecta la señal del AP. No se va a utilizar en la solución final ya que el algoritmo escogido es la media aritmética pero se utilizó a principios del desarrollo para realizar pruebas teóricas sobre este.
- hora, la hora y día en la que insertó el registro dentro de la tabla como un TimeStamp.

Y por último la tabla posicionGPS, donde se inserta el resultado final del cálculo de la localización del dispositivo receptor, que en este caso se trata del ESP32. Esta tabla consta de los siguientes atributos para cada registro:

- id, creado automáticamente e incremental para cada inserción en BBDD.
- dispositivo, que es igual a un nombre definido en el código interno de cada dispositivo.
- latitud.
- longitud.
- horaPosicion, la hora y día en la que insertó el registro dentro de la tabla como un TimeStamp.

Sabiendo qué datos recopilar y cómo, ahora debe considerar cómo transferir la información al servidor, procesándola en el formato correcto. Esto significa que los datos se transmiten directamente en el momento de la intercepción.

Para ello, es mejor haber enviado previamente los datos recopilados (desde el dispositivo) directamente a la nube (servidor) con la necesidad de generar un código que permita conectarse al servidor y descargarlo automáticamente.

Este código que conecte con el servidor sería óptimo que se utilizara una página web con un método GET para pasarle los datos directamente una vez obtenidos. Este es un método de solicitud HTTP que se usa junto con otro método llamado POST. Estos escenarios que usan los datos pasados a la página son de para facilitar la lectura. De esta manera, tanto GET como POST pueden utilizar los mismos tipos de datos. Aunque surgen diversas diferencias según la importancia de cada uno, los desarrolladores web prefieren utilizar este método. Con éste es posible recuperar los datos, devolviéndolos de forma precisa y concreta en un formato. Una de las ventajas del método "get" (obtener en español) es que forma parte de la URL, por lo que es fácil de compartir con los colaboradores.

Esto incluye poner datos en una URL, recopilarlos en el servidor, ejecutar el código que hace los cálculos necesarios y devolver una respuesta en formato de



página web. La respuesta, para mayor conveniencia, será como la salida estándar de una ventana de salida de un debugger, siendo 'OK' cuando los datos se insertan de manera correcta y saltando una excepción si ocurre en alguna parte del código. Es posible realizarlo con el ESP32 y la programación de Arduino ya que se aprovecha una librería con las funcionalidades para realizarlo, esta librería tiene el nombre de 'HTTPClient.h'.

Ahora que se ha explicado cómo se diseñará este proyecto, se muestran los esquemas/diseños que se utilizarán para que funcione correctamente.

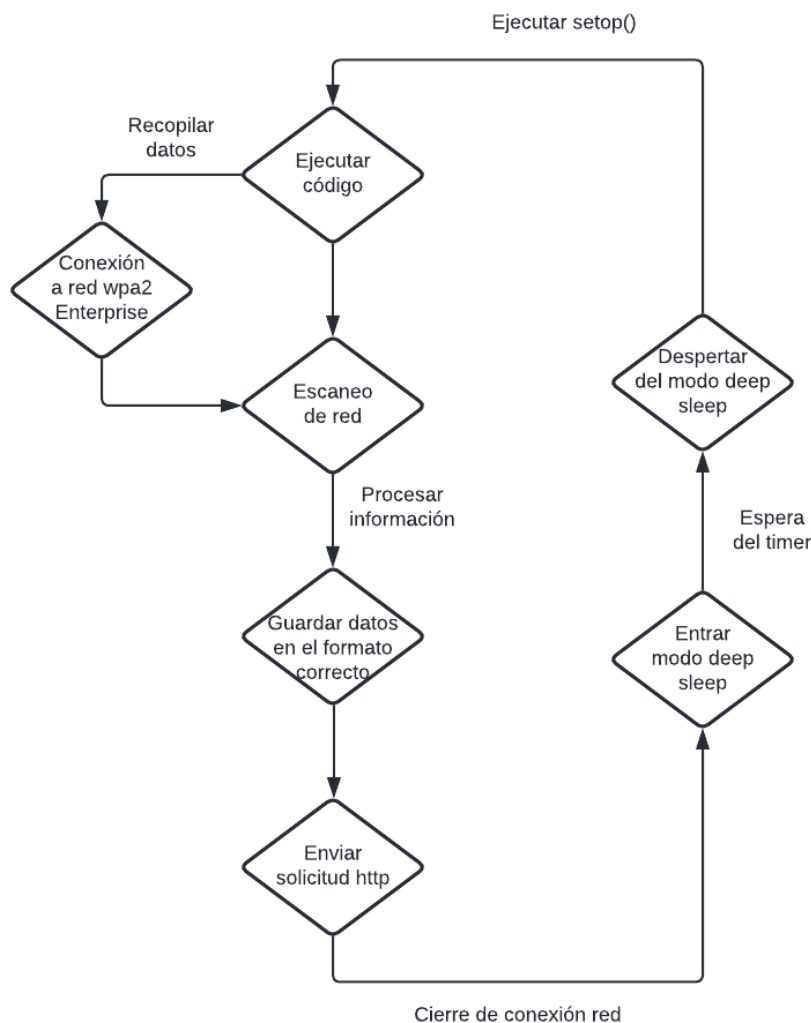


Figura 14. Diagrama de código Arduino a seguir

Una explicación rápida, dada la forma en que funcionará completamente como se mostró anteriormente, es que la operación de recopilación de datos pasará por un escaneo de red con un tiempo de escaneo específico. Luego, estos datos se procesarán para obtener solo la información solicitada y luego se vincularán al sitio web mediante el envío de los datos procesados utilizando el método GET.

Por otro lado tenemos la página web y la parte del servidor.

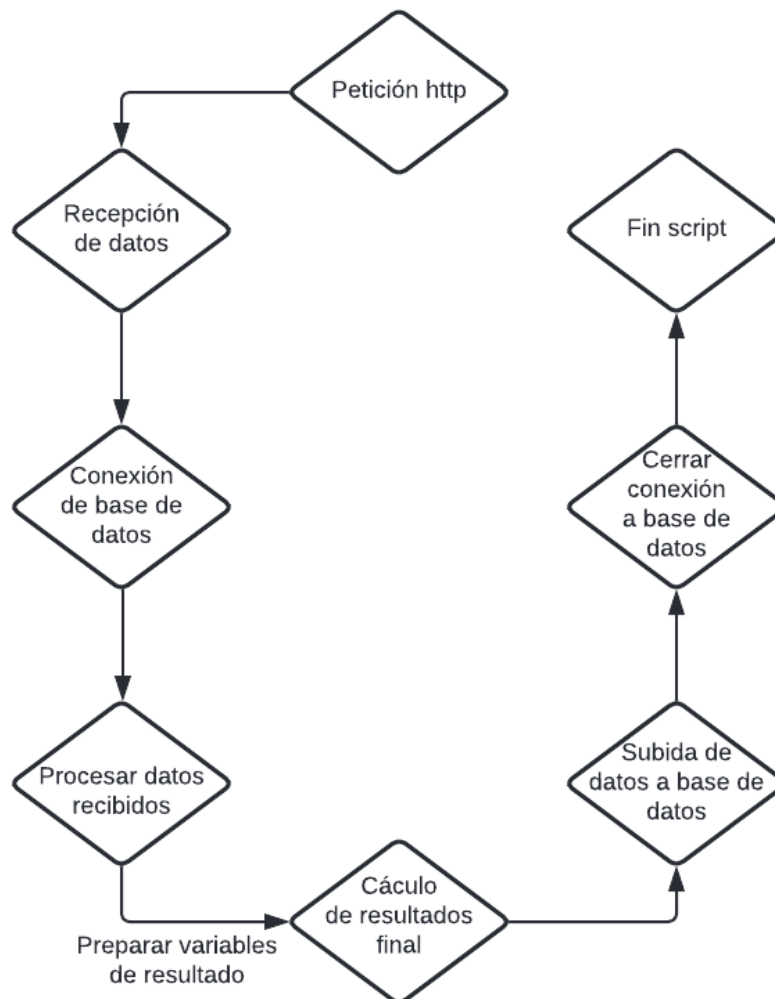


Figura 15. Diagrama del código de la página web a seguir junto al servidor

4.1 Escaneo de los AP

A continuación se presenta el código fuente correspondiente a la placa ESP32, este realiza el escaneo de redes y realiza la conexión con la página web, a continuación se explicará más en detalle cada una de sus partes. El contenido se divide en dos módulos que se detallarán a continuación.

Primeramente, se encuentra la importación de las bibliotecas para el uso de la función de escaneo, la conexión por http y la librería que permite la conexión a la red eduroam, se van a describir con más detalle:

- La biblioteca WiFi se encarga esta biblioteca permite que una placa Arduino se conecte a Internet. Puede servir como un servidor que acepta conexiones

entrantes o como un cliente que realiza conexiones salientes. La biblioteca admite el cifrado PERSONAL WEP y WPA2, pero no WPA2 Enterprise, por esta razón es importante el uso de una biblioteca externa.

- La biblioteca HTTPClient se utiliza para realizar fácilmente solicitudes HTTP GET, POST y PUT a un servidor web. Funciona con cualquier clase derivada de Client, por lo que cambiar entre Ethernet, WiFi y GSMClient requiere cambios mínimos en el código.
- La biblioteca esp_wpa2 nos permite añadir la funcionalidad sobre la placa ESP32 de conectarnos a una red WPA2 Enterprise, por ello se utilizará para conectarse a la red eduroam.

Seguidamente aparece la definición de las variables estáticas que se van a utilizar en el programa, estas variables son las siguientes:

- EAP_IDENTITY "joala11j@alumno.upv.es" → Credencial de usuario para la conexión con la red eduroam.
- EAP_PASSWORD "*****" → Credencial de contraseña para la conexión con la red eduroam.
- uS_TO_S_FACTOR 1000000 → Factor de conversión de microsegundos a segundos.
- TIME_TO_SLEEP 43200 → Tiempo para que el dispositivo despierte del deep sleep, 12 horas en segundos.
- ssid = "eduroam" → SSID de la red Eduroam, nombre por el cual se identifica la red eduroam.
- PAGINAWEB = "<http://joala11j.upv.edu.es/miphp.php?texto=>" → Página web del servidor donde realiza la inserción a base de datos y el tratamiento de estos.
- counter = 0 → Contador de intentos de conexión a la red.

```
#include <WiFi.h> //Wifi library
#include <HTTPClient.h>
#include "esp_wpa2.h" //wpa2 library for connections to Enterprise networks
#define EAP_IDENTITY "joala11j@alumno.upv.es"
#define EAP_PASSWORD "*****" //your Eduroam password
#define uS_TO_S_FACTOR 1000000 //Conversion factor for microseconds to seconds
#define TIME_TO_SLEEP 43200 //12 hours to seconds
const char* ssid = "eduroam"; // Eduroam SSID
static const char *PAGINAWEB = "http://joala11j.upv.edu.es/miphp.php?texto=";
int counter = 0;
```

Figura 16. Importación de las librerías y definición de variables estáticas

En segundo lugar, encontramos la parte del código fuente donde se ejecuta la conexión a la red eduroam, el escaneo de la información de los AP disponibles en el perímetro de la placa y el envío de los datos procesados al servidor.

En el setup del programa se encuentra todo el código ejecutable que realiza las funcionalidades mencionadas, esto se debe a que se utiliza el modo de funcionamiento deep sleep que incluye el ESP32. Este modo hace que el programa dentro del setup se repita cada cierto periodo de tiempo determinado, haciendo que el dispositivo solo trabaje ejecutando esta parte y se ponga en modo sueño profundo. Este modo hace que el dispositivo no malgaste energía, o bien esperando un periodo de tiempo pero en estado activo o ejecutando un programa excesivas veces en el loop, lo que acortaría su vida útil y causaría recargar la batería cada cierto periodo corto de tiempo. Utilizando este modo se puede lograr un gasto mínimo de energía y logrando a su vez que trabaje sin que haya que modificarlo ni desconectarlo de su puesto de trabajo.

Una vez se ha dejado clara la utilización del modo deep sleep, el programa sigue primeramente con la conexión a la red eduroam. Con el uso de las variables estáticas mencionadas anteriormente y utilizando las funciones que nos brinda la biblioteca esp_wpa2.h permite conectarse mediante las siguientes instrucciones a la red:

- esp_wifi_sta_wpa2_ent_set_identity → Establece la identidad para el método PEAP/TTLS. La API sólo pasa el parámetro identidad a la variable de puntero global en el módulo enterprise wpa2.
- esp_wifi_sta_wpa2_ent_set_username → Establece el nombre de usuario para el método PEAP/TTLS. La API sólo pasa el parámetro nombre de usuario a la variable de puntero global en el módulo enterprise wpa2.
- esp_wifi_sta_wpa2_ent_set_password → Establece la contraseña de usuario para el método PEAP/TTLS. La API sólo pasa el parámetro nombre de usuario a la variable de puntero global en el módulo enterprise wpa2.
- esp_wifi_sta_wpa2_ent_enable → Habilita la autenticación de la red enterprise wpa2.
- WiFi.begin → Inicializa la configuración de red de la biblioteca WiFi y proporciona el estado actual.

Seguidamente, una vez realizada la conexión a la red, el programa sigue con el escaneo de los AP disponibles en el perímetro del dispositivo, para realizar esta funcionalidad el programa necesita de la biblioteca WiFi.h. Esta biblioteca permite utilizar la clase Scan de Arduino al utilizar la función scanNetworks. La función nos permite buscar redes Wi-Fi disponibles en una sola ejecución y devuelve la cantidad de redes descubiertas.

Una vez obtenemos estas redes con la función scanNetworks se debe realizar un correcto tratamiento de los datos recibidos. Esto quiere decir que la información debe estar en el correcto formato para proporcionarle a la página web mediante un método GET que recabará los datos. Este tratamiento de datos entra en el bucle del programa que encontramos como un foreach. Dentro de este bucle, para cada elemento encontrado, realizará una concatenación en una cadena de texto el nombre del dispositivo que ha realizado el escaneo, el BSSID del punto de acceso y la intensidad de señal con la que se recibe el escaneo (se utilizó para el estudio previo del proyecto), separando cada atributo por una coma y cada elemento por un '>'.>



Finalmente, una vez se completa de realizar el bucle y se tiene la cadena de texto en el formato correcto se realiza la conexión con el sitio web. Esta se realiza gracias a la biblioteca HTTPClient.h, que permite enviar solicitudes HTTP y recibir respuestas HTTP de un recurso identificado por un URI, en este caso la URI se encuentra definida en las variables estáticas del programa (PAGINAWEB). Los datos a enviar desde la aplicación incluirán todos los puntos de acceso y/o routers que la aplicación detecte por su dirección MAC y su respectiva intensidad de señal, en una etiqueta denominada texto.

Una vez se realiza la petición http se cierra la conexión y se activa el modo deep sleep durante el periodo de tiempo definido (12h), y una vez ese periodo acabe el dispositivo seguirá funcionando en el modo activo y ejecutará de nuevo el setup.

Cabe destacar que todos las líneas del código fuente que realizan una llamada a Serial.print se utilizaron para demostrar y verificar el correcto funcionamiento del programa en modo debug, esto en la release final se eliminaría y/o se sustituirá por un tratamiento de excepciones en caso de haber un error de conexión o al realizar la petición a la página web, aunque eso comprendería más en la verificación del correcto funcionamiento de todos los módulos del servidor.

```

void setup() {
    Serial.begin(115200);
    delay(1000);
    esp_sleep_enable_timer_wakeup(TIME_TO_SLEEP * uS_TO_S_FACTOR);
    //preparamos el timer para el modo deep sleep

    Serial.println(); //Línea disponible solo en debug
    Serial.print("Connecting to network: "); //Línea disponible solo en debug
    Serial.println(ssid); //Línea disponible solo en debug

    WiFi.disconnect(true);
    WiFi.mode(WIFI_STA);
    esp_wifi_sta_wpa2_ent_set_identity((uint8_t *)EAP_IDENTITY,
strlen(EAP_IDENTITY)); //provide identity
    esp_wifi_sta_wpa2_ent_set_username((uint8_t *)EAP_IDENTITY,
strlen(EAP_IDENTITY)); //provide username --> identity and username is same
    esp_wifi_sta_wpa2_ent_set_password((uint8_t *)EAP_PASSWORD,
strlen(EAP_PASSWORD));
    //provide password
    esp_wifi_sta_wpa2_ent_enable();
    WiFi.begin(ssid); //connect to wifi
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print("."); //Línea disponible solo en debug
        counter++;
        if(counter>=60){
            ESP.restart();
        }
    }
    //Conectado al WiFi
    if(WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        String enviar = "";
        int n = WiFi.scanNetworks();
        if (n == 0) {}
        else {

            Serial.print(n);
            //Línea disponible solo en debug
            Serial.println("networks found");
            //Línea disponible solo en debug

            for (int i = 0; i < n; ++i) {
                enviar += "GPS1,";
                enviar += WiFi.BSSIDstr(i);
            }
        }
    }
}

```



```
        enviar += ",";
        enviar += WiFi.RSSI(i);
        enviar += ">";
        delay(10);
    }

    Serial.println(enviar); //Línea disponible solo en debug
}
http.begin(PAGINAWEB + enviar);
int codigo_respuesta = http.GET();
if(codigo_respuesta > 0) {

    Serial.print("El servidor ha respondido --> ");
    //Línea disponible solo en debug
    Serial.println(codigo_respuesta);
    //Línea disponible solo en debug

}
delay(100);
http.end();
}
esp_deep_sleep_start(); //
}
void loop() {}
```

Figura 17. Programa setup para escaneo y envío de datos

4.2 Inserción y cálculo de datos

Los datos se obtendrán y procesarán para incluirlos en la base de datos, una vez hayan llegado a esta página (hello.php), donde se debe alojar tanto los datos recibidos como el resultado del cálculo final obtenido.

A continuación se explica el funcionamiento del código fuente de la página web que recibe la información de la placa una vez el dispositivo lanza la petición http. Esta se encarga de recabar la información mediante un método GET, ya que se utiliza el lenguaje PHP compatible con métodos GET y POST es muy sencillo implementar esta parte del funcionamiento, se puede observar como con una instrucción sencilla dentro del script se guarda en la variable 'texto' (\$texto = \$_GET['texto']).

En primer lugar, una vez se tiene la información en la variable mencionada, se ha de realizar la correcta conexión a la base de datos. Esta conexión a base de datos se realiza mediante la función 'mysqli_connect', la cual se incluye en la extensa librería de métodos que ofrece PHP. La versión utilizada de PHP en esta solución es la 7.4.30, ya que la versión de PHP utilizada respecto a PHP 5 representa algunas ventajas como:

- Los Bytes de almacenamiento variable reducidos para reducir el espacio de memoria y acelerar las operaciones variables.
- La mejora en la estructura de la matriz: los elementos de la matriz y el mapa hash se asignan en la misma memoria, lo que reduce la huella de la memoria y mejora la tasa de aciertos de la memoria caché de la CPU.
- El mecanismo de llamada de función. Al optimizar la conexión de paso de parámetros, se reducen algunas instrucciones y esto causa que se mejore la eficiencia de ejecución.

Siguiendo con el análisis del código, en cuanto a la conexión a base de datos, PHP realiza la conexión sobre el servidor donde se aloja la página pasándole como parámetros el usuario, contraseña y el nombre a la base de datos la cual se debe conectar, esta conexión queda plasmada en la variable 'link'. Seguidamente se verifica la conexión, en caso de que haya fallado la conexión a la base de datos nos mostrará el error concreto que se ha producido.

En segundo lugar, ya que la variable 'texto' contiene los datos enviados por el dispositivo localizador, deberemos añadir estos a la tabla scanWiFi de la BBDD. Este paso se realiza diferenciando cada elemento a insertar en un registro, para cada conjunto del bloque de la cadena de texto separado mediante el carácter '>', se guarda en la variable dinámica correspondiente el valor del conjunto seleccionado. Estas variables son:

- filas: Array donde cada elemento es un registro a insertar, con sus tres elementos, en la tabla de scanWiFi.
- comas: Array que contiene los tres elementos de cada registro a insertar en la tabla scanWiFi.
- nombreDispo: Nombre del dispositivo del registro.
- mac: MAC que pertenece al registro a insertar.
- intensidad: Intensidad de señal con la que se detectó ese AP en el escaneo.
- sql: Inserción sql para la base de datos scanWiFi

Una vez se ha realizado la inserción de los datos en la tabla scanWiFi se guarda mediante una suma la longitud y la latitud correspondiente a ese AP debido a dos consultas sql a la tabla infoRouters. Estas variables a cada pasada del bucle se irán acumulando junto al número de elementos que se ha detectado en el escaneo por medio de la variable elementos. Estas variables son:

- sqlLong: Consulta sql para la obtención de la longitud del registro que se encuentra en el bucle.
- sqlLat: Consulta sql para la obtención de la latitud del registro que se encuentra en el bucle.
- longSuma: Variable acumulativa de las longitudes que se están consultando de la BBDD.
- latSuma: Variable acumulativa de las latitudes que se están consultando de la BBDD.



- posicionLatFinal: Latitud media calculada según la variable latSuma y el número de elementos encontrados.
- posicionLongFinal: Longitud media calculada según la variable longSuma y el número de elementos encontrados.
- sqlPosFinal: Variable donde se realiza la inserción sql de la posición final calculada en la tabla posicionGPS.

Una vez realizada la inserción del resultado de la posición final en la tabla correspondiente se debe cerrar la conexión con mysql, esto es un requisito que viene en la documentación oficial de php.

```
<html>
  <head>
    <title>Acabas de insertar elementos en mi base de datos</title>
  </head>
  <body>
    <label>Inserción de la información en la base de datos del TFG</label>
    <?php
      $texto = $_GET['texto'];
      //Conexión a la BBDD
      $link = mysqli_connect("localhost", "joala11j_joala11j",
"_Y2217dni20916896_", "joala11j_TFG",3306);
      $nombreDispo = "";
      // Check connection
      if($link === false){
        die("ERROR: Could not connect. " . mysqli_connect_error());
      }
      $elementos = 0;
      $longSuma = 0.0;
      $latSuma = 0.0;
      $texto = str_replace(":", "", $texto);
      $filas = explode(">", $texto);
      echo var_dump($filas);
      foreach($filas as $fila){
        $comas = explode(",", $fila);
        $nombreDispo = $comas[0];
        $mac = $comas[1];
        $intensidad = $comas[2];
        $sql = "INSERT INTO scanWiFi (idDispositivo, bssid, intensidad) VALUES
('$nombreDispo', '$bssid', '$intensidad')";
        if($mac != " " && mysqli_query($link, $sql)){
          echo $sql;
          echo "Signal added successfully.";
        } else{
          echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
        }
      }
    </?php>
  </body>
</html>
```

```

$elementos += 1;
$sqlLong = "SELECT longitud FROM infoRouters WHERE mac='$mac'";
$sqlLat = "SELECT longitud FROM infoRouters WHERE mac='$mac'";
$longSuma += (float) mysqli_query($link, $sqlLong);
$latSuma += (float) mysqli_query($link, $sqlLat);
}

$posicionLongFinal = $longSuma / $elementos;
$posicionLatFinal = $latSuma / $elementos;
$sqlPosFinal = "INSERT INTO posicionGPS (dispositivo, latitud, longitud)
VALUES ('$nombreDispo', '$posicionLatFinal', '$posicionLongFinal')";
if(mysqli_query($link, $sqlPosFinal)){
    echo $sqlPosFinal;
    echo "Signal added successfully.";
} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
}

// Close connection
mysqli_close($link);
?>
</body>
</html>

```

Figura 18. Código fuente del sitio web con método GET

5. Resultados obtenidos

Llegado a este punto, es preciso hacer mención y exponer los resultados obtenidos según las tecnologías y el entorno en el que ha sido realizado este proyecto, además de enunciar las propuestas de posibles soluciones manteniendo el mismo objetivo y con el mismo fin.

Una vez se ha realizado la explicación del código fuente que pertenece al dispositivo localizado ESP32 y el perteneciente a la página web donde se controla la inserción a base de datos se ha desarrollado una solución gráfica sencilla de como mostrar los datos en una tabla, pudiendo ver el recorrido que ha realizado un dispositivo concreto en una página web utilizando html y php.

Se debe tener en cuenta que el objetivo del proyecto es llevar a cabo un dispositivo de seguimiento de obras de arte para el campus de la UPV y su consiguiente explotación.

Teniendo en cuenta las ventajas del dispositivo de seguimiento para obras de arte, se ha optimizado para garantizar un rendimiento a largo plazo con una impresión precisa. Estos datos se manejan en una base de datos alojada en el servidor donde se ha creado una solución en la que se muestra en forma de tablas la información de las posiciones donde ha estado el dispositivo y la fecha y hora exactas en las que realizó el cálculo, está realizado de esta forma ya que el tiempo transcurrido desde que el dispositivo realiza el cálculo hasta que se insertan los datos en la tabla correspondiente es mínimo y no ha de haber diferencia en la posición en que se encuentra.

Asimismo, considerando que se han utilizado los puntos de acceso (AP) que son unos dispositivos que conectados a un Router forman una red inalámbrica (WLAN) en un área específica; el módulo ESP32 que es una solución integrada Wi-Fi/Bluetooth todo en uno, que proporciona no solo comunicación de radio inalámbrica, sino también un procesador integrado con interfaces para conectarse a diferentes dispositivos periféricos. Igualmente, se ha empleado la programación Arduino, que consiste en traducir en líneas de código las tareas automatizadas que queremos realizar leyendo los sensores.

Es por todo ello, que considerando que este sistema tiene que ser útil para localizar obras de arte dentro del territorio de la UPV, se pueden analizar los siguientes resultados.

Se presenta el código fuente de una página web creada con el objetivo de mostrar los resultados obtenidos del sistema de localización mediante tablas según el número de dispositivos que tengan un resultado en la tabla de la base de datos con nombre posición GPS.

El código fuente se describe en dos partes, primeramente encontramos la conexión a la base de datos mysql, realizada igualmente con PHP como se ha explicado en el capítulo anterior. Una vez realizada la conexión a base de datos se realiza una consulta sql con el objetivo de conocer la cantidad de dispositivos de los

que se debe realizar una tabla con la información pertinente de la base de datos. Entonces, para cada dispositivo que se han encontrado se realiza otra consulta para conocer la información total de este según la tabla posición GPS.

Una vez se ha ejecutado la consulta y se obtiene el resultado, se realiza una tabla html posicionando la información obtenida de forma clara y que se entiende a la perfección, ubicando de forma ascendente los registros con hora más actual.

El código fuente de la página propuesta como solución se presenta a continuación.

```
<html>
<head>
<title>Página tabla posiciones</title>
</head>
<body>
<?php

    $link = mysqli_connect("localhost", "joala11j_joala11j", "_Y2217dni20916896_",
"joala11j_TFG",3306);
    if($link === false){
        die("ERROR: Could not connect. " . mysqli_connect_error());
    }

    $sqlNumeroDispositivos = "SELECT DISTINCT(idDispositivo) FROM
`posicionGPS`";
    $dispositivos = mysqli_query($link , $sqlNumeroDispositivos);

    while ($fila = mysqli_fetch_array($dispositivos)) {
        $sqlRegistrosDispositivo = "SELECT * FROM `posicionGPS` WHERE
dispositivo = ".$fila['dispositivo']." ORDER BY horaPosicion";
        $registros = mysqli_query($link , $sqlRegistrosDispositivo);
        printf("<p></p>");
        printf("<h2>Localizaciones del dispositivo %s</h2>", $fila['dispositivo']);

printf("<p>-----</p>");
        printf("<p></p>");
        printf("<table border>");
        printf("<tr ALIGN=center>");
        printf("<td>Hora</td>");
        printf("<td>Latitud</td>");
        printf("<td>Longitud</td>");
        printf("<tr>");
        while($filasRegistro = mysqli_fetch_array($registros)){
            printf("<tr ALIGN=center><td>%s</td><td>%s</td><td>%s</td></tr>",
$filasRegistro["horaPosicion"],$filasRegistro["latitud"],$filasRegistro["longitud"]);
        }
        printf("<p></p>");
    }
}
```



```
}
mysql_close($link);
?>
</body>
</html>
```

Figura 19. Código fuente página web con tablas de posición

El resultado de acceder a este sitio web es el mostrado en la Figura #, donde se puede observar como para dos dispositivos distintos se muestran los resultados del cálculo de su posición.

Localizaciones del dispositivo Monoliso

Hora	Latitud	Longitud
2022-07-26 18:24:35	39.481937163	-0.344762848
2022-07-26 18:14:09	39.481901207	-0.344715259
2022-07-26 18:04:34	39.482030026	-0.345170445

Localizaciones del dispositivo El grito"

Hora	Latitud	Longitud
2022-07-25 16:27:05	39.483075036	-0.347648893

Figura 20. Representación gráfica de las tablas con el recorrido de un dispositivo

La veracidad de estos resultados obtenidos es muy precisa en comparación con la localización tomada en la vida real, siendo la diferencia como máximo de 3 metros en cuanto a la calculada y la real.

En la figura 21 se observa la diferencia de precisión con la posición calculada y la real (Punto 1), donde el margen es de exactamente 3 metros. En este ejemplo se utilizó un dispositivo averiado, ya que tras analizar los resultados del escaneo presencié que había escasa información en cuanto a otros escaneos que realicé al principio del proyecto, debido a esto el margen de error se ha tomado como 3 metros.

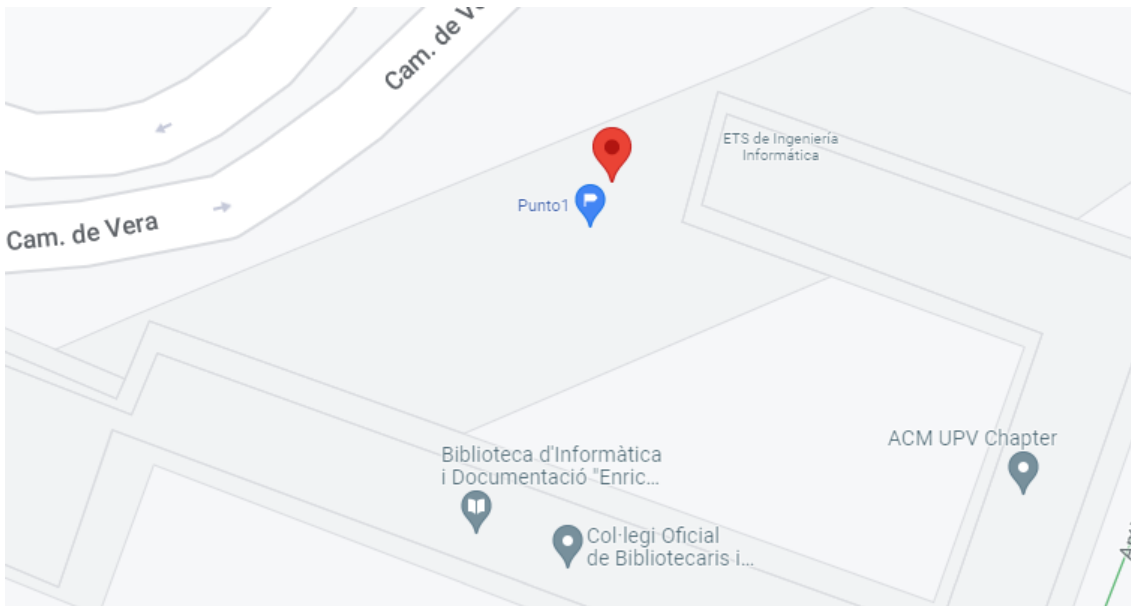


Figura 21. Comparación en Google Maps de la localización real tomada y la calculada

5.1 Propuestas de mejora

Se van a proponer diferentes soluciones para la utilización de las tecnologías mencionadas y utilizadas en este proyecto.

En primer lugar, como ya se ha mencionado anteriormente, una posibilidad sería diseñar una aplicación web para mostrar gráficamente la ubicación de varios dispositivos en el campus e incluso mostrar una ruta tomada durante un período de tiempo. Esta propuesta se podría realizar mediante el uso del API de Google Maps, ya que es muy sencillo de implementar y tiene herramientas para definir rutas, colocar marcadores y mucho más.

También se podría realizar la propuesta anterior utilizando un plano de la infraestructura de la UPV en formato svg, pudiendo implementar mediante una biblioteca pública herramientas como colocar marcadores o rutas interactivas con varias pestañas a su favor. Se muestra a continuación un ejemplo en la figura de abajo.



Figura 22. Ejemplo de solución con plano svg

Otra posible solución podría ser el crear una aplicación web con el objetivo de generar informes sobre los dispositivos de seguimiento, utilizando la información conocida de los AP se podría localizar en qué zona del campus está, el tiempo que ha tardado en cambiar de posición, qué recorrido ha realizado en el cambio de posición y una descripción de donde se encuentra la obra. Para eso último, según a qué solución se requiera, cabe la posibilidad de que se deba cambiar en la estructura de la base de datos.

Esta última solución propuesta plantea también la posibilidad de una interacción con la información que se dispone en base de datos, pudiendo si fuera el caso, añadir nuevas infraestructuras y, a su vez, la información de estas sobre los puntos de acceso

disponibles. Aunque si no se utiliza un API implementada de un mapa geográfico se debería añadir más planos svg por el estilo.



6. Conclusión

En general, ha sido interesante el proyecto a la hora de desarrollarlo.

Existen 3 algoritmos que se han podido probar usando diferentes métodos de localización. El uso de la tecnología Wi-Fi solo está relacionada con su uso diario, por lo que en otras situaciones puedes incluso utilizar dispositivos como Raspberry. El procesamiento de datos incluso se ha desarrollado utilizando MySQL o alguna otra base de datos.

Entre las las ventajas que se han podido detectar, es necesario destacar que el resultado ha sido favorable y además, la implantación del dispositivo desarrollado en instalaciones universitarias sería factible, ya que se trata de un pequeño chip que se podría colocar en obras de arte (cuadros, esculturas, grabados, artes decorativas y otras) siendo así imperceptible a simple vista por el ojo humano.

Asimismo, se ha comprobado experimentalmente que la diferencia máxima entre la posición real del entorno y la posición medida es de aproximadamente 3 metros, utilizando el algoritmo de la media aritmética. Teniendo en cuenta que, aún así, si se hubiera aplicado alguno de los otros algoritmos posibles la precisión obtenida hubiera sido equivalente e incluso superior.

Se debería destacar de forma esquemática las siguientes ventajas:

- El uso del lenguaje PHP 7, mucho más veloz que sus anteriores versiones, su mejor rendimiento y como consiguiente mejores resultados se obtienen.
- El uso de algoritmos y métodos de localización desarrollados en este proyecto es uno de los más fáciles de implementar.
- El uso del IDE de Arduino es fácil, conlleva un bajo coste, es una plataforma abierta y además tiene una alta flexibilidad.
- El almacén de posibilidades que se encuentran disponibles para crear soluciones únicamente con los datos finales de la base de datos.

Si seguimos haciendo mención de los aspectos positivos de la implementación de este dispositivo, es necesario mencionar, como ya se ha hecho anteriormente, que este tiene un precio asequible, con lo cual también lo es el desarrollo del proyecto en sí. Es por ello, que si este chip se tuviera que poner en práctica a gran escala no supondría un coste excesivo.

Sin embargo, el sistema que se ha aplicado funciona bien y debería ser eficaz siempre que trabaje dentro de la infraestructura en la que existe la red eduroam y se conoce su ubicación y puntos de acceso. Se puede decir que se han cumplido los objetivos, aunque algunos de ellos son mejorables. La arquitectura de recopilación de datos subyacente es clara y concisa. Subir datos al servidor puede no ser lo más adecuado, pero esto no debería afectar el desarrollo general del proyecto. La informática del lado del servidor permite usarlos como quieras cuando están en la base de datos, desde aplicaciones web hasta aplicaciones de escritorio.

Finalmente, cabe señalar que todos aquellos conocimientos adquiridos en la universidad se han convertido en una valiosa ayuda para el desarrollo de aplicaciones, porque cuando uno se familiariza sobre algún aspecto, todo funciona mejor, y esta es la combinación de herramientas como MySQL, conocimiento del servidor y conexiones http entre otras.



7. Bibliografía

- [1] *Documentation*. (s/f). Php.Net. Recuperado el 5 de septiembre de 2022, de <https://www.php.net/docs.php>
- [2] *Documentation and help portal for plesk obsidian*. (s/f). Plesk.com. Recuperado el 5 de septiembre de 2022, de <https://docs.plesk.com/es-ES/obsidian/>
- [3] (S/f). Pbworks.com. Recuperado el 5 de septiembre de 2022, de <https://arduinoobot.pbworks.com/f/Manual+Programacion+Arduino.pdf>
- [4] Gómez, P. (2014, julio 22). *Localización en Interiores, Artículo en Bit*. El Blog de Pablo Gómez. <https://pablogomezoviedo.wordpress.com/2014/07/22/localizacion-interiores-articulo-bit/>
- [5] David Sánchez, Sergio Afonso, Elsa M. Macías, Member IAENG y Álvaro Suárez. “*Devices Location in 802.11 Infrastructure Networks using Triangulation*”, en IMECS 2006.
- [6] Eiman Elnahrawy, Xiaoyan Li y Richard P. Martin. “*The Limits of Localization Using Signal Strength: A Comparative Study*”, en Sensor and Ad Hoc Communications and Networks, 2004.
- [7] G. Mao, B. Fidan, B. Anderson. “*Wireless sensor network localization techniques*”, en Computer networks, vol. 51, no. 10, pp. 2529-2553, November 2007.
- [8] Received Signal Strength Indicator and Its Analysis in a Typical WLAN System. (2013).
- [9] Carcavilla Sanz, A. (2006). *Sistemas de posicionamiento basados en WiFi*. Universitat Politècnica de Catalunya.
- [10] Lee, J., Yoon, C., Park, H., & So, J. (2013). *Analysis of Location Estimation Algorithms for Wifi Fingerprint-based Indoor Localization*.

8. Anexo



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



8.1 OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.				X
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.			X	
ODS 8. Trabajo decente y crecimiento económico.				X
ODS 9. Industria, innovación e infraestructuras.		X		
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles.				X
ODS 12. Producción y consumo responsables.				X
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.				X



Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

Los Objetivos de Desarrollo Sostenible (ODS) son un conjunto de 17 objetivos identificados por las Naciones Unidas en 2015 cuando adoptó la Agenda 2030.

El objetivo con el que más se relaciona este trabajo de grado es el número 9, que refleja una industrialización inclusiva y sostenible. Esto se debe a que las uniones económicas son posibles y enfrentan una amplia gama de objetivos entre diferentes personas, con o sin membresía. Se supervisa la industria de la inclusión.

El objetivo es permitir que todos participen en este proyecto. Esto se usará para crear un modelo similar para este proyecto al mejorarlo personalmente para su caso y viceversa. La funda está perfecta y no hay que cambiar nada.

El objetivo principal de esto es que cada persona aporte su conocimiento para crear cada vez un proyecto de mayor nivel hasta que se convierta en un producto que se pueda producir en cualquier entorno deseado.

Por otra parte, los países de ingresos medianos pueden beneficiarse al ingresar a las industrias de metales básicos y de fabricación, que ofrecen una gama de productos que enfrentan una demanda internacional en rápido crecimiento. Esto es destacable por el hecho de que para implantar un sistema así se requiere mucha fabricación de componentes electrónicos y favorece el crecimiento económico del sector industrial.

Un sistema bien desarrollado también es importante, ya que permite la mejora de la investigación científica y la mejora del potencial tecnológico de las industrias de todos los países, especialmente los países en desarrollo, fomentando la innovación y aumentando significativamente el número de personas para 2030. Personal de I+D por millón de habitantes; y el gasto público y privado en I+D.

Otro objetivo con el que se relaciona este trabajo es el número 7, ya que el dispositivo que realiza el escaneo de los AP, en todo tipo de entornos, va a necesitar de un sustento energético para funcionar, ya bien sea conectado directamente a la corriente eléctrica o mediante una batería. Por esta razón, es importante proponer un sustento de energía renovable o sostenible, ya que, aunque el dispositivo funciona en periodos cortos de tiempo y no realiza un gasto eléctrico grande, es positivo que se promueva este tipo de cambios en el entorno comercial en que se instalarán.

En respuesta a la crisis climática que afecta a nuestro planeta, es vital que trabajemos juntos para promover las energías renovables. Reemplazar los combustibles fósiles con energía solar, eólica e hidroeléctrica puede ayudarnos a revertir el calentamiento global y sus terribles consecuencias. La energía renovable no es tan sólo del futuro, sino también del presente. Beneficia al medio ambiente, a la sociedad y a la economía. Una gran ventaja de este tipo de energía es que hace que, como nación, ganemos en independencia energética. España tiene mucho sol así como un gran potencial para la energía eólica y mareomotriz. Si nos enfocamos en promover las fuentes de energía renovables, se puede lograr tener una fuente inagotable de energía. Por esta razón, si el dispositivo desarrollado en el trabajo se instalara en industrias cercanas a estas fuentes de energía, es posible que las grandes

industrias se conciencien y empiecen a fabricar dispositivos que únicamente funcionaran con este tipo de energía en masa y los beneficios económicos serían exponenciales.

En conclusión, todo esto significa que las empresas pueden decidir poner en producción los proyectos creados, y así conseguir puestos de trabajo que garanticen el crecimiento del sector manufacturero, dando a los países en desarrollo la posibilidad de acelerar el desarrollo de este sector, incluso en el sector energético, que puede impulsar las energías renovables.

