

ROBOMAP: Autonomía, Sistema de alertas y Postproceso de datos.

TITULACIÓN: Grado en Ingeniería Informática

AUTOR: Fabián Alfonso Beirutti Pérez

TUTORIZADO POR:

Alexis Quesada Arencibia

Gabriele Salvatore de Blasio

Junio 2023

Agradecimientos

A mi padre y madre, que sacrificaron su carrera profesional y vida social en busca de una mejor vida para sus hijos. Por el apoyo diario incondicional en los buenos y malos tiempos y, aguantar conmigo durante toda esta etapa.

Sin ninguna duda, a mi hermana, que aunque peleemos casi sin parar, resulta ser un pilar fundamental en mi vida, que aporta una chispa de alegría y locura día tras día. Y cómo no, porque siempre que lo necesitaba, tú estabas ahí.

A mis tutores Alexis y Gabriele, por su comprensión y apoyo a lo largo de este trabajo. También, a José Carlos e Iru, compañeros del IUCES que me ayudaron a entender y poner en marcha a ROBOMAP.

Resumen

Este proyecto se ubica dentro del ámbito de los Sistemas de Posicionamiento en Interiores o IPS (del inglés *Indoors Positioning System*) mediante tecnologías Bluetooth y Wi-Fi. Dado un entorno de interior en el que se despliega una serie de balizas Bluetooth Low Energy (BLE) o, puntos de acceso Wi-Fi, es necesario un procedimiento que haga un barrido exhaustivo en el que se recopile información de la intensidad de señal recibida o RSSI (del inglés *Received Signal Strength Indicator*) de cada una de las balizas. Dicha recopilación resulta crítica para el calibrado de un sistema de localización en ambientes interiores. Tras un análisis se revela muy conveniente desarrollar una solución basada en la robótica para automatizar dicho proceso: con este propósito el grupo de investigación CIPERBIG (Computación Inteligente, Percepción y Big Data) del IUCES (Instituto Universitario de Cibernética, Empresa y Sociedad) ha desarrollado la plataforma ROBOMAP.

Este Trabajo de Fin de Grado (TFG) pretende dar continuidad al trabajo desarrollado dotando a ROBOMAP de nuevas características y funcionalidades: autonomía energética, sistema de control de errores y alertas durante la recopilación de datos y, nuevos módulos de postproceso de datos recolectados por la plataforma...

Abstract

This project falls within the scope of Indoor Positioning Systems (IPS) using Bluetooth and Wi-Fi technologies. Given an indoor environment in which a series of Bluetooth Low Energy (BLE) beacons or Wi-Fi access points are deployed, a procedure is needed to perform an exhaustive sweep in which information on the Received Signal Strength Indicator (RSSI) of each beacon is collected. This data is critical for the calibration of an indoor localization system. After an analysis, it is very convenient to develop a robotics-based solution to automate this process: for this purpose, the CIPERBIG research group from the IUCES developed the ROBOMAP platform.

This Final Degree Project aims to give continuity to the work developed by providing ROBOMAP with new features and functionalities: energy autonomy, error and alert control system during the data collection and new modules for post-processing of the data collected by the platform...

Índice general

Índice de figuras	VI
Índice de tablas	VIII
1. Introducción.....	1
1.1 Motivación personal.....	1
2. Competencias y objetivos	3
2.1 Competencias específicas	3
2.2 Objetivos.....	4
3. Estado del arte.....	5
3.1 Posicionamiento en Interiores	5
3.1.1 Exactitud y precisión	5
3.1.2 Tecnologías.....	6
3.1.3 Métodos	6
3.1.3.1 Proximidad.....	6
3.1.3.2 Triangulación y Trilateración.....	7
3.1.3.3 Fingerprinting.....	8
3.2 ROBOMAP	9
3.2.1 Módulo ROBOMAP (MR)	10
3.2.2 Módulo Meteorológico (MM).....	11
3.2.3 Módulo de Adquisición de Datos BLE (MAD-BLE)	12
3.2.4 Módulo Orientador del Receptor BLE (MOR-BLE)	13
3.2.5 Módulo Controlador (MC).....	14
3.2.6 Evolución.....	15
4. Desarrollo	17
4.1 Sistema Energético (SE)	17
4.1.1 Repaso conceptos electricidad	17
4.1.2 Aparatos de medición	19
4.1.3 Estudio ROBOMAP	19
4.1.4 Consumo	22
4.1.5 Desviación.....	25
4.1.6 Propuesta autonomía.....	26
4.2 Sistema de alertas.....	29
4.2.1 <i>Service_Alerts</i>	29
4.2.1.1 CheckWeatherDataJson	30

4.2.1.2 CheckNumberPacketsBLE	31
4.2.1.3 CheckIfAllMacsDetected.....	31
4.2.2 Servidor_MA.....	32
4.2.3 Interfaz	33
4.3 Postproceso de datos	33
4.3.1 Datos empleados	34
4.3.2 WKNN.....	35
4.3.3 Propuesta método de posicionamiento	36
5. Conclusiones y trabajo futuro	40
5.1 Conclusión personal.....	40
5.2 Trabajo futuro	41
6. Referencias	42
Anexo 1. Manual ROBOMAP	47

Índice de figuras

Figura 1. Ejemplo de exactitud y precisión.	6
Figura 2. Triangulación.	7
Figura 3. Trilateración.	7
Figura 4. Fase de calibrado.	8
Figura 5. Fase de posicionamiento.	8
Figura 6. ROBOMAP.	9
Figura 7. Ejemplo diseño entorno de interiores y ruta a seguir en Blender.	10
Figura 8. Componentes movimiento omnidireccional ROBOMAP.	11
Figura 9. Balizas ultrasonido de ROBOMAP.	11
Figura 10. Estación meteorológica de ROBOMAP.	12
Figura 11. Dongle receptor de señales Bluetooth.	12
Figura 12. Plataforma del módulo orientador.	13
Figura 13. Componentes módulo orientador.	13
Figura 14. Instrucciones principales del archivo controlador.py.	14
Figura 15. Métodos auxiliares para inicializar variables y servicios de una campaña.	14
Figura 16. Método que describe el proceso de una captura de datos en una posición.	15
Figura 17. Evolución de ROBOMAP.	16
Figura 18. Tipos de corriente eléctrica.	18
Figura 19. Aparatos de medición.	19
Figura 20. Esquema alimentación principal.	20
Figura 21. Fuentes de alimentación ROBOMAP.	20
Figura 22. Esquema fuente de alimentación (1).	21
Figura 23. Esquema fuente de alimentación (2).	21
Figura 24. Esquema fuente de alimentación (3).	21
Figura 25. Cable manguera y sus partes.	22
Figura 26. Cable fase positivo.	23
Figura 27. Esquema propuesta sistema energético.	28
Figura 28. Montaje baterías sin soldadura de VRUZEND.	28
Figura 29. Código relacionado con el test CheckWeatherDataJson.	30
Figura 30. Código relacionado con el test CheckNumberPacketsBLE.	31
Figura 31. Código relacionado con el test CheckIfAllMacsDetected.	32
Figura 32. Código relacionado con el almacenamiento de las alertas.	32

Figura 33. Interfaz gráfica del sistema de alertas, ROBOMAP Inspector.	33
Figura 34. Mapas zona de prueba con puntos marcados.....	34
Figura 35. Expresión para obtener la RSSI de cada baliza en un punto.	35
Figura 36. Expresión distancia euclídea y peso.....	35
Figura 37. Gráficas resultados WKNN.....	36
Figura 38. Ejemplo de hiperplano óptimo.	37
Figura 39. Resultados métodos de posicionamiento.....	38
Figura 40. Resultados con la zona del 60-100% aumentada.	39

Índice de tablas

Tabla 1. Tipos de potencia en corriente alterna.....	18
Tabla 2. Resultados mediciones en modo de espera (stand by).	23
Tabla 3. Resultado mediciones puesta en marcha del MOR-BLE.	24
Tabla 4. Resultados mediciones durante la puesta en marcha del MR.....	24
Tabla 5. Resultados conversiones de corriente alterna a corriente continua.....	24
Tabla 6. Resultados desviación potencias obtenidas por diferentes métodos.	26
Tabla 7. Niveles de una alerta.....	29
Tabla 8. Resultados métodos de posicionamiento.	38

1. Introducción

La innovación tecnológica en los últimos años ha permitido la mejora y ampliación de las funcionalidades de los *smartphones*. Estos teléfonos inteligentes, resultan casi indispensables en las tareas cotidianas de hoy en día del ser humano como son la comunicación y la localización, siendo este último aspecto, la idea principal detrás de este trabajo. Una de las tecnologías que se ha visto impulsada por las mejoras en los dispositivos móviles en los últimos años son los Sistemas de Posicionamiento en Interiores o IPS (en inglés *Indoors Positioning System*) [1] [2] [3].

Los Sistemas de Posicionamiento Global o GPS (en inglés *Global Positioning System*) fueron y son, tecnologías revolucionarias que han cambiado por completo el cómo funciona el mundo en la actualidad. El GPS consiste en una constelación de satélites que transmiten señales a la superficie de la Tierra, las cuales, proporcionan a los usuarios información sobre posicionamiento, navegación y cronometría. De esta manera, con un receptor de dichas señales, se puede estimar la posición y la hora de manera precisa [4] [5].

Tal y como se describe en la web de Información oficial del Gobierno de los Estados Unidos relativa al Sistema de Posicionamiento Global y temas afines:

“Los resultados extraordinarios obtenidos por el GPS a lo largo de muchos años le han ganado la confianza de millones de usuarios civiles en todo el mundo. Ya probada su fiabilidad en el pasado, el GPS promete ser de gran beneficio a los usuarios en el nivel mundial por largo tiempo” [5].

Y es que, no solo se ha utilizado esta tecnología para facilitar la realización de actividades cotidianas, como conducir, sino que también, se ha usado como una herramienta imprescindible para la prestación de ayuda y servicios, para una mejor comprensión de los fenómenos naturales terrestres, entre otros [6].

En los últimos años, con el crecimiento y desarrollo continuo de las ciudades, pasamos más tiempo en entornos cerrados como centros comerciales o aeropuertos. Surge entonces la necesidad de sistemas de navegación precisos y fiables en este tipo de espacios. A pesar de todos los beneficios que ofrece el GPS, la recepción de señales se ve debilitada, pudiendo incluso a llegar a ser inexistente en entornos de interiores. Es aquí, donde surge la utilidad de los IPS.

1.1 Motivación personal

Durante mis años de estudio del Grado de Ingeniería Informática, he dedicado incontables horas a investigar y entender el funcionamiento y la arquitectura de los ordenadores, sistemas operativos, sistemas inteligentes... Además, he tenido la oportunidad de compaginar los estudios con el trabajo, llegando a formar parte de los equipos de desarrolladores, en dos empresas diferentes, como programador web.

Los IPS son un área de conocimiento relativamente nueva y, por tanto, poco común, interesante y en auge. Así pues, he querido aprovechar el Trabajo de Fin de Grado (TFG) para poder trabajar en un proyecto de ingeniería que no solo implica el desarrollo de código, sino más bien, en uno que, dado un problema, se presenta una solución que consiste en la elaboración de un nuevo producto que combina *hardware* y *software*.

2. Competencias y objetivos

El trabajo realizado cubre las competencias generales, básicas y necesarias para la obtención del título de Grado en Ingeniería Informática [7]. Sin embargo, se mencionan a continuación, las más presentes durante el desarrollo del proyecto:

Código FB04. “Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación de la ingeniería.”

Código FB05. “Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de la programación, y su aplicación para la resolución de problemas propios de la ingeniería.”

Código CII06. “Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.”

Para realizar el análisis previo sobre el estado actual del proyecto y después, la implementación de nuevas funcionalidades, saber cómo interactuar y conectar los sistemas informáticos fue un aspecto clave, dado que todas las partes que funcionan en sintonía para dar vida a ROBOMAP se comunican a través del protocolo UDP y, se programan en diferentes sistemas operativos y entornos. Además, tener en cuenta la idoneidad y complejidad de los algoritmos empleados fue sin duda imprescindible ya que, todo el proyecto se basa en la obtención de señales y estimación de la posición de un objeto u persona en el menor tiempo posible; por lo que, un algoritmo no debe ni consumir demasiada memoria ni tardar en exceso en realizar las operaciones que corresponden.

Gracias a los conocimientos y habilidades adquiridas durante los estudios en la carrera y en la realización de este trabajo, se ha conseguido desarrollar, de la mejor manera posible, las partes que conforman este TFG.

2.1 Competencias específicas

La competencia específica relacionada con este trabajo es la siguiente:

Código CP06. “Capacidad para desarrollar y evaluar sistemas interactivos y de presentación de información compleja y su aplicación a la resolución de problemas de diseño de interacción persona computadora.”

Este TFG consistió principalmente en el análisis en profundidad de las distintas partes que conforman el proyecto de investigación sobre sistemas de posicionamiento en interiores del Instituto Universitario de Cibernética, Empresa y Sociedad (IUCES). Seguidamente, se estudió el consumo energético y desarrolló nuevas funcionalidades para la plataforma robótica ROBOMAP. Luego, se añadieron métodos alternativos para el posicionamiento.

2.2 Objetivos

El objetivo general de este trabajo es contribuir en el proyecto de investigación de Sistemas de Posicionamiento en Interiores del IUCES, se realizó un estudio sobre la arquitectura y funcionamiento de ROBOMAP. De esta manera, se consiguió tener los conocimientos necesarios para poder trabajar sin ningún impedimento las diferentes partes de este proyecto.

Tal y como se describió en la propuesta de este TFG (documento TFT01) [8], los objetivos específicos son:

- Estudio de un Sistema Energético (SE) que dote de autonomía a la plataforma robótica.
- Análisis y desarrollo de un Módulo de Control de Errores (MCE) así como de un Módulo de Alertas (MA) que informe al usuario de la plataforma robótica sobre su estado.
- Análisis del Módulo de Postproceso de Datos (MPD) e implementación de un algoritmo de estimación de posicionamiento.

3. Estado del arte

Los Sistemas de Posicionamiento en Interiores se encuentran en constante evolución y desarrollo, no solo por la creciente demanda de un producto que satisfaga la necesidad de un sistema de navegación preciso en espacios cerrados, sino también, por la continua innovación tecnológica, sobre todo, en los dispositivos inteligentes o *smartphones* [9].

3.1 Posicionamiento en Interiores

“Una localización o posición, es un lugar en un espacio tridimensional (3D) o, sobre una superficie bidimensional (2D). Una posición, puede ser físico o simbólica.” [10].

La finalidad de un IPS, dado un entorno de interiores, es estimar la posición y/o ubicación de una persona o dispositivo, incluso llegando a determinar la planta o altura en la que se encuentra, de manera precisa y fiable. Para conseguirlo se usan balizas (en inglés *beacons*) [11], que emiten señales de diversos tipos como Wi-Fi, Zigbee, ultrasonido, Bluetooth, entre otros.

3.1.1 Exactitud y precisión

La exactitud y precisión, términos que aunque coloquialmente son sinónimos, en este contexto son dos conceptos diferentes e importantes que miden el desempeño de un IPS.

La exactitud indica la diferencia entre un valor estimado y el real, mientras que, la precisión indica la dispersión del conjunto de valores de una estimación. En la Figura 1, se usa como ejemplo un gráfico que representa un conjunto de disparos a una diana:

- A. Alta precisión por estar los disparos cercanos entre sí y alta exactitud al dar en el centro de la diana.
- B. Alta precisión al igual que en el caso anterior y baja exactitud al estar los disparos alejados del centro de la diana.
- C. Baja precisión por estar los disparos separados entre sí y alta exactitud por estar los disparos repartidos por el centro de la diana.
- D. Baja precisión y exactitud ya que los disparos están separados entre sí y alejados del centro de la diana.

Ambos conceptos son independientes y el resultado de un IPS será de mejor calidad cuanto mayor sea la exactitud y la precisión [10] [12] [13].

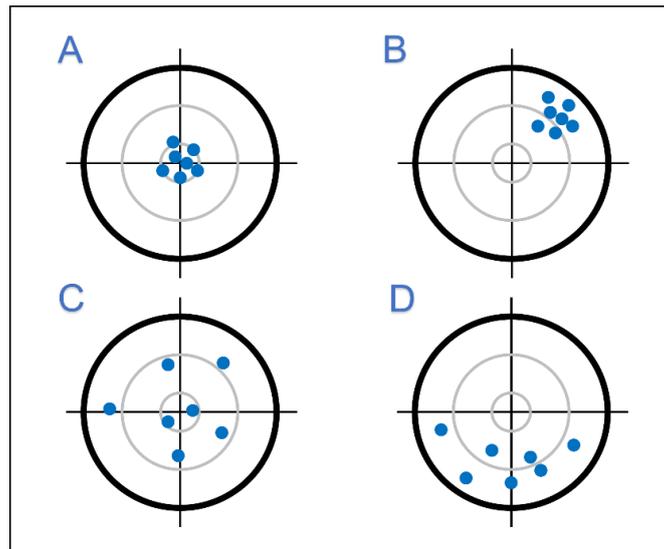


Figura 1. Ejemplo de exactitud y precisión.

3.1.2 Tecnologías

Los principales tipos de tecnologías usadas en posicionamiento en interiores son [13] [14] [15]:

- **Señales de Radio Frecuencia:** Wi-Fi, Bluetooth, ZigBee, RIFD (en inglés *Radio Frequency Identification*) o UWB (en inglés *Ultrawideband*).
- **Sonido:** ultrasonido o el uso de señales acústicas del entorno.
- **Ópticas:** infrarrojo o comunicación mediante luz visible o VLC (en inglés *Visible Light Communication*).
- **Magnéticas:** campos magnéticos artificiales o el generado naturalmente por la Tierra.
- **Híbridas:** resulta en la combinación de dos o más tecnologías.

3.1.3 Métodos

Algunos de los métodos principales y más empleados para estimar la posición de un objeto o persona, se describen a continuación [13]:

3.1.3.1 Proximidad

Aunque no es uno de los métodos más empleados, al no alcanzar alta precisión en la estimación de la posición, es de los más fáciles de implementar. Se asume que la posición de la persona u objeto en cuestión es la misma que la de la baliza más

próxima a este, y por tanto, se recibe el valor más alto del Indicador de la Intensidad de Señal Recibida o RSSI (en inglés *Received Signal Strength Indicator*) [10] [15].

3.1.3.2 Triangulación y Trilateración

En inglés, los dos métodos son variantes de la triangulación y se denominan *Multiangulation* y *Multilateration*, respectivamente. Esto es porque ambos permiten obtener las posiciones de un objeto o persona, usando la geometría de los triángulos. La diferencia principal reside en la información previa requerida.

Para el método de triangulación se debe conocer la distancia entre dos puntos de referencia y, medir dos valores que corresponden a los ángulos de llegada o AoA (en inglés, *Angle of Arrival*), uno por cada punto de referencia. En la Figura 2, se ve un gráfico que ilustra el funcionamiento de este método.

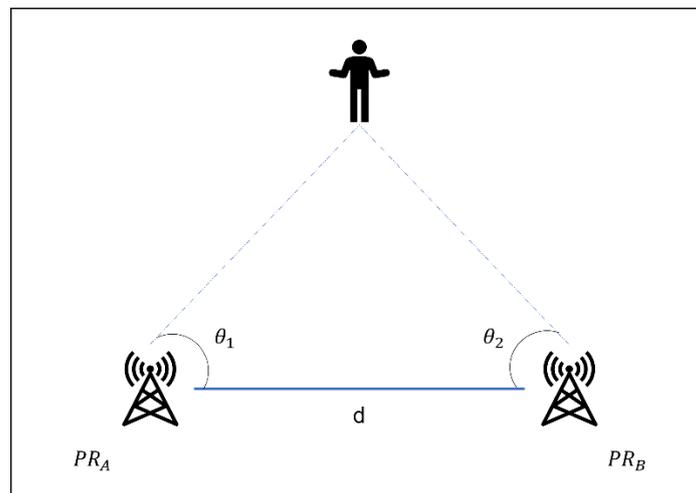


Figura 2. Triangulación.

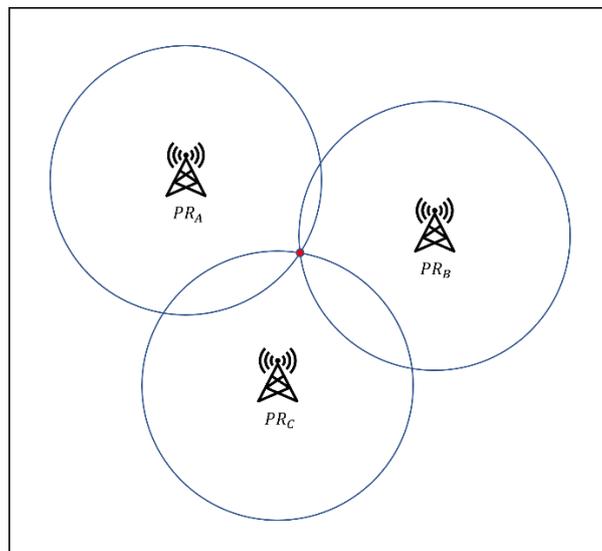


Figura 3. Trilateración.

En trilateración, para cada punto de referencia se trazará un círculo de radio la distancia que lo separa del objeto o persona cuya posición se busca estimar. Luego, con al menos tres puntos de referencia, el resultado del método será la intersección de los círculos. En la Figura 3, se puede apreciar visualmente el funcionamiento de esta técnica [10] [15].

3.1.3.3 Fingerprinting

El análisis de escena o *fingerprinting*, es un método que consiste en realizar un estudio previo del entorno en donde se implementará el IPS. Esto se conoce como fase de calibrado u *offline*. De esta manera, se elabora una base de datos en la que se recoge información sobre los parámetros característicos de la tecnología en uso. En la Figura 4, se puede observar un esquema de los pasos a seguir para el proceso de captura de datos.

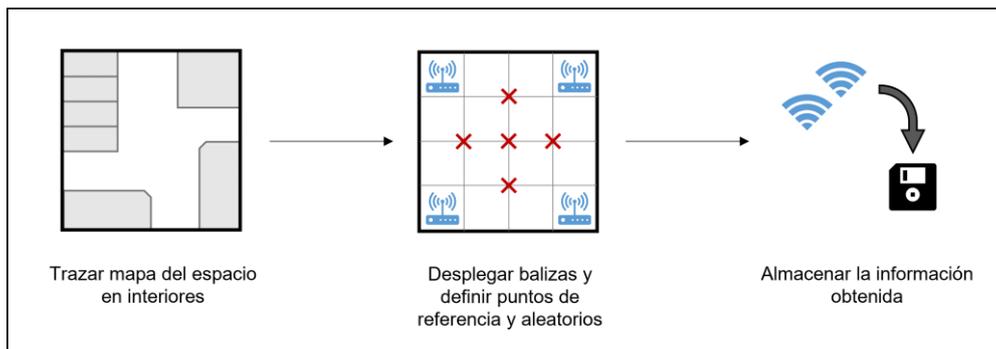


Figura 4. Fase de calibrado.

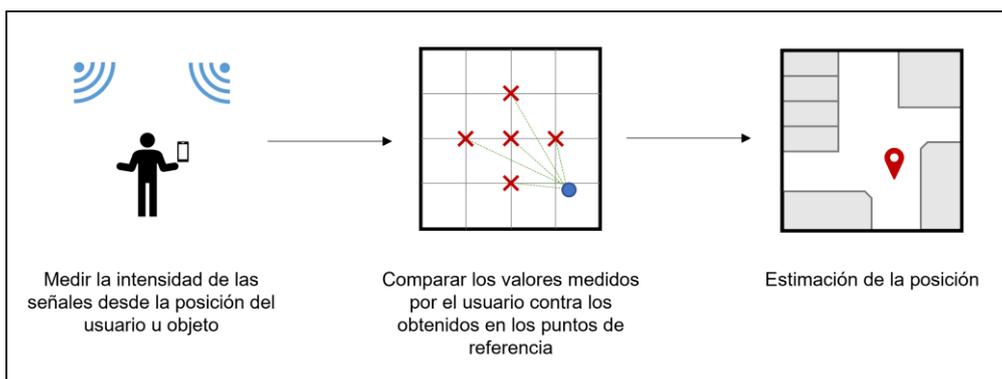


Figura 5. Fase de posicionamiento.

Después, tiene lugar la fase de posicionamiento u *online*, que trata de estimar la posición de una persona u objeto, comparando los valores obtenidos en el momento y los previamente almacenados. En esta segunda fase, se pueden utilizar diferentes algoritmos para estimar la posición del objeto o persona en cuestión. El más aplicado, por su fácil comprensión e implementación es el NN (en inglés *Nearest Neighbor*) y

sus variantes, kNN y WKNN (en inglés *k-Nearest Neighbor* y *Weighted k-Nearest Neighbor*). Las posibilidades son amplias, se pueden usar algoritmos como Proceso Gaussiano, Inferencia Bayesiana e incluso modelos basados en Inteligencia Artificial o AI (en inglés *Artificial Intelligence*). En la Figura 5 se observa un esquema de los pasos de la fase posicionamiento [10] [13] [15] [16].

3.2 ROBOMAP

ROBOMAP es el nombre de la plataforma robótica creada por el IUCES en la línea de investigación de Sistemas de Posicionamiento en Interiores. La metodología actual que utiliza el equipo de investigación es el análisis de escena o *fingerprinting*, usando balizas *Bluetooth Low Energy* (BLE) y almacenando información sobre RSSI (en inglés *Received Signal Strength Indicator*).

El proceso de captura de datos durante el análisis de escena puede llegar a consumir bastante tiempo y, además, se deben tener en cuenta múltiples factores que pueden afectar a la emisión y recepción de las señales, como son la temperatura y humedad del ambiente, la posición y orientación del dispositivo receptor, entre otros. Con la idea de automatizar este proceso, que también se conoce como campaña de recogida de datos, nace ROBOMAP (véase Figura 6).

ROBOMAP se ha ido creando por módulos a medida que diferentes miembros del IUCES y estudiantes de grado de la Universidad de Las Palmas de Gran Canaria (ULPGC), han trabajado en el proyecto como parte de sus Prácticas Externas (PE) y/o Trabajos de Fin de Grado o Master (TFG y TFM respectivamente). Todos ellos se comunican entre sí utilizando la arquitectura cliente-servidor con el protocolo UDP (en inglés *User Datagram Protocol*).

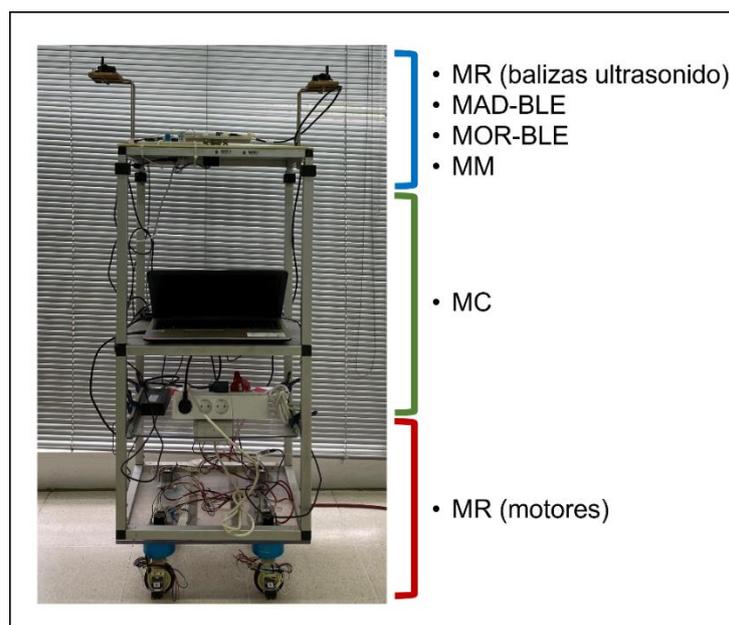


Figura 6. ROBOMAP

Durante la elaboración de este trabajo se creó un documento que reúne toda la información de ROBOMAP hasta ahora, que incluye también una serie de directrices para realizar de manera individual una campaña de recogida de datos (ver Anexo 1. Manual ROBOMAP).

3.2.1 Módulo ROBOMAP (MR)

Los compañeros José Carlos Rodríguez Rodríguez y Alberto Ramos Sánchez dotaron a la plataforma de movimiento omnidireccional que, con balizas ultrasónicas determinan su posición y luego, indican la ruta a seguir dentro del espacio designado [17].

Con la extensión para la herramienta Blender [18] desarrollada por Alberto [19] y el programa en C++ de José Carlos, es posible diseñar el entorno de interiores en 3D, definir la ruta a seguir por ROBOMAP junto con la posición de las balizas ultrasonido y/o las balizas BLE. Luego, se establece una comunicación entre ambas partes, donde la extensión transmitirá las coordenadas de los puntos al programa C++, el cual, convertirá la información obtenida en órdenes para los motores, consiguiendo así, el desplazamiento de la plataforma por los diferentes puntos de una campaña.

En la Figura 7, se observa un ejemplo del programa Blender con la extensión “Robotcontrol editor toolbox” de Alberto. Como se puede ver, se ha diseñado en 3D el entorno de interiores de prueba, la biblioteca del IUCES y, se está definiendo la ruta a seguir por ROBOMAP, estando esta, representada por el cubo en la zona central.

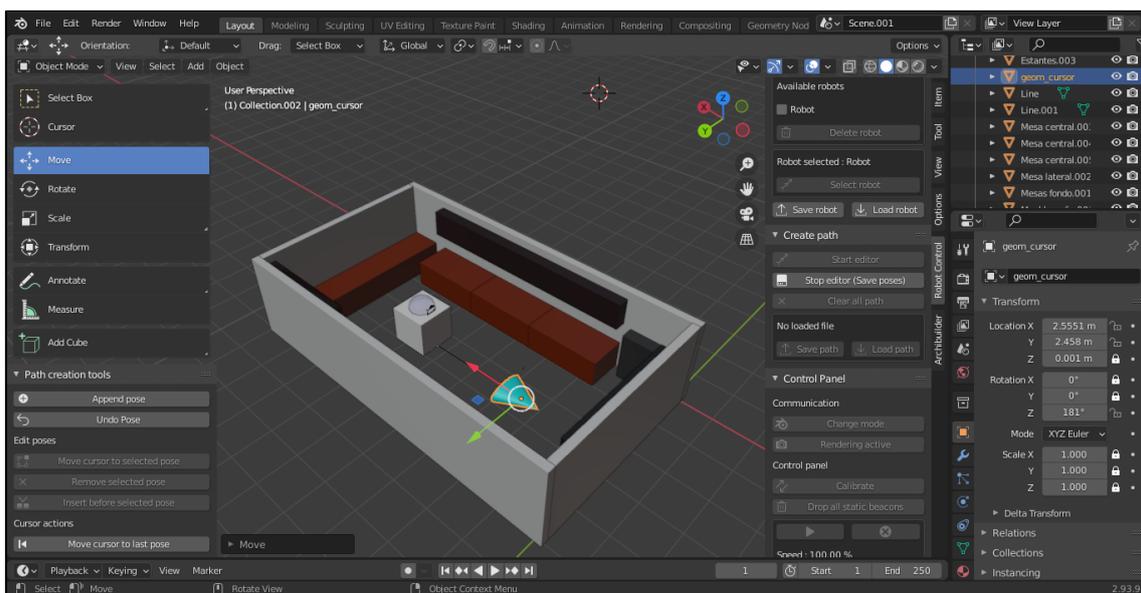


Figura 7. Ejemplo diseño entorno de interiores y ruta a seguir en Blender.

En la Figura 8 y Figura 9, se tienen dos imágenes en las que se indica las piezas que participan en el funcionamiento del MR. Primero, la serie de 4 pares de motores conectados entre sí por un anillo deslizante pasante Senring, esto es, una pieza mecánica que permite rotar en 360° cada uno de los motores de tracción (los que están pegados al suelo). Segundo, se muestra la parte superior de la plataforma marcando en azul las balizas de ultrasonido que permiten determinar la posición de esta.

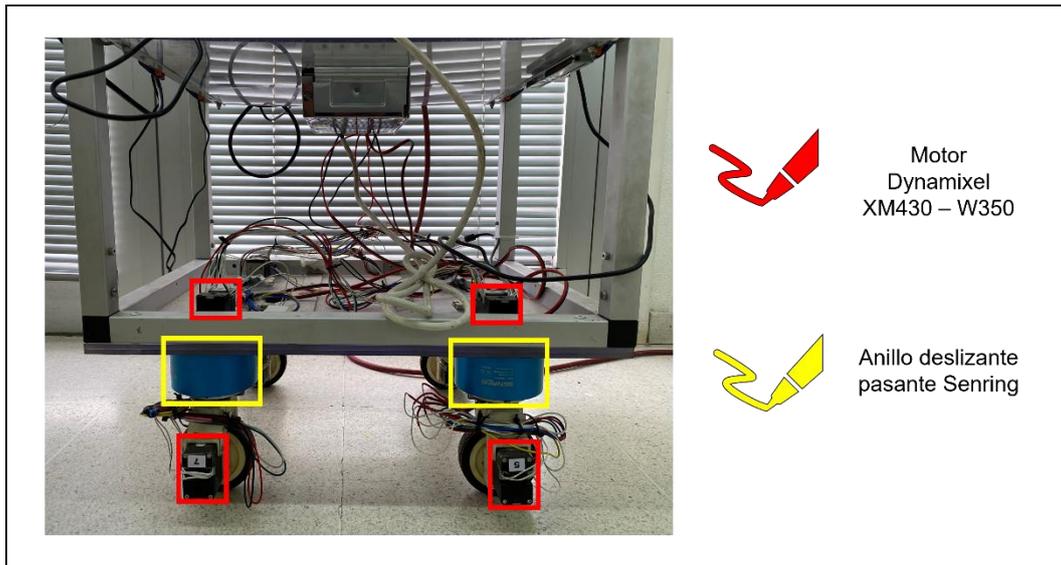


Figura 8. Componentes movimiento omnidireccional ROBOMAP.

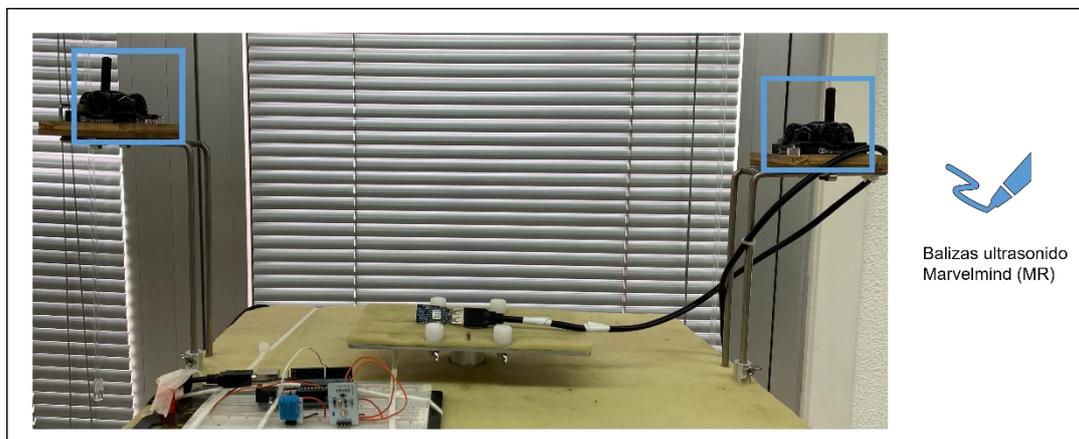


Figura 9. Balizas ultrasonido de ROBOMAP.

3.2.2 Módulo Meteorológico (MM)

David Suárez Suárez desarrolló, utilizando una placa Arduino y una serie de sensores, un sistema de toma de datos de la luminosidad, temperatura y humedad del ambiente en la zona de trabajo. Estos valores son importantes ya que pueden influir en la emisión y recepción de las señales Bluetooth [20]. En cierto modo, se

puede decir que se diseñó e implementó una estación meteorológica, la cual, se ve en el recuadro naranja de la Figura 10.

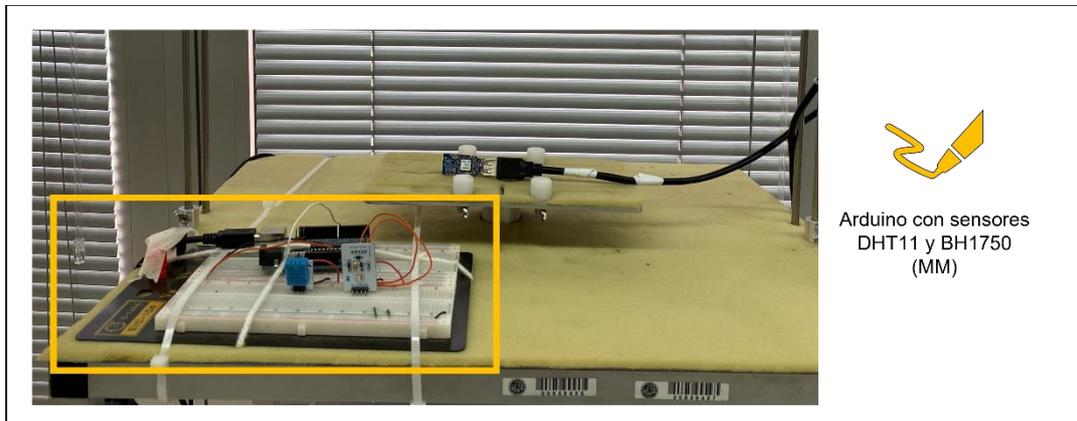


Figura 10. Estación meteorológica de ROBOMAP.

El programa de Arduino está diseñado de tal manera que funciona como un servidor que espera la recepción de peticiones y las procesa devolviendo los valores obtenidos de cada sensor. Luego, al estar conectado al portátil de la plataforma, es su correspondiente módulo en Python el que actúa como intermediario entre la placa y el resto de las partes durante una campaña.

3.2.3 Módulo de Adquisición de Datos BLE (MAD-BLE)

Los compañeros Iru Nervey Navarro e Yousuf Boutahar El Maachi, añadieron a la plataforma el aparato receptor de señales de balizas BLE (del inglés *Bluetooth Low Energy*) [21] [22].



Figura 11. Dongle receptor de señales Bluetooth.

Aunque ya se usaba este elemento en pruebas previas incluso de la aparición de la plataforma, hasta entonces no se había incluido en la misma. Así, ellos pusieron el receptor sobre la base del orientador para obtener medidas en diferentes ángulos (véase Figura 11). Además, Iru e Yousuf fueron los encargados de implementar el código que recopila y filtra la información recogida de las señales emitidas por las balizas BLE.

3.2.4 Módulo Orientador del Receptor BLE (MOR-BLE)

Nuevamente el compañero David fue quién desarrolló la funcionalidad de este MOR-BLE, utilizando también la arquitectura cliente-servidor y protocolo UDP [20]. Sin embargo, al igual que el módulo anterior, no se llegó a integrar en la plataforma hasta más adelante con el trabajo realizado por Yousuf e Iru.

En la Figura 12, se muestra lo que es la versión final de este módulo, una base sobre la que posar el receptor de señales BLE. Por otro lado, en la Figura 13, se ve la otra cara de la moneda, es decir, el cómo se integró en la parte superior de ROBOMAP el motor y la placa de alimentación y programación de este.

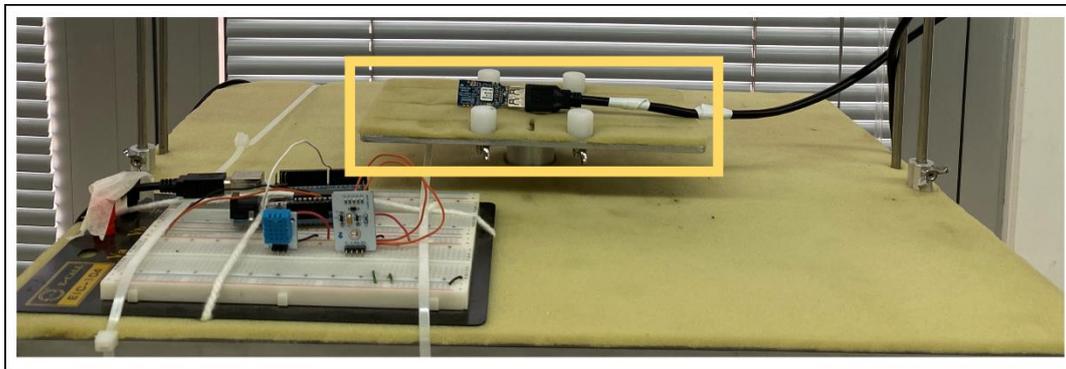


Figura 12. Plataforma del módulo orientador.

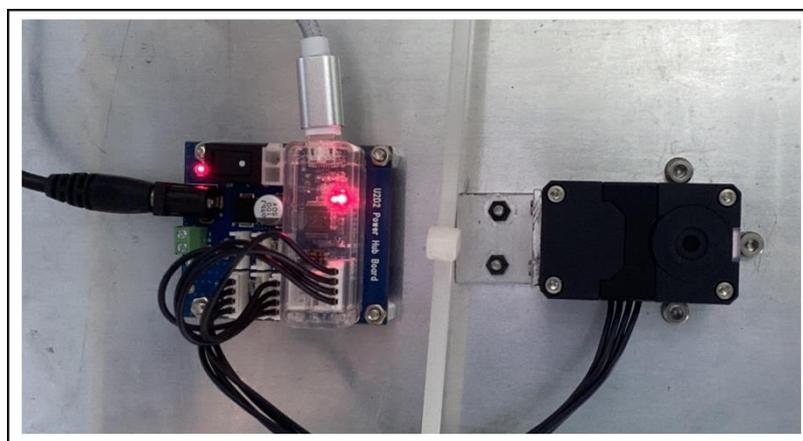


Figura 13. Componentes módulo orientador.

3.2.5 Módulo Controlador (MC)

Yousuf e Iru, diseñaron e implementaron lo que se conoce como el director de la plataforma robótica. Es el programa principal de una campaña de recogida de datos, esto es, el que se encarga de comunicarse con las demás partes, indicándoles cuándo activarse, cómo operar y dónde guardar los datos a lo largo de todo el proceso [21] [22].

Para ello, los compañeros diseñaron una serie de programas en Python que actúan como intermediarios entre el MC y el resto de módulos antes mencionados. Así, implementaron lo que llamaron servicios. Un servicio se define como una clase en Python que, en la mayoría de los casos representa un cliente para un servidor (conectarse, solicitar datos, ...) y, en otros, como por ejemplo *ServiceBBDD*, describe las diferentes formas con las que interactuar con la base de datos. Entonces, durante una campaña, desde el MC se crea una instancia de cada clase, consiguiendo así, acceso a los resultados y métodos de cada parte de ROBOMAP.

En las figuras a continuación, se muestran fragmentos de código todos parte del MC y, como se puede comprobar, es una instancia de la clase *Controller* la que se encarga de iniciar todos los procesos y registrar temporalmente los valores relativos a una captura antes de ser guardados en la base datos (ver Figura 14 y Figura 15).

```
if __name__ == "__main__":
    controller = Controller()
    controller.inicializarVariables()
    controller.nombreBD = controller.serviceBBDD.iniciarBaseDeDatos()

    controller.serviceAlerts.relayMsgToServer(("Bd", controller.nombreBD, str(datetime.now().strftime("%X - %d/%m/%Y"))))

    controller.startServices()
    controller.serviceAlerts.relayMsgToServer(("Ok", "Campaña iniciada", str(datetime.now().strftime("%X - %d/%m/%Y"))))

    controller.startCaptureSinRot()

    controller.serviceBBDD.confirmarDatosBBDD()

    controller.serviceAlerts.checkIfAllMacsDetected(controller.nombreBD, controller.fechaCaptura)
    controller.serviceAlerts.relayMsgToServer(("Ok", "Campaña finalizada\n", str(datetime.now().strftime("%X - %d/%m/%Y"))))
```

Figura 14. Instrucciones principales del archivo controlador.py.

```
def inicializarVariables(self):
    self.fechaCaptura = 0
    self.BLEjsonAllCaptures=[]
    self.weatherDataJson = 0
    self.robomapPos = [1,2,3,4]
    self.rotacion = 0
    self.sonido = False
    self.nombreBD = ""

def startServices(self):
    #self.serviceRoboMap.startServiceRobomap()
    self.serviceOrientador.startServiceOrientator()
    self.serviceWeather.startServiceWeather()
```

Figura 15. Métodos auxiliares para inicializar variables y servicios de una campaña.

En la Figura 16 se observa el método *startCaptureSinRot*, el cual, describe la serie de pasos y/o instrucciones involucradas en el proceso de captura de datos en un

punto pero sin el módulo orientador. Así, se aprecia cómo a través de los servicios es que el MC consigue comunicarse con el resto de módulos y ser este, el que maneja toda la información relativa a una campaña.

```
def startCaptureSinRot(self):
    self.weatherDataJson = self.serviceWeather.readServiceWeather()
    self.serviceAlerts.checkWeatherDataJson(self.weatherDataJson)

    self.fechaCaptura = datetime.now()
    self.BLEjsonAllCaptures = self.serviceBLE.startServiceBLE(self.duracion)
    self.serviceAlerts.checkNumberPacketsBLE(self.BLEjsonAllCaptures)

    self.serviceBBDD.guardarDatosBBDD(self.fechaCaptura, self.BLEjsonAllCaptures, self.weatherDataJson, self.robomapPos, self.rotacion)
```

Figura 16. Método que describe el proceso de una captura de datos en una posición.

Es importante mencionar que las figuras anteriores muestran fragmentos de código del MC que definen cómo realizar una campaña de recogida de datos pero sin el MOR-BLE ni el MR, esto es, por el inconveniente que su produjo al inicio de este trabajo y se comenta en el siguiente apartado.

3.2.6 Evolución

En 2019 fue cuando se creó lo que vendría siendo el primer prototipo de la plataforma robótica ROBOMAP. Consistía en una mesa con ruedas que transportaba un portátil y el receptor de señales Bluetooth. Este se encontraba incorporado a una pequeña base rotatoria hecha con LEGO, en su versión programable MINDSTORM, para poder captar las emisiones de las balizas BLE en diferentes ángulos. Por aquel entonces, toda la recolecta y procesado de datos, más la estimación de la posición se hacían manualmente con hojas Excel.

La segunda versión se construyó en 2020, lo que viene siendo el tronco inferior de la plataforma, dotando así, a ROBOMAP de movimiento omnidireccional. Como se puede ver en Figura 17, ya la estructura de la plataforma estaba diseñada para funcionar por niveles y de manera modular. Aún se realizaban algunos cálculos a mano pero ya se empezaba a incorporar la programación para la automatización de algunos procesos, por ejemplo, el desplazamiento de la plataforma o el uso de Matlab para las operaciones matemáticas.

No fue hasta 2022-2023 que se consiguió por fin alcanzar el estado actual del proyecto. Ahora ROBOMAP puede hacer casi de manera independiente la campaña de captura de datos, salvo por la excepción de que aún hay que estar pendiente del cableado eléctrico. Además, ya se emplea la programación y la intercomunicación de los diferentes módulos para conseguir así, automatizar por el completo el proceso de toma de datos. Sin embargo, se mantiene la limitación de la alimentación eléctrica, ya que, el cable no es infinito y la plataforma puede enredarse con él o algún otro elemento del entorno de interiores no previsto.



Figura 17. Evolución de ROBOMAP.

4. Desarrollo

El desarrollo de este trabajo se compone de tres partes principales, una por cada objetivo. Además de describir la solución para cada una de ellas, se proporciona información que permite entender el contexto de cada situación así como, aquellos datos relevantes investigados que ayudaron a alcanzar los objetivos propuestos para este TFG.

El último estadio de ROBOMAP era completamente funcional, hacía la recolección de datos durante una campaña sin problema alguno. Sin embargo, al momento de realizar pruebas de puesta en marcha, todas las balizas de ultrasonido estaban siendo detectadas incorrectamente, impidiendo así, el conocer la posición de la plataforma y, por tanto, el seguimiento de una ruta.

Afortunadamente, esto no bloqueó el desarrollo de este trabajo por completo, ya que, para dos de los objetivos a cumplir, no suponía ningún impedimento. En cambio, para el estudio de un sistema energético sí resultó ser una dificultad. En el apartado 4.1.4 Consumo, se comenta como se solventó dicho problema.

4.1 Sistema Energético (SE)

El objetivo de este apartado es realizar un estudio sobre el consumo energético de la plataforma robótica para, a continuación, hacer una propuesta de un sistema que dote de autonomía al mismo. De esta manera, se realiza un análisis del hardware que conforma a la plataforma.

Hasta el momento, ROBOMAP se encuentra conectado a la red eléctrica para alimentar los dispositivos que lo hacen funcional. Sin embargo, existe únicamente un componente que sí dispone de autonomía, el ordenador principal. Es por ello, por lo que no se ha tenido en cuenta su impacto en este estudio energético, es decir, no se ha analizado el consumo generado al cargar la batería del portátil.

4.1.1 Repaso conceptos electricidad

La **electricidad** es un conjunto de fenómenos producidos por el movimiento e interacción entre partículas cargadas, protones (carga positiva) y electrones (carga negativa). Además, es una forma de **energía** que puede manifestarse tanto en reposo (electricidad estática) como en movimiento (corriente eléctrica) [23] [24].

El **voltaje** es una magnitud física que indica la diferencia de potencial entre dos puntos de un circuito eléctrico. Debido a esto se le conoce también como diferencia de potencial o tensión eléctrica. En otras palabras, se puede decir que el voltaje es la

fuerza que induce el movimiento de las cargas eléctricas en un circuito. Se mide en voltios (V) y suele representarse mediante la letra V [24] [25] [26].

La **intensidad de corriente** es una magnitud física que dada una diferencia de potencial eléctrico, indica la cantidad de carga eléctrica que circula por un elemento conductor por unidad de tiempo. Se mide en amperios (A) y suele representarse con la letra I. Los dos tipos más comunes de corriente son la **corriente continua (CC)** y la **corriente alterna (CA)**. La diferencia entre ambas se encuentra en la forma en la que circulan las cargas por el medio conductor. En CC, se desplazan en un único sentido mientras que, en CA, el sentido y la magnitud de la corriente varía de forma cíclica a lo largo del tiempo. En la Figura 18 se puede apreciar gráficamente la comparación entre ambos tipos [24] [25] [26] [27].

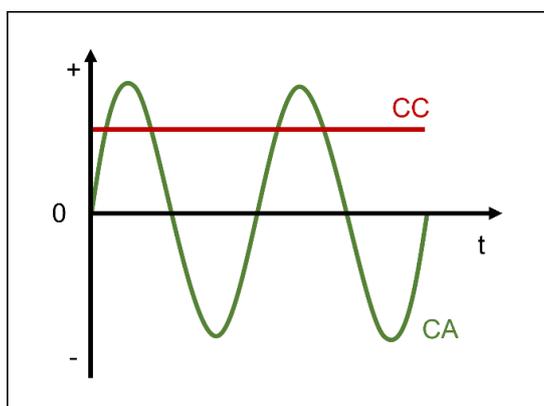


Figura 18. Tipos de corriente eléctrica.

La **energía** es la capacidad que posee un cuerpo para realizar un trabajo que, en consecuencia, implica un cambio en el mismo cuerpo u otros. Coloquialmente, se puede definir como la capacidad de hacer las cosas funcionar. Se puede manifestar y obtener de diferentes formas, dando lugar así, a distintos tipos de energía, como por ejemplo, la energía eléctrica que tiene su origen en la corriente eléctrica y, puede ser producida al aprovechar la radiación solar con paneles fotovoltaicas [24] [28] [29].

La **potencia eléctrica** se define como la cantidad de energía eléctrica disipada o generada por un elemento en un instante de tiempo. Se mide en vatios (W) y suele representarse mediante la letra P. Según el tipo de corriente empleada, la potencia eléctrica se obtiene de diferente manera. La expresión de potencia eléctrica en CC, se define como el producto del voltaje y la intensidad. En cambio, en CA, la potencia es un poco más compleja y existen tres tipos diferentes. Todos ellos se describen en la Tabla 1 [24] [30] [31].

Tabla 1. Tipos de potencia en corriente alterna.

Tipo	Expresión	Unidad
Potencia Activa (P) Única que resulta en energía útil (trabajo). φ corresponde con el factor de potencia.	$P = V \cdot I \cdot \cos(\varphi)$	Vatios (W)

<p>Potencia Reactiva (Q) Es aquella consumida por ciertos elementos de un circuito, por ejemplo, para establecer los campos magnéticos de las bobinas.</p>	$Q = V \cdot I \cdot \text{sen}(\varphi)$	<p>Voltio amperio reactivo (VAR)</p>
<p>Potencia Aparente (S) Es la potencia total que consume un circuito eléctrico.</p>	$S = V \cdot I$	<p>Voltio amperio (VA)</p>

La **energía eléctrica** o consumo eléctrico viene dada por la potencia eléctrica de un elemento durante un período de 1 hora. Se mide en kilovatio-hora (kWh) y se suele representar con la letra E [24] [28] [29].

4.1.2 Aparatos de medición

En función de la magnitud eléctrica a medir, el tipo de corriente y la situación en la que se trabaje, existen una gran variedad de aparatos de medición de entre los que elegir. En la Figura 19, se muestran algunos dispositivos de medición. Los que se encuentran en la izquierda de la imagen se destinan a un público más experto, al implicar trabajar directamente con los componentes eléctricos, mientras que, por el contrario, los de la derecha están pensados para cualquier usuario al no requerir de ningún proceso de instalación exhaustivo [32].



Figura 19. Aparatos de medición. Imágenes extraídas de [32].

4.1.3 Estudio ROBOMAP

Para poder dotar de autonomía a ROBOMAP, hay que remplazar las fuentes de alimentación y/o los adaptadores de corriente por algún tipo de batería. Entonces, se investigó y analizó primero, de dónde provenía la energía. Como se observa en la

Figura 20 y Figura 21, la electricidad empleada para poner en marcha a la plataforma proviene de una toma de corriente del entorno de interiores que luego, se distribuye mediante una regleta con protección para sobrecarga [33]. A esta, se conectan tres adaptadores de corriente (dispositivos que convierten la corriente alterna a corriente continua) que terminan de proporcionar la energía necesaria al resto de los aparatos de la plataforma.

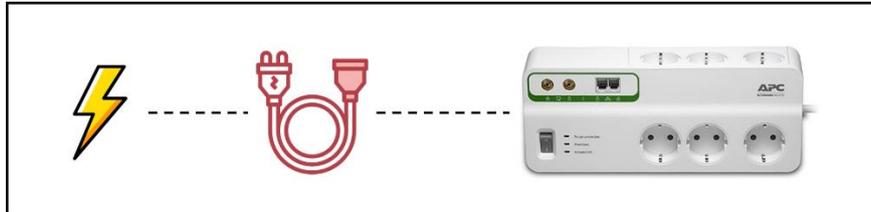


Figura 20. Esquema alimentación principal. Iconos extraídos de [33] [34].

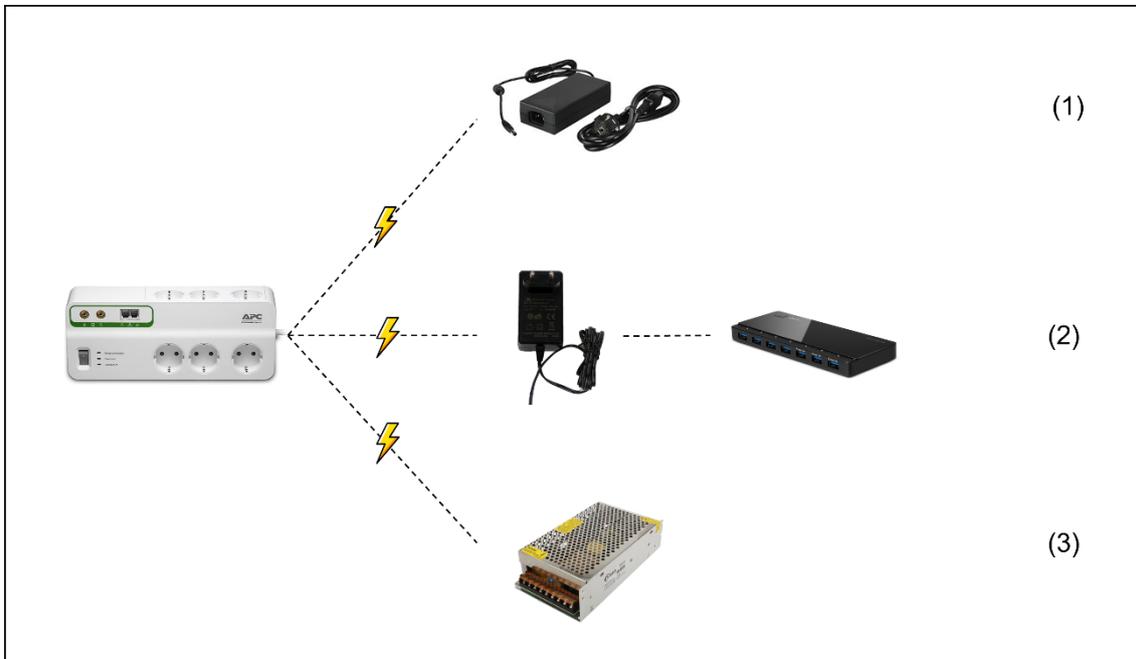


Figura 21. Fuentes de alimentación ROBOMAP.

Las fuentes de alimentación y/o adaptadores de corriente ilustrados antes se describen a continuación:

1. Adaptador DVE
 - Entrada: 100-240V / 50-60Hz / 2A (CA)
 - Salida: 12V / 5A (CC)
2. Adaptador 3YE
 - Entrada: 100-240V / 50-60Hz / 1A (CA)
 - Salida: 12V / 2A (CC)
3. Fuente NIMO Electronics
 - Entrada: 110-220V \pm 10% / 47-65Hz (CA)
 - Salida: 12V / 20.8A (CC)

Con la Figura 22, Figura 23 y Figura 24, se muestran las conexiones entre las fuentes y el resto de los aparatos electrónicos de la plataforma.

En resumen, la energía eléctrica proviene de una toma de corriente y permite a ROBOMAP funcionar. Esta energía, pasa al resto de componentes por medio de la regleta blanca y los diferentes adaptadores de corriente y/o fuentes de alimentación (la tensión eléctrica pasa de 220-240V a 12V). Por consiguiente, se consigue conocer una magnitud imprescindible para la propuesta de autonomía, el voltaje de salida.

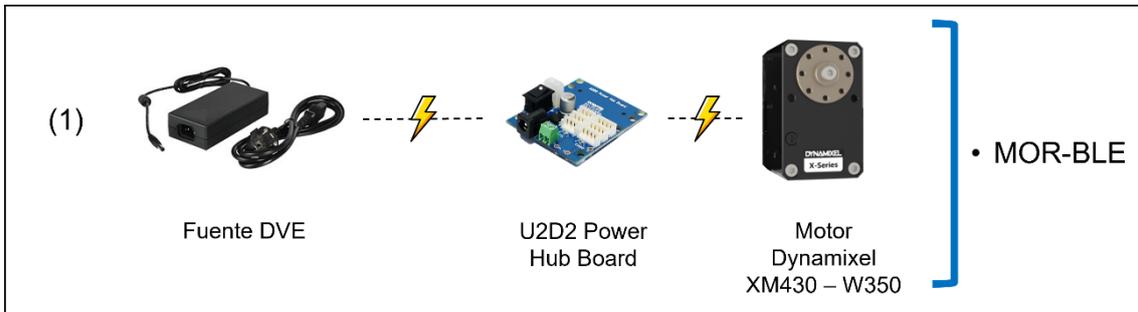


Figura 22. Esquema fuente de alimentación (1).

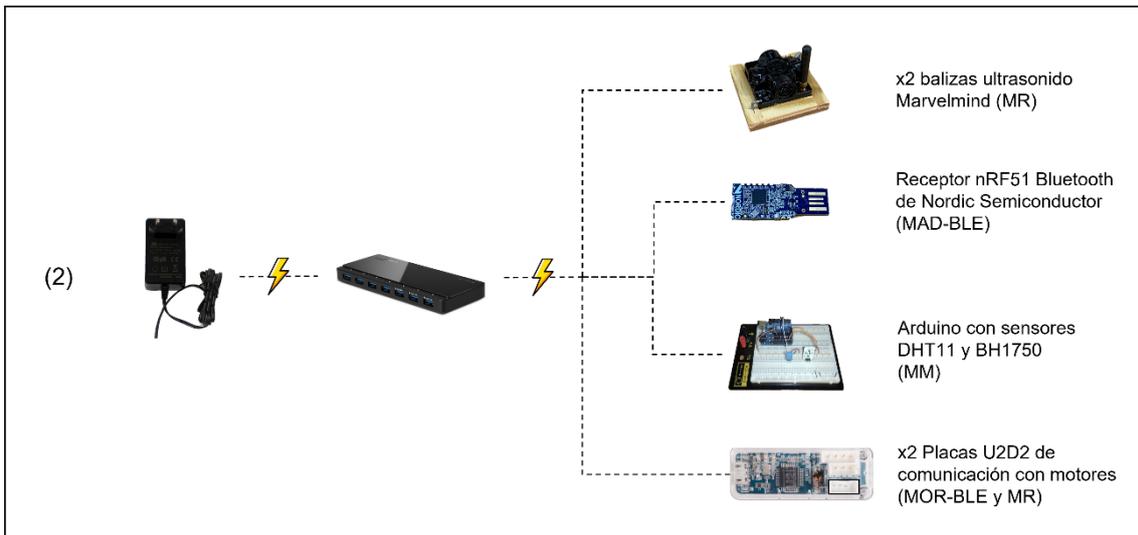


Figura 23. Esquema fuente de alimentación (2).

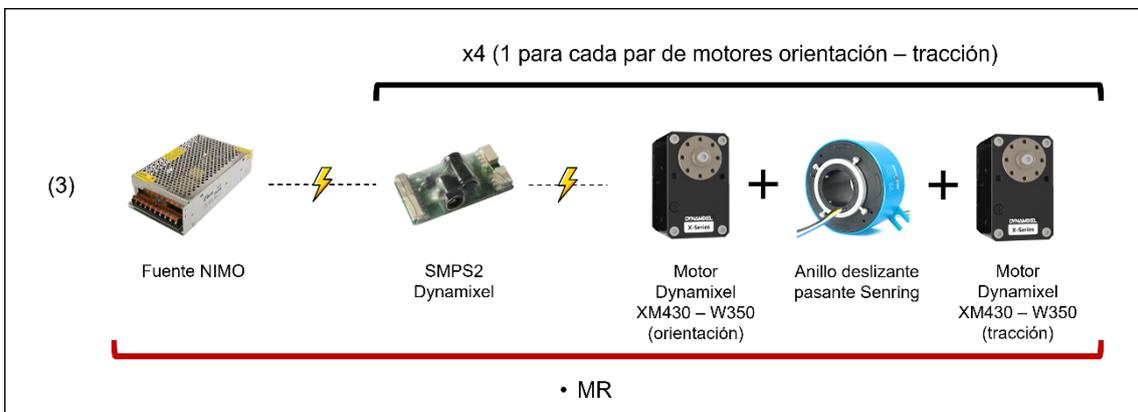


Figura 24. Esquema fuente de alimentación (3).

4.1.4 Consumo

Una vez conocido el voltaje con el que trabajan los aparatos de la plataforma robótica, falta saber la potencia de estos para así, poder estimar la capacidad de la propuesta de autonomía.

Antes de medir el consumo, se tuvo en cuenta la petición de los tutores de intentar no alterar el estado de los circuitos eléctricos de la plataforma, puesto que, se quiere evitar posibles fallas adicionales en el proyecto. Por consiguiente, usar el multímetro queda descartado al requerir abrir el circuito y conectar el aparato para poder medir. Luego, los aparatos y/o sensores que se colocan en el cuadro eléctrico, además de ser caros, precios superiores a 100€, implicaba también el conocer cómo estaba distribuido el cableado de la zona donde se estaban haciendo pruebas. Por último, la pinza amperimétrica funciona no solo como un multímetro si no también, permite medir las magnitudes físicas simplemente rodeando los cables con la parte superior del aparato. Sin embargo, hay que quitar la cubierta protectora y/o el aislamiento y tratar con los cables de manera individual, ya que, si se intenta medir todo el cable completo o varios cables a la vez, se pueden obtener lecturas con valor 0. Esto se debe a que el flujo de corriente entre cables se anula entre sí.

En la Figura 25, se muestran las partes del típico cable negro, llamado cable manguera, usado en la mayoría de los aparatos eléctricos domésticos, cuyo interior puede variar según el caso, añadiendo más conexiones internas para la comunicación de información por ejemplo.

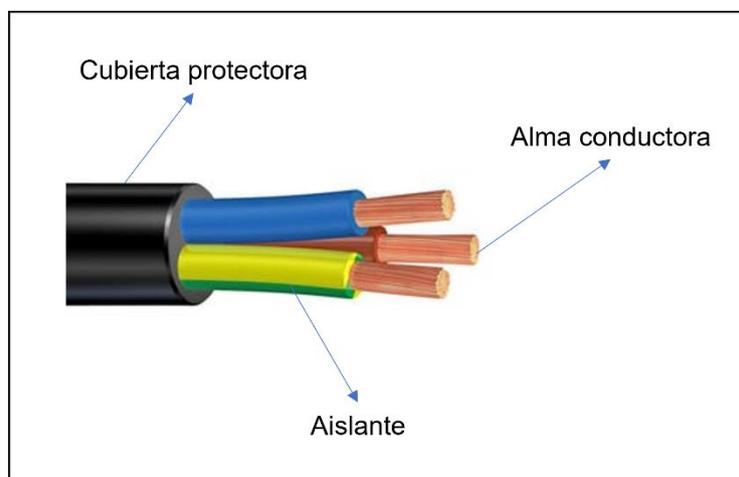


Figura 25. Cable manguera y sus partes.

Entonces, dadas las condiciones especificadas por los tutores y el contexto de la situación para la medición, el principal aparato empleado fue el enchufe monitor de corriente. Aunque la mayoría del cableado de ROBOMAP está protegido por una cubierta protectora, en la parte del módulo MR los cables están algo más expuestos, como los de la Figura 26, lo que permite usar la pinza amperimétrica en ese caso.

Una vez seleccionados los aparatos de medición a usar, se conoce también que la mayoría de los valores obtenidos serán en corriente alterna. Por ello, más adelante, habrá que realizar la conversión a CC, pero no solo por ser el tipo de corriente usado por los dispositivos, sino también porque es el que sí permite almacenar energía.

La idea detrás de este apartado es que la potencia activa en CA corresponde con la energía utilizada en CC. Eso sí, al no tratarse de una situación ideal donde no existen pérdidas, se debe tener en cuenta que no toda la energía en CA será empleada en CC. Esto es porque las fuentes de alimentación no son 100% eficientes, existe un cierto porcentaje de la energía convertida que se pierde en el proceso como calor. Y, en la expresión de potencia activa, se tiene en consideración lo que se conoce como el factor de potencia, magnitud que indica que tanta energía se convierte realmente en trabajo.

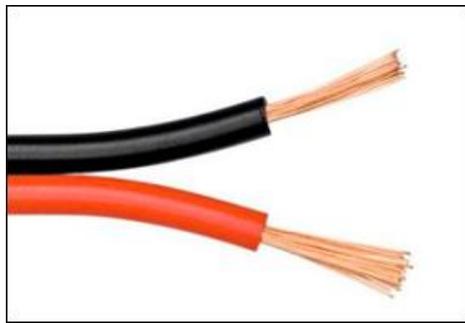


Figura 26. Cable fase positivo.

Para solucionar este inconveniente, al saber que la fuente de alimentación del MR tiene una eficiencia del 85% y que los demás adaptadores de corriente tienen un valor similar, se usará un factor de corrección de 1.2 en la conversión de CA a CC.

Los valores de las magnitudes eléctricas obtenidas con el enchufe monitor de consumo son las que se muestran en la Tabla 2. Todas estas mediciones se realizaron de forma estática, esto es, estando todo conectado y encendido, a la espera de recibir órdenes, lo que se conoce en inglés como *standby*. Asimismo, para cada magnitud, se indica el valor máximo leído durante un período de 2 minutos.

Tabla 2. Resultados mediciones en modo de espera (*standby*).

Fuente de alimentación	Voltaje (V)	Intensidad (A)	Factor de potencia (%)	Potencia activa (W)	Consumo (kWh)
DVE (1)	230	0.02	0.40	0.70	0.00070
3YE (2)	230	0.04	0.40	3.51	0.00351
NIMO (3)	230	0.11	0.40	11.23	0.01123
Regleta PCA	230	0.17	0.40	14.82	0.01482

Ya conocidos los valores de voltaje e intensidad de las diferentes fuentes en *standby*, hay que medir nuevamente los mismos aparatos pero esta vez durante una campaña de recogida de datos, momento en el cual, si realizan trabajo dictado por las órdenes recibidas del MC. No obstante, con la falla en la detección de las balizas de ultrasonido, no se puede realizar una campaña completa, al no conocer la posición de la plataforma y, por ende, no poder seguir una ruta. Por esto, se probó a realizar capturas que no involucraban al módulo MR. Los resultados obtenidos fueron casi idénticos a los del modo de espera, salvo por la fuente DVE o (1), cuyos valores están reflejados en la Tabla 3.

Tabla 3. Resultado mediciones puesta en marcha del MOR-BLE.

Fuente de alimentación	Voltaje (V)	Intensidad (A)	Factor de potencia (%)	Potencia activa (W)	Consumo (kWh)
DVE (1)	230	0.05	0.40	2.89	0.00289

Entonces, faltaba por evaluar el consumo del MR. De antemano, se estimaba que los motores fuesen los aparatos con mayor consumo, ya que, son los encargados de desplazar a la plataforma con peso aproximado de 10 kg. Para ello, se aprovechó la herramienta desarrollada por el compañero Alberto, una extensión o *add-on* en inglés, llamada “Robotcontrol” [19]. Así, se pudo controlar a ROBOMAP de manera remota mientras se analizaba la potencia de los motores.

En la Tabla 4, se observan las potencias mínimas y máximas medidas durante la puesta en marcha de los motores que conforman el MR, en dos puntos diferentes, uno midiendo la fuente NIMO (3) y el otro, la regleta PCA.

Tabla 4. Resultados mediciones durante la puesta en marcha del MR.

Fuente de alimentación	Voltaje (V)	Intensidad (A)	Factor de potencia (%)	Potencia activa (W)	Consumo (kWh)
Mínimo NIMO	230	0.11	0.41	10.09	0.01009
Máximo NIMO	230	0.30	0.45	28.83	0.02883
Mínimo regleta	230	0.16	0.37	14.43	0.01443
Máximo regleta	230	0.27	0.42	27.83	0.02783

Tabla 5. Resultados conversiones de corriente alterna a corriente continua.

Fuente de alimentación	Potencia (W)	Factor de corrección	Voltaje (V)	Intensidad (A)	Consumo (kWh)
DVE (1)	0.70	1.2	12	0.07	0.00007
3YE (2)	3.51	1.2	12	0.35	0.00035
NIMO (3)	11.23	1.2	12	1.12	0.00112
Regleta PCA	14.82	1.2	12	1.48	0.00148
DVE (1)	2.89	1.2	12	0.29	0.00029
Mínimo NIMO	10.09	1.2	12	1.01	0.00101
Máximo NIMO	28.83	1.2	12	2.88	0.00288
Mínimo regleta	14.43	1.2	12	1.44	0.00144
Máximo regleta	27.83	1.2	12	2.78	0.00278

Finalmente, al conocer que el voltaje de entrada de todos los componentes de ROBOMAP es de 12V y, la potencia activa de las diferentes fuentes de alimentación, se puede calcular la intensidad proporcionada por estas en corriente continua. Antes, a los valores de potencia hay que aplicarles el factor de corrección para así, tener en cuenta las pérdidas en forma de calor de las fuentes y posibles sobrecargas. Los valores de la Tabla 5, se consiguieron al aplicar la siguiente expresión:

Ecuación 1. Intensidad de corriente.

$$I = \frac{P \cdot \text{factor de corrección}}{V}$$

4.1.5 Desviación

En el apartado anterior, los resultados obtenidos de la potencia de las fuentes de alimentación fueron medidos con el enchufe monitor de consumo. Sin embargo, se probó a contrastar las mensuraciones anteriores utilizando la fórmulas reflejadas a continuación:

Ecuación 2. Potencia en corriente alterna.

$$P = V \cdot I \cdot \cos(\varphi) \qquad P = V_{rms} \cdot I_{rms} \cdot \cos(\varphi)$$

La diferencia radica en que la segunda emplea los RMS (en inglés *Root Mean Squared*) o valores eficaces del voltaje y la intensidad. Ambos, dada una señal de corriente alterna, indican la cantidad de energía o trabajo equivalente a una señal en corriente continua. En otras palabras, son un par de valores que tienen en cuenta el comportamiento de una señal de onda. Sus expresiones son simplemente el valor pico de cada magnitud (el valor máximo alcanzado) dividido por la raíz cuadrada de dos [35].

Ecuación 3. Voltaje e intensidad de corriente eficaz.

$$V_{rms} = \frac{V_{pico}}{\sqrt{2}} \qquad I_{rms} = \frac{I_{pico}}{\sqrt{2}}$$

Manteniendo los valores del voltaje, intensidad y factor de potencia intactos, se procedió a realizar los cálculos según las nuevas expresiones, siendo la potencia A la de la Tabla 1 y la B, la que incluye los valores eficaces o RMS. Además, se calcula la desviación de los resultados para cada caso frente a las potencias medidas con el enchufe monitor de consumo.

Tabla 6. Resultados desviación potencias obtenidas por diferentes métodos.

Fuente de alimentación	Potencia activa enchufe	Potencia activa A	Desviación (enchufe - A)	Potencia activa B	Desviación (enchufe - B)
DVE (1)	0.70	4.60	3.9	2.30	1.6
3YE (2)	3.51	9.20	5.69	4.60	1.09
NIMO (3)	11.23	25.30	14.07	12.65	1.42
Regleta PCA	14.82	39.10	24.28	19.55	4.73
DVE (1)	2.89	11.50	8.61	5.75	2.86
Mínimo NIMO	10.09	25.30	15.21	12.65	2.56
Máximo NIMO	28.83	69.00	40.17	34.50	5.67
Mínimo regleta	14.43	36.80	22.37	18.40	3.97
Máximo regleta	27.83	62.10	34.27	31.05	3.22

Tal y como se puede observar en la Tabla 6, existe una gran diferencia entre la potencia medida por el enchufe y la calculada por la expresión A. En cambio, la desviación de los valores obtenidos por el enchufe monitor de consumo y la expresión B, son levemente menos significativos.

De todas formas, para el resto del proyecto se siguió trabajando con los valores correspondientes al aparato de medición, el enchufe monitor de consumo.

4.1.6 Propuesta autonomía

Ya conocidas las dos magnitudes eléctricas claves, el voltaje y la intensidad, se puede estimar las características del sistema que dote de autonomía a ROBOMAP.

La idea principal es conseguir total libertad de movimiento de la plataforma, suprimiendo así la necesidad de que esté algún operario pendiente de que el cable de alimentación se enrede o atasque en algún punto del espacio de interiores. Para ello, se cree oportuno el uso de baterías.

Una batería o acumulador, es un dispositivo formado por una o varias celdas electroquímicas capaces de almacenar energía química y al convertirla, suministrar energía eléctrica [36] [37]. Existen una gran variedad de baterías diferentes según su uso o aplicación y su composición química interna. De esta forma, descartamos aquellas que no sean recargables, las basadas en níquel por ejemplo, y todas las que sean de grandes dimensiones y pesos, como las baterías de plomo-ácido usadas en los coches.

Para comparar y elegir entre los diferentes tipos y modelos de baterías hay que tener en cuenta sus 3 características principales [38] [39]:

- Voltaje o tensión en voltios (V).
- Capacidad en miliamperios hora (mAh) que indica la corriente eléctrica que la batería puede suministrar durante 1 hora de uso.
- Intensidad de corriente o descarga máxima en amperios (A).

Sin ninguna duda, la opción a elegir son las baterías de iones de litio o Li-ion. Al ser de desarrollo reciente, presentan múltiples ventajas en comparación con el resto de alternativas [36] [37] [40]:

- ✓ No suelen verse afectadas por el efecto memoria, fenómeno que ocurre cuando se carga una batería sin haber sido descargada por completo.
- ✓ Mayor densidad de energía sin sacrificar espacio o peso.
- ✓ Mayor rendimiento y bajo factor de autodescarga.
- ✓ Descarga lineal (durante su uso el voltaje proporcionado varía poco).

No obstante, también acarrearán algunas desventajas [36] [37] [40]:

- × Independientemente de su uso tienen una vida útil de 3 años.
- × Dada su composición química se aplican medidas de seguridad adicionales para evitar explosiones o incendios, lo que hace que su costo sea elevado.
- × Al ser susceptibles a los cambios de temperatura, requieren de circuitería que regule el voltaje y la temperatura de cada una de las celdas. Esto se conoce como BMS (en inglés *Battery Management System*).

A pesar de las desventajas antes mencionadas, los beneficios a favor de este tipo de batería tienen más peso y las hace por tanto, una de las mejores opciones del mercado actual.

Ahora, para conseguir alimentar a ROBOMAP, hay que conseguir un sistema que sea capaz de proporcionar 12V y según la

Tabla 5, al menos 3A, pero no solo por un instante sino durante un periodo de tiempo determinado. Por ejemplo, la batería Li-ion de LG MJ1 cuyas características son [41]:

- Voltaje: 3.7 V
- Capacidad: 3500 mAh
- Descarga máxima: 10 A
- Peso: 46 g

Por defecto, no satisface los requisitos energéticos de la plataforma. Entonces, ¿cómo se alcanzan esos 12V y al menos 3A? Conectando varias de estas baterías:

- Si se conectan en serie incrementa el voltaje proporcionado.
- Si se conectan en paralelo se consigue aumentar tanto la capacidad como la descarga máxima.

Por consiguiente, si se conectan cuatro baterías LG MJ1 en serie, se alcanza unos 14,8 voltios pero se mantienen la capacidad de 3500 mAh, lo que bastaría para 1h de uso. Luego, aproximadamente, para este caso, cada vez que se conecte un par de baterías en paralelo, se consigue 1 hora más de puesta en marcha. Asimismo, si se busca que ROBOMAP funcione durante un periodo de 3 horas y suministrando los 12V requeridos, se elaborarán 4 “paquetes” de 3 baterías en paralelo que luego se conectarán en serie para finalmente, por así decirlo, acabar con una matriz de 4 filas y 3 columnas de baterías capaz de suministrar 14,8V, 30A y capacidad 10500mAh.

En resumen, independientemente del modelo y marca de batería de Li-ion que se acabe usando, será necesario conectar múltiples de estas en serie y paralelo para

alcanzar el voltaje requerido y todo funcione durante el período de tiempo deseado. Además, será imprescindible el incorporar un BMS (en inglés *Battery Management System*) que permita controlar la carga, descarga y temperatura de cada batería del paquete creado para poder aumentar la vida útil del sistema de autonomía y también garantizar la seguridad de los usuarios, el proyecto y el entorno [42].

Finalmente, la propuesta de autonomía para ROBOMAP consiste en un *pack* de baterías, en este caso las de iones de litio o Li-ion por sus enormes ventajas y gran rendimiento; y un BMS que las gestione. De esta manera, se consigue sustituir los adaptadores de corriente y la regleta blanca por una única fuente de alimentación, además, se reduce también la cantidad de cableado de la plataforma. Esto se refleja en la Figura 27.

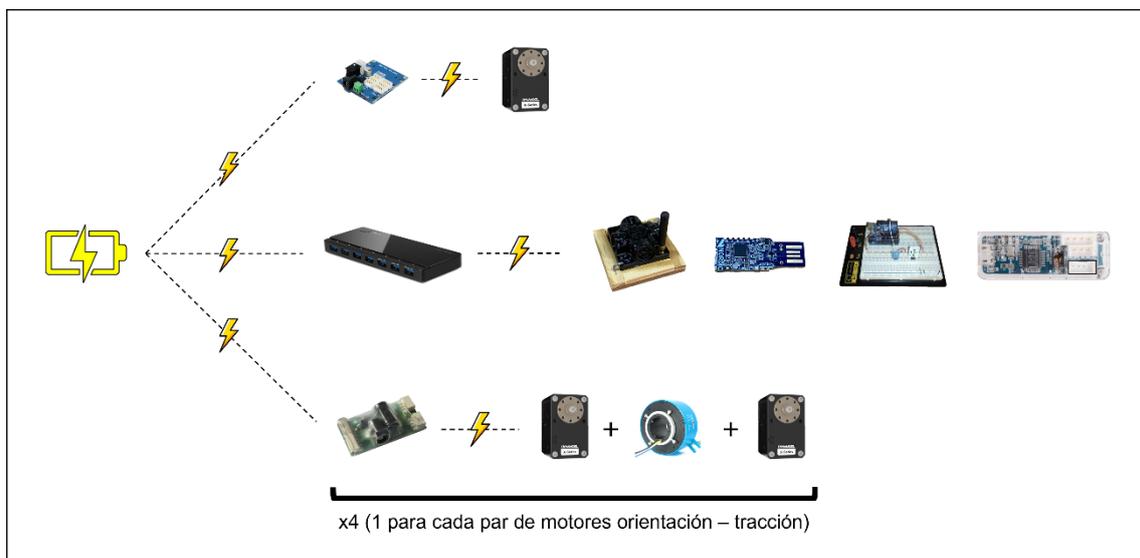


Figura 27. Esquema propuesta sistema energético.



Figura 28. Montaje baterías sin soldadura de VRUZEND. Imagen extraída de [43].

Por otro lado, las características del *pack* de baterías serán en función del presupuesto y la disponibilidad del material. Existen dos alternativas para obtener

y/o elaborar el sistema energético, o bien se compra todo el material por separado y se monta a mano la batería personalizada o, se acude a algún especialista que ayude a diseñarla y crearla. Un ejemplo curioso e interesante es la empresa VRUZEND [43] y sus productos, ya que tiene paquetes para elaborar baterías de manera más cómoda y sin necesidad de soldar, todo se conectan como si fueran piezas de un *puzzle* (ver Figura 28).

4.2 Sistema de alertas

Antes de este TFG, durante una campaña de recogida de datos, la plataforma robótica no contaba con ningún tipo de comprobación de errores ni de un sistema de visualización del estado funcional del mismo. La única manera de saber el estado de una campaña era observar las consolas desde donde se ejecutaban los programas en el ordenador principal. Esto, daba lugar en múltiples ocasiones, a situaciones en las que la información recopilada no era correcta y, al tener que reiniciar una campaña desde cero, se perdía un recurso muy preciado, tiempo.

La idea detrás de este apartado es la de dotar a ROBOMAP de un sistema de alertas que notifique al usuario de su estado funcional durante una campaña, así como, la de un módulo de control de errores que consistiese en una serie de pruebas sobre los datos procesados, señales recibidas, entre otros. Para la elaboración de esta nueva funcionalidad, se hizo hincapié en que era necesario mantener la arquitectura y métodos de comunicación que empleaban el resto de módulos del proyecto. Además, se creyó conveniente la unión de ambas partes entendiendo que, una alerta también puede ser un mensaje de estado del resultado de una serie de pruebas o *tests*.

Se definió una alerta como un mensaje estructurado en dos partes: el nivel o importancia y la descripción. La primera indica la relevancia de la segunda (véase la Tabla 7). Además, cada nivel tiene asociado un color para aumentar la visibilidad de las alertas.

Tabla 7. Niveles de una alerta.

Color	Nivel	Significado
	Ok	Todo correcto.
	Warning	Algo puede estar mal, comprobar el cuerpo de la alerta.
	Error	Algo no salió como se esperaba.
	Ack	Reconocimiento, por ejemplo, la conexión con el servidor.

4.2.1 *Service_Alerts*

El código de *Service_Alerts* actúa como intermediario entre el MC y el servidor del sistema de alertas, teniendo así, acceso a toda la información relevante durante una campaña. Por ello, es aquí donde reside la programación necesaria para

realizar el control de errores mediante *tests*. Tal y como se observa en la Figura 14, Figura 15 y Figura 16 del apartado 3.2.5 Módulo Controlador (MC), al ser un servicio manejado por el módulo controlador se tiene acceso a las variables que almacenan temporalmente información relacionada con las capturas BLE, meteorología, nombre de la base de datos, entre otros.

El principal y más básico tipo de alertas son las que indican el punto de inicio y fin de una campaña, así como, las que notifican cuando acaba la recolecta de datos en un punto u orientación.

Dado el inconveniente mencionado al inicio del apartado 4. Desarrollo, algunas de las funcionalidades de la plataforma sometidas a control de errores son:

- *CheckWeatherDataJson*
- *CheckNumberPacketsBLE*
- *CheckIfAllMacsDetected*

Todo el programa de este apartado fue desarrollado en Python con la ayuda de las librerías:

- *Socket*: establecer comunicaciones con el protocolo UDP [44].
- *Msgpack*: serialización en formato binario [45].
- *Sqlite3*: realizar consultas SQL a una base de datos [46].
- *Configparser*: lectura de archivos de configuración [47].

4.2.1.1 *CheckWeatherDataJson*

Al solicitar información meteorológica al módulo MM, en caso de no resolverse la petición correctamente no se hace nada al respecto, simplemente se imprime en consola un mensaje indicando que algo salió mal. De esta manera, se pretende como primera iteración comprobar que se ha recibido la información correctamente y que ningún campo del objeto *WeatherDataJson* esté vacío o sea nulo.

```
def checkWeatherDataJson(self, weatherData):
    level = "Ok"
    emptyFields = 0
    for key in weatherData:
        if weatherData[key] == 0 or weatherData[key] == '':
            emptyFields = emptyFields + 1
    if emptyFields == len(weatherData):
        level = "Error"
    elif emptyFields > 0 and emptyFields < len(weatherData):
        level = "Warning"
    alert = (level, "Weather data --> "+str(weatherData), str(datetime.datetime.now().strftime("%X - %d/%m/%Y")))
    self.relayMsgToServer(alert)
```

Figura 29. Código relacionado con el test *CheckWeatherDataJson*.

El MC será el encargado de una vez recibida una respuesta del MM, pasar el resultado al Módulo de Alertas para realizar el test y notificar al usuario del estado

de este proceso. En la Figura 16, se muestra cómo se invoca el método de control de errores *CheckWeatherDataJson* que se aprecia en la siguiente figura, Figura 29.

4.2.1.2 *CheckNumberPacketsBLE*

Antes del desarrollo de este trabajo no se realizaba ningún tipo de prueba o control sobre el número de paquetes BLE obtenidos, esto es, objetos personalizados que recogen datos sobre las señales recibidas de las balizas BLE. La única manera de comprobarlo era una vez finalizada la campaña, lo que en ocasiones daba lugar a errores o campañas incompletas.

A continuación, Figura 30, se observa un fragmento de código que describe el funcionamiento del *test* que estima teóricamente el número de muestras esperadas de las balizas BLE y, compara el resultado con el número de muestras obtenidas real.

```
def checkNumberPacketsBLE(self, BLEdata):
    # Theoretical number of samples per beacon
    # N = (sampleTime(ms)/frequency(ms)) * channels
    st = int(self.readConfig("SampleTime"))
    f = int(self.readConfig("Frequency"))
    n = int(self.readConfig("NumBeacons"))
    c = int(self.readConfig("NumChannels"))
    theoreticalN = int(((st/f) * c) * n)
    samples = len(BLEdata)
    diff = abs(theoreticalN - samples)
    level = "Ok"
    if samples <= 0.5*theoreticalN:
        level = "Error"
    elif samples < 0.8*theoreticalN:
        level = "Warning"
    alert = (level, "Received "+str(samples)+"/"+str(theoreticalN)+" BLE packets",
            str(datetime.datetime.now().strftime("%X - %d/%m/%Y")))
    self.relayMsgToServer(alert)
```

Figura 30. Código relacionado con el test *CheckNumberPacketsBLE*.

4.2.1.3 *CheckIfAllMacsDetected*

Este test está ligado con el anterior, ambos dan información muy relevante para conocer si se está realizando una campaña de manera correcta y si se detectan todas las balizas (según corresponda, puede darse la situación en que una de ellas se encuentre a una distancia a la que es normal no recibir señales de esta).

Para poder realizar esta prueba hay que esperar a que se haya realizado una captura y se haya almacenada la información correspondiente en las tablas *Capture*

y *Beacon_BLE_Signal* de la base de datos. Esto se debe a que, en una captura se pueden recibir 0, 1 o múltiples señales BLE y la forma de relacionarlas es a través de sus identificadores únicos de las entradas en las base de datos. Esto se puede ver en la Figura 31.

```
def checkIfAllMacsDetected(self, nombreBD, fechaCaptura):
    macs = self.readConfig("Macs").split(',')
    bd = "C:/IPS/BBDD/"+nombreBD+".sqlite3"
    conn = sqlite3.connect(bd)
    cur = conn.cursor()
    id_captura = cur.execute("SELECT Id FROM Capture WHERE Date='"+str(fechaCaptura)+"").fetchone()[0]
    res = cur.execute("SELECT DISTINCT Mac FROM Beacon_BLE_Signal WHERE Id_captura="+str(id_captura)).fetchall()
    conn.close()
    macsDetected = []
    missing = []
    for mac in res:
        macsDetected.append(mac[0])
    for mac in macs:
        if mac not in macsDetected:
            missing.append(mac)
    level = "Ok"
    if len(missing) >= (len(macs)*0.5):
        level = "Error"
    elif len(missing) > (len(macs)*0.2):
        level = "Warning"
    alert = (level, "Detected "+str(len(macsDetected))+ " macs. Missing: "+str(missing), str(datetime.datetime.now().strftime("%X - %d/%m/%Y")))
    self.relayMsgToServer(alert)
```

Figura 31. Código relacionado con el test *CheckIfAllMacsDetected*.

4.2.2 Servidor_MA

El código del Servidor_MA se encarga de manejar las peticiones de solicitud de lectura y de almacenado de las alertas. Como se observa en la Figura 32, estas se guardan en la base de datos de la campaña y también, se plasman en un *log* o fichero de texto; eso sí, sin alterar el formato de la alerta.

```
def logAlert(level, data, timestamp):
    logFile = readConfig("LogFile")
    log = open(logFile, 'a', encoding='utf-8')
    log.write(level+": "+data+", "+timestamp+"\n")
    log.close()

def insertAlert(level, data, timestamp):
    conn = sqlite3.connect(nombreBD)
    cur = conn.cursor()
    sql = "INSERT INTO Alert (Level, Body, Timestamp) VALUES (?, ?, ?)"
    values = (level, data, timestamp)
    cur.execute(sql, values)
    conn.commit()
    conn.close()
```

Figura 32. Código relacionado con el almacenamiento de las alertas.

Al igual que el apartado anterior, el código implementado fue en Python y con las mismas librerías, salvo por *threading*, que permite la ejecución del programa a partes de él, en diferentes hilos y de forma paralela [48].

4.2.3 Interfaz

La última parte por desarrollar del sistema de alertas era la integración con algún sistema de visualización de estas que no fuera una consola o terminal. Como se buscaba una alternativa que fuera portable y no redujera o limitase los recursos del ordenador principal, se optó por elaborar una interfaz gráfica en Python también, con la ayuda de la librería PySimpleGUI [49]. Luego, para conseguir la portabilidad de la interfaz, se utilizó también la librería pyinstaller [50].

ROBOMAP Inspector es la interfaz gráfica o GUI (del inglés *Graphical User Interface*) que se muestra en la Figura 33. Como se puede ver, es bastante sencilla, comprendida por dos campos de textos, uno que muestra las alertas en tiempo real y, el otro, el historial de alertas de una campaña. Para obtener dichos registros, hay un botón que permite solicitarlos al 4.2.2 Servidor_MA (“Get History”).

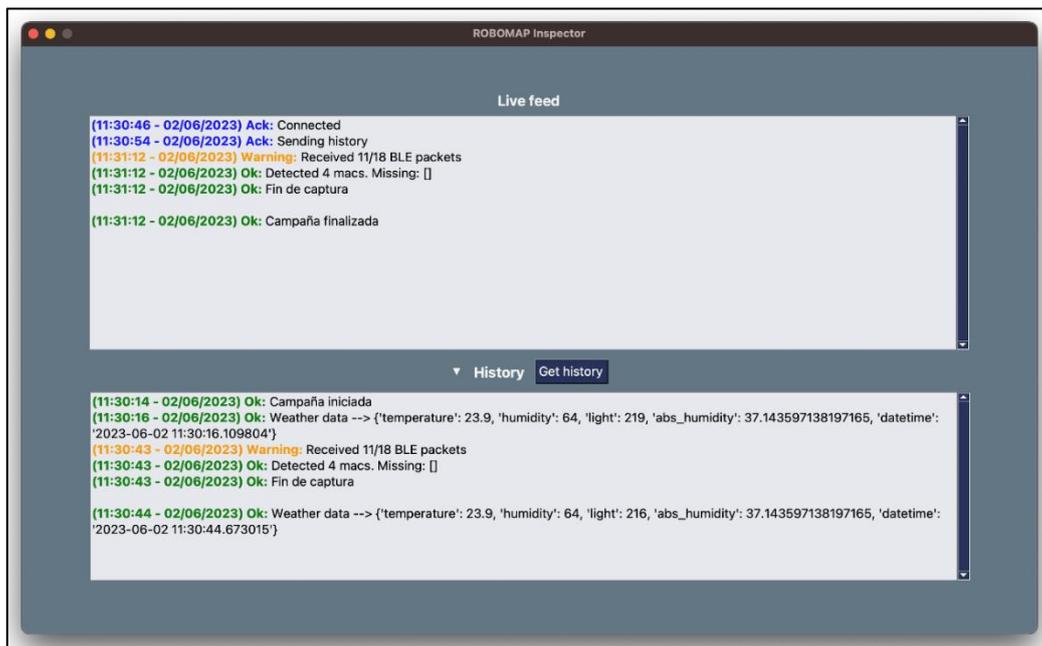


Figura 33. Interfaz gráfica del sistema de alertas, ROBOMAP Inspector.

4.3 Postproceso de datos

El procesamiento de los datos recolectados durante una campaña ha sufrido varios cambios desde el inicio del proyecto de investigación. Inicialmente, se realizaba con

hojas de Excel a mano, luego, se usó la herramienta Matlab hasta que recientemente, se desarrollaron una serie de programas que extraen la información de las bases de datos y la dejan con el formato necesario para su posterior uso en los métodos de posicionamiento.

4.3.1 Datos empleados

Como el objetivo de este tercer apartado consistía en implementar un nuevo método de estimación de la posición, se utilizó la misma base de datos la empleada en un artículo publicado por los tutores de este trabajo, Gabriele Salvatore de Blasio y Alexis Quesada Arencibia [51]. De esta manera, se podría no solo corroborar los resultados obtenidos en dicho artículo, sino también, compararlos con los nuevos métodos.

Por aquel entonces no existía ROBOMAP, lo que suponía realizar las capturas de señales manualmente. Para ello, se utilizó como zona de prueba un patio exterior del IUCES con la forma de la Figura 34.

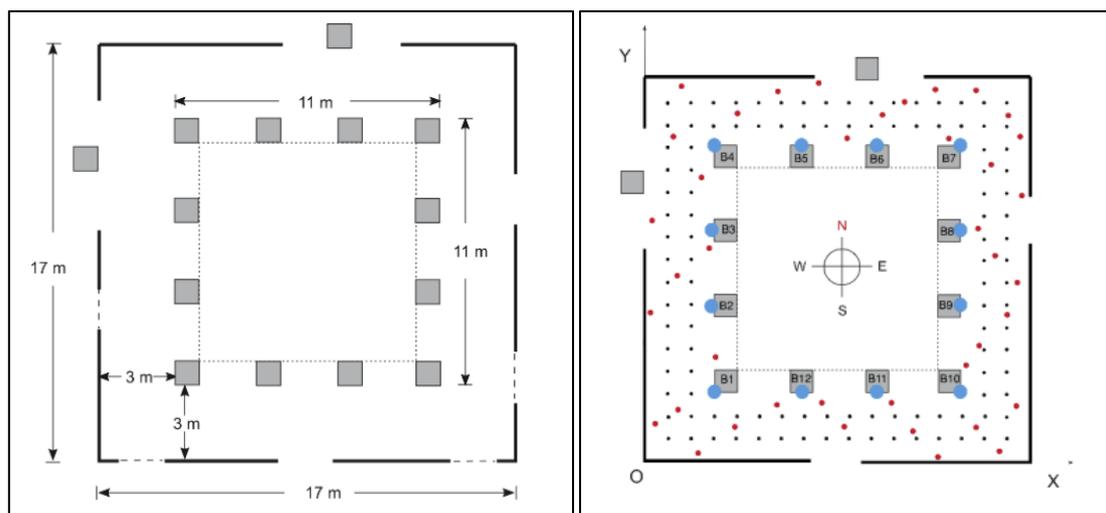


Figura 34. Mapas zona de prueba con puntos marcados [51].

Como se puede apreciar en la Figura 34, se optó por trabajar con una rejilla o matriz de 112 puntos de referencia, midiendo cada uno 1x1 metro, lo que resultó en el despliegue de 12 balizas BLE a una altura de 2,1 metros, llegando a cubrir cada una alrededor de 14 m². Luego, la frecuencia y la potencia de emisión de señales se estableció en 500 ms y 0 dBm (decibelio-milivatios) respectivamente.

Por otro lado, para cada punto de referencia y aleatorio, se capturaron señales en orientaciones según los cuatro puntos cardinales (Norte, Sur, Este y Oeste). Para la creación de la base de datos, dado un punto de la zona de prueba, se obtenía para cada uno de los beacons la señal máxima (como los valores de las RSSI son números

enteros negativos, el mayor valor es aquel más cercano a 0) en cada orientación y, finalmente, se calculaba la media de los cuatro máximos obtenidos (véase Figura 35).

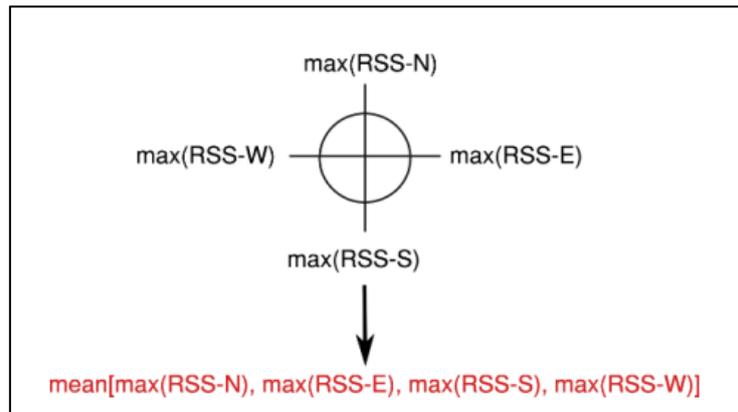


Figura 35. Expresión para obtener la RSSI de cada baliza en un punto [51].

4.3.2 WKNN

WKNN (en inglés *Weighted k-Nearest Neighbor*) es uno de los algoritmos más usados como método de posicionamiento por su fácil comprensión e implementación. Resulta en una variante mejorada de NN (en inglés *Nearest Neighbor*) cuya principal diferencia radica en la asignación de pesos a los vecinos en función a la distancia a la que se encuentren del objetivo, esto es, cuánto más cerca, mayor influencia tendrá de cara al resultado frente a otros más lejanos [52].

Antes de implementar este algoritmo, hay que definir el tipo de distancia que se empleará y el valor de los pesos a asignar a cada vecino. Siguiendo la línea del artículo del apartado anterior, se aplicará la distancia euclídea y como peso el inverso de la distancia (véase Figura 36). Para definir el número de vecinos K, se suele hacer pruebas a ver con qué valor se obtienen mejores resultados.

$$d(PA, PR) = \sqrt{\underbrace{(RSSI_{PR} - RSSI_{PA})^2}_{\text{Beacon 1}} + \underbrace{(RSSI_{PR} - RSSI_{PA})^2}_{\text{Beacon 2}} + \dots}_{\text{n}}$$

Figura 36. Expresión distancia euclídea y peso.

Se computa la distancia entre las RSSI obtenidas en un punto aleatorio y las correspondientes a todos los puntos de referencia. A continuación, se calcula la suma ponderada de las distancias por sus pesos y se divide cada distancia por el resultado.

De esta manera, según el número de vecinos K , si es 1 se asigna la posición del vecino más cercano (menor distancia); si es mayor o igual que 2, se hace una media de las coordenadas de los vecinos más cercanos.

Para poner a prueba WKNN, se desarrolló un programa en Python con ayuda de las librerías Pandas, Numpy, Matplotlib y Sklearn [53] [54] [55] [56] [57]. Este, consistía en:

- Lee las RSSI obtenidas para los puntos de referencia y los puntos aleatorios.
- Calcula los resultados con diferentes valores de K manteniendo constante los parámetros de la distancia euclídea y el inverso de la distancia como peso.
- Genera dos gráficas a partir del modelo con mejores valores de precisión y exactitud.

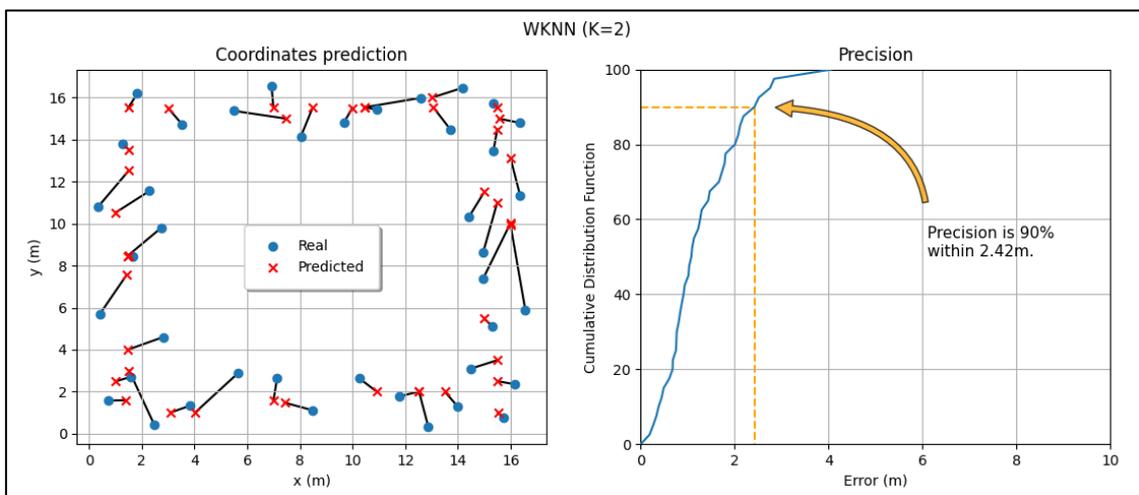


Figura 37. Gráficas resultados WKNN.

Los resultados obtenidos con WKNN en la elaboración de este proyecto fueron próximos a los de 4.3.1 Datos empleados. En ese trabajo de investigación se consiguió un error promedio de 1.47-2.15m y una precisión de 1.81-3.58m el 90% del tiempo. Por otro lado, con el número de vecinos $K=2$, se obtuvo un error promedio de 1.3 m y, el 90% de las veces una precisión de 2.4 m. En la Figura 37, se aprecian dos gráficas, una que muestra visualmente la diferencia entre las coordenadas reales – predichas y, la otra, la precisión alcanzada en las pruebas de validación del método (estimación posición puntos aleatorios).

4.3.3 Propuesta método de posicionamiento

Tras mucha investigación sobre métodos de posicionamiento en relación con sistemas de localización en interiores, y buscar información sobre diferentes modelos y algoritmos de *Machine Learning* actuales, se optó, dada su reciente popularidad y eficacia, por Máquinas Vector Soporte o SVM (en inglés *Support Vector Machines*).

Las SVM son una serie de métodos de aprendizaje supervisado usados para problemas de regresión y clasificación. Su objetivo es el de encontrar una línea o hiperplano que separe lo mejor posible el conjunto de datos. Para ello, se busca lograr definir un plano con margen máximo, esto es, que tenga la máxima distancia entre ambas instancias de los datos. Asimismo, esto último se consigue al utilizar ciertos valores del conjunto de datos como vectores de soporte (de ahí el nombre del método), influyendo en la dirección en la que se define el plano. En la Figura 38, se demuestra lo antes explicado gráficamente [58] [59] [60] [61] [62].

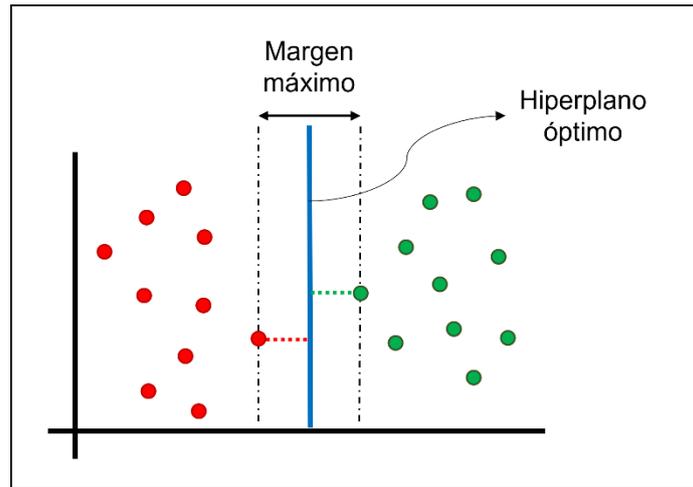


Figura 38. Ejemplo de hiperplano óptimo.

Para los casos en los que se trabaja con conjuntos de datos linealmente no separables, se emplea lo que se denomina, el *kernel trick* [63]. Con ello, se consigue trabajar en un espacio de mayores dimensiones sin tener que realizar las transformaciones asociadas, aplicando unas expresiones matemáticas que permiten representar los datos por las coordenadas “simuladas”, donde sí es posible definir un hiperplano que separe las clases del conjunto de datos.

Al principio, aunque lo que se buscaba era obtener las coordenadas predichas de un punto aleatorio, como el resultado de los métodos consistía en un único valor, se intentó replicar en cierto modo, el funcionamiento del vecino más cercano o NN. Para ello, se implementó una variante del SVM, el SVC (en inglés *Support Vector Classification*) [64]. Todas las señales recibidas en un punto de referencia se consideraba una clase, de tal manera que el resultado del SVC era el número del punto con RSSI de valores similares. Sin embargo, a pesar de probar diferentes combinaciones en la configuración del modelo, es decir, sus parámetros y variantes, los resultados de los porcentajes de acierto en la clasificación no superaban el 40% en ningún caso.

Finalmente, se encontró un artículo en Internet que describe una forma para obtener múltiples salidas con un modelo de regresión que por defecto no lo permite. En él se comentan dos posibilidades que difieren en cómo se relacionan las salidas, es decir, si los resultados son independientes entre sí o no (*MultioutputRegressor* y *RegressorChain*, respectivamente) [65] [66] [67]. Así, se pasó a trabajar con la

variación de regresión de SVM, el SVR (en inglés *Support Vector Regressor*) [68]. La idea detrás de las opciones antes mencionadas es que funcionan como un envoltorio o *wrapper*, esto es, se invoca el modelo como si fuera un parámetro de las funciones multi-salida.

No obstante, como implementar el modelo de regresión no fue tan complicado gracias al uso de librerías, se decidió desarrollar las variantes del SVR para tener más puntos de comparación de los resultados obtenidos (LinearSVR y NuSVR) [69] [70]. Eso sí, como todos los modelos permitían la adición y modificación de parámetros, se hizo un programa para cada uno de ellos que, probaba diferentes combinaciones de estos, almacenando los resultados en archivos Excel y, por último, graficaba el modelo con mejor precisión y exactitud.

La Tabla 8, muestra los resultados obtenidos, además de los del WKNN, por los diferentes modelos y con su mejor configuración de parámetros característicos.

Tabla 8. Resultados métodos de posicionamiento.

Modelo	Exactitud (m)	Precisión (m)
SVR MultioutputRegressor (rbf_C5_G0.05)	1.89	3.37
SVR RegressorChained (rbf_C5_G0.05)	1.84	3.17
LinearSVR MultioutputRegressor (C0.5_I500)	2.26	3.78
LinearSVR RegressorChained (C8_I500)	2.09	3.07
NuSVR MultioutputRegressor (Nu0.5_rbf_C5_G0.05)	1.84	3.31
NuSVR RegressorChained (Nu0.6_rbf_C5_G0.05)	1.81	3.19

La Figura 39 y Figura 40, muestran una gráfica con la precisión obtenida de todos los diferentes métodos probados. Como se puede ver, el algoritmo o modelo con mayor precisión es el WKNN, seguido de LinearSVR con *RegressorChain*, mientras que, por el contrario, el peor fue LinearSVR con *MultioutputRegressor*.

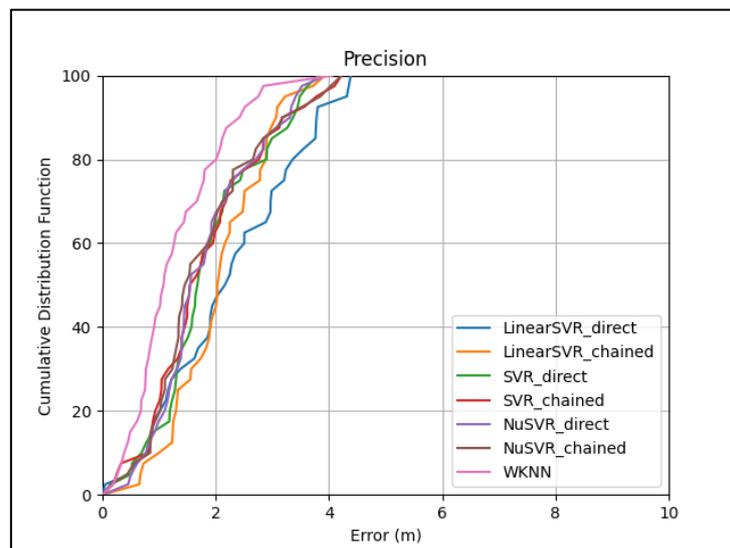


Figura 39. Resultados métodos de posicionamiento.

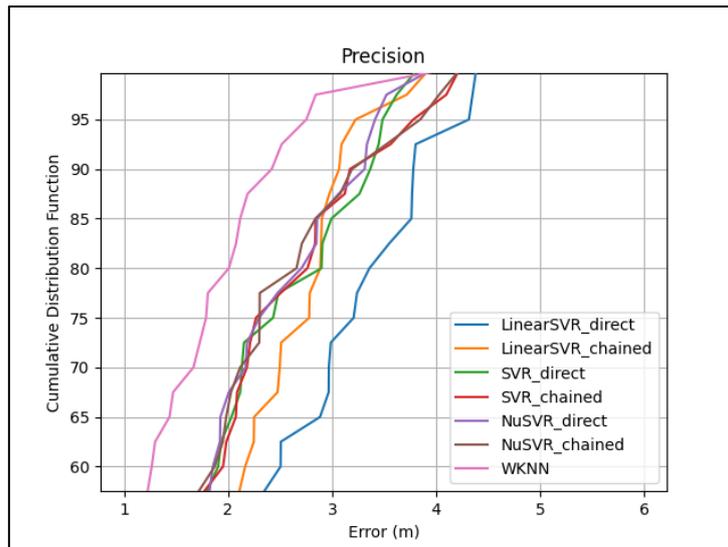


Figura 40. Resultados con la zona del 60-100% aumentada.

5. Conclusiones y trabajo futuro

Una vez analizados los objetivos propuestos para este trabajo y los resultados alcanzados, se puede decir que se han cumplido todos.

En cuanto al consumo energético de la plataforma, se realizó la medición con el enchufe monitor de consumo y se pudo comprobar, aunque fue solo para las partes del módulo ROBOMAP, que, con la pinza amperimétrica se obtenían mediciones iguales sino similares. Por tanto, ya se sabe que, si por ejemplo se quiere mantener en funcionamiento a la plataforma durante tres horas, las características del paquete de baterías deberán ser:

- Al menos 12V.
- Capacidad de al menos 10.000 mAh.
- Al menos 3A de descarga máxima.

Se consiguió diseñar e implementar la base del sistema de alertas y control de errores durante una campaña de recogida de datos, permitiendo así tener acceso a toda la información relevante y que puede ser sometida a *tests* para mejorar el rendimiento y resultados de ROBOMAP. Asimismo, se desarrolló una simple interfaz portable que permite visualizar las alertas tanto en estado real como las generadas y almacenadas previamente.

Finalmente, se compararon los resultados obtenidos del WKNN y los nuevos algoritmos implementados en la estimación de la posición. Se probaron diferentes versiones de las máquinas vector soporte o SVM, como la de clasificación o regresión, alcanzando un factor de acierto del 40% y 93% respectivamente. Luego, a partir de los diferentes modelos de regresión basados en SVM, se cotejaron los resultados que, aunque no fueron mejores que el método actual, el WKNN, se obtuvieron resultados bastante cercanos que se considera tenerlos en cuenta para futuras pruebas en bases de datos diferentes.

- WKNN: Exactitud 1.31m y Precisión 2.42m.
- NuSVR (mejor modelo basado en SVM): Exactitud 1.81m y Precisión 3.19m.

5.1 Conclusión personal

El desarrollo de este trabajo ha sido sin duda alguna, un reto, pero no solo por las fallas inoportunas, sino por la inmensa cantidad de información, de diferentes áreas de conocimientos, que hay saber y controlar para poder avanzar en esta línea de investigación. Y es que, al ser los IPS una tecnología reciente, no existe una única solución, y por ende, resulta en investigar y probar un poco en lo desconocido. Por ejemplo, la creación de una plataforma robótica no es algo trivial, mucho menos para gente que no es experta en electrónica y la física que involucra.

No obstante, ha sido un proceso tremendamente interesante y gratificante. El compartir ideas, hacer prototipos y realizar pruebas junto con otros profesionales de otros campos, ha sido una experiencia enriquecedora para mi persona.

Importante mencionar, que el trabajo descrito en este documento no hubiera sido posible de no ser por el tiempo y la dedicación previa de estudiantes y miembros del IUCES y el grupo CIPERBIG.

Por otro lado, es agradable saber que gracias a los conocimientos adquiridos durante mis estudios, he conseguido tener una base sólida que me permitió trabajar sin ninguna dificultad todos los apartados de este trabajo. De hecho, la información plasmada en este TFG e investigada durante su desarrollo, no hubiera sido posible de no haber dedicado tiempo a entender los conceptos e ideas que se enseñan en esta carrera.

5.2 Trabajo futuro

En lo que se refiere a trabajo futuro, aún quedan muchos aspectos a mejorar y funcionalidades por añadir, tanto para la plataforma robótica ROBOMAP como para el proyecto en la línea de investigación de los sistemas de posicionamiento en interiores. Algunos son:

- Para la realización de este TFG se realizó bastante investigación previa sobre ROBOMAP, sus componentes y puesta en marcha para lograr trabajar en sus nuevas funcionalidades. Por ello, se desarrolló el manual de ROBOMAP en el que se documentó todo lo que se descubrió y probó en su momento. Hasta el momento se requiere de algún conocimiento de programación para poder realizar una campaña de recogida de datos. Es por esto por lo que es quizás el aspecto más importante para tener en cuenta para el futuro. La idea sería diseñar algún método que unifique todos los procesos que actualmente se inician y controlan de manera separada.
- Con el estudio realizado de la arquitectura *hardware* y, conocer las potencias y la energía estimada que consume ROBOMAP, toca elaborar e incorporar una batería capaz de dotar de autonomía a la plataforma.
- Queda desarrollada lo que consolida la base de un sistema de alertas y control de errores durante una campaña de recogida de datos con ROBOMAP. Ahora, faltaría definir e implementar más tipos de alertas o tests, para así, mejorar la fiabilidad de los resultados obtenidos. Asimismo, la interfaz inspectora de alertas es el primer prototipo, y por tanto, está sujeto a cambios y mejoras. Por ejemplo, permitir el control remoto de la campaña desde la interfaz o, elaborar una web que sea accesible desde fuera de la red local de la campaña.
- Los nuevos métodos de posicionamiento, aunque resultaron no ser mejores en la base de datos empleada en este trabajo, aún pueden ser utilizados como puntos de comparación o incluso, punto de partida para el desarrollo de un método híbrido.

6. Referencias

- [1] Y. Bhattacharjee, «National Geographic,» 28 Enero 2019. [En línea]. Available: <https://www.nationalgeographic.es/historia/2019/01/los-smartphones-revolucionan-nuestras-vidas-pero-a-que-precio>. [Último acceso: Mayo 2023].
- [2] J. F. Otero, «El Economista,» 24 Octubre 2019. [En línea]. Available: <https://www.economista.com.mx/opinion/Impacto-de-la-innovacion-tecnologica-20191024-0019.html>. [Último acceso: Mayo 2023].
- [3] C. Otto, «La Vanguardia,» 26 Febrero 2017. [En línea]. Available: <https://www.lavanguardia.com/tecnologia/20170226/42274940927/diez-anos-smartphones-cambiado-vida.html>. [Último acceso: Mayo 2023].
- [4] «Wikipedia,» [En línea]. Available: <https://es.wikipedia.org/wiki/GPS>. [Último acceso: Mayo 2023].
- [5] «GPS.gov,» [En línea]. Available: <https://www.gps.gov/systems/gps/spanish.php>. [Último acceso: Mayo 2023].
- [6] A. Witze, «Scientific American,» 30 Octubre 2019. [En línea]. Available: <https://www.scientificamerican.com/article/gps-is-doing-more-than-you-thought/>. [Último acceso: Mayo 2023].
- [7] «ULPGC,» [En línea]. Available: http://www2.ulpgc.es/archivos/plan_estudios/4008_40/ObjetivosyCompetenciasdelGII.pdf. [Último acceso: Mayo 2023].
- [8] E. d. I. Informática, «EII ULPGC,» Enero 2023. [En línea]. [Último acceso: Mayo 2023].
- [9] «Innovation News Network,» 3 Octubre 2018. [En línea]. Available: <https://www.innovationnewsnetwork.com/gps-indoor-navigation/468/>. [Último acceso: Mayo 2023].
- [10] R. S. Campos y L. Lovisolo, de *RF POSITIONING Fundamentals, Applications and Tools*, Artech House, 2015.
- [11] «Kontakt.io,» [En línea]. Available: <https://kontakt.io/what-is-a-beacon/>. [Último acceso: Mayo 2023].
- [12] «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Precisi%C3%B3n_y_exactitud. [Último acceso: Mayo 2023].
- [13] G. S. de Blasio, A. Quesada-Arencibia, C. R. García y J. C. Rodríguez-Rodríguez, «Bluetooth Low Energy Technology Applied to Indoor Positioning Systems: An Overview,» p. 8, 2020.

- [14] Situm, «Situm Indoor Positioning,» 24 Febrero 2022. [En línea]. Available: <https://situm.com/es/blog/posicionamiento-en-interiores/sistemas-de-localizacion-en-interiores/>. [Último acceso: Mayo 2023].
- [15] R. F. Brena, J. P. García-Vázquez, C. E. Galván-Tejada, D. Muñoz-Rodríguez, C. Vargas-Rosales y J. Fangmeyer, «Evolution of Indoor Positioning Technologies: A Survey,» *Journal of Sensors*, vol. vol. 2017, n° 2630413, p. 21, 2017.
- [16] Pointr, «Pointr.tech,» 5 Mayo 2023. [En línea]. Available: <https://www.pointr.tech/blog/location-fingerprinting-what-is-it-should-you-choose-it>. [Último acceso: Mayo 2023].
- [17] A. R. Sánchez, *TFG - Desarrollo de módulo de comunicación en Blender para la interoperación con plataforma robótica*, 2020.
- [18] «Blender,» [En línea]. Available: <https://www.blender.org/>. [Último acceso: Mayo 2023].
- [19] A. R. Sánchez, «Github,» Junio 2021. [En línea]. Available: <https://github.com/bertoramos/blender-editor>. [Último acceso: Mayo 2023].
- [20] D. S. Suárez, *TFM - Desarrollo de una estación meteorológica y un sistema orientador para una plataforma robótica de toma de datos en un Sistema de Posicionamiento en Interiores (IPS)*, 2020.
- [21] I. N. N. Alejo, *TFG - Desarrollo de módulo controlador e interconexión con módulo de adquisición de datos BLE y módulo orientador de receptor BLE: Integración con una plataforma robótica de toma de datos BLE*, 2022.
- [22] Y. B. E. Maachi, *TFG - Desarrollo de módulo controlador e interconexión con módulos de adquisición de datos BLE y meteorológicos: Integración con plataforma robótica de toma de datos Bluetooth Low Energy*, 2022.
- [23] R. A. Española, «Diccionario de la lengua española,» 2022. [En línea]. Available: <https://dle.rae.es/electricidad?m=form>. [Último acceso: Mayo 2023].
- [24] E. U. d. Oficios, «UNLP,» [En línea]. Available: <https://yold.unlp.edu.ar/frontend/media/86/27586/9f6fb6121050cf6e1acd93f79c73eb6d.pdf>. [Último acceso: Mayo 2023].
- [25] Iberdrola, «Iberdrola,» 12 Mayo 2021. [En línea]. Available: <https://www.iberdrola.es/blog/luz/que-es-voltaje-electrico-como-medirlo>. [Último acceso: Mayo 2023].
- [26] W. McAllister, «Khan Academy,» [En línea]. Available: <https://www.khanacademy.org/science/physics/circuits-topic/circuits-resistance/a/ee-voltage-and-current>. [Último acceso: Mayo 2023].
- [27] «Ferrovial,» [En línea]. Available: <https://www.ferrovial.com/en/stem/electric-current/>. [Último acceso: Mayo 2023].

- [28] Endesa, «Fundación Endesa,» [En línea]. Available: <https://www.fundacionendesa.org/es/educacion/endesa-educa/recursos/que-es-la-energia>. [Último acceso: Mayo 2023].
- [29] «Repsol,» [En línea]. Available: <https://www.repsol.com/es/energia-futuro/transicion-energetica/energia-solar/index.cshtml>. [Último acceso: Mayo 2023].
- [30] «Mi Universo Electrónico,» [En línea]. Available: <https://miuniversoelectronico.com/corriente-alterna/potencia-en-corriente-alterna/>. [Último acceso: Mayo 2023].
- [31] «Area Tecnología,» [En línea]. Available: <https://www.areatecnologia.com/electricidad/potencia-electrica.html>. [Último acceso: Mayo 2023].
- [32] «Leroy Merlin,» [En línea]. Available: <https://www.leroymerlin.es/productos/electricidad-y-domotica/multimetros-detectores-de-voltaje-y-pinzas-amperimetricas/>. [Último acceso: Mayo 2023].
- [33] «Amazon,» [En línea]. Available: <https://www.amazon.es/APC-Surge-Arrest-Home-Office/dp/B00T7BZFFS>. [Último acceso: Mayo 2023].
- [34] «Flaticon,» [En línea]. Available: <https://www.flaticon.es/>. [Último acceso: Mayo 2023].
- [35] «Learning About Electronics,» [En línea]. Available: <http://www.learningaboutelectronics.com/Articulos/Voltaje-RMS.php>. [Último acceso: Mayo 2023].
- [36] Solectro, «Solectroshop,» 19 Febrero 2021. [En línea]. Available: <https://solectroshop.com/es/blog/x-n71>. [Último acceso: Mayo 2023].
- [37] «Equipos y Laboratorio,» [En línea]. Available: [https://www.equiposylaboratorio.com/portal/articulo-ampliado/baterias-\(electricidad\)](https://www.equiposylaboratorio.com/portal/articulo-ampliado/baterias-(electricidad)). [Último acceso: Mayo 2023].
- [38] A. Sharma, «Tech Counsellor,» [En línea]. Available: <https://www.techcounsellor.com/2017/08/selecting-proper-battery-engineering-projects/>. [Último acceso: Mayo 2023].
- [39] H. Shops, «Todo Sai,» 24 Noviembre 2016. [En línea]. Available: <https://todosai.com/blog/baterias-tipos-y-usos-b50.html>. [Último acceso: Mayo 2023].
- [40] J. Insa, «Monsolar,» [En línea]. Available: <https://www.monsolar.com/blog/baterias-de-ion-litio-ventajas-e-inconvenientes/>. [Último acceso: Mayo 2023].
- [41] «HDI Battery,» [En línea]. Available: https://www.hdibattery.com/p11807070_bateria-litio-ion-lg-inr18650-mj1-3-7v-3500mah-10a.html. [Último acceso: Mayo 2023].
- [42] «Synopsys,» [En línea]. Available: <https://www.synopsys.com/glossary/what-is-a-battery-management-system.html>. [Último acceso: Mayo 2023].

- [43] «VRUZEND DIY,» [En línea]. Available: <https://vruzend.com/product-category/battery-kits/>. [Último acceso: Mayo 2023].
- [44] «Python.org - Socket,» [En línea]. Available: <https://docs.python.org/3/library/socket.html>. [Último acceso: Mayo 2023].
- [45] «Message Pack,» [En línea]. Available: <https://msgpack.org/>. [Último acceso: Mayo 2023].
- [46] «Python.org - Sqlite3,» [En línea]. Available: <https://docs.python.org/3/library/sqlite3.html>. [Último acceso: Mayo 2023].
- [47] «Python.org - Configparser,» [En línea]. Available: <https://docs.python.org/3/library/configparser.html>. [Último acceso: Mayo 2023].
- [48] «Python.org - Threading,» [En línea]. Available: <https://docs.python.org/3/library/threading.html>. [Último acceso: Mayo 2023].
- [49] «PySimpleGUI,» [En línea]. Available: <https://www.pysimplegui.org/en/latest/>. [Último acceso: Mayo 2023].
- [50] «PyInstaller,» [En línea]. Available: <https://pyinstaller.org/en/stable/>. [Último acceso: Mayo 2023].
- [51] G. De Blasio, A. Quesada-Arencibia, C. R. García, J. C. Rodríguez- Rodríguez y R. Moreno-Díaz, «IEEE Access,» *A Protocol-Channel-Based Indoor Positioning Performance Study for Bluetooth Low Energy*, vol. 6, n° 10.1109/ACCESS.2018.2837497, p. 12, 2018.
- [52] J. McCaffrey, «Visual Studio Magazine,» 04 Octubre 2019. [En línea]. Available: <https://visualstudiomagazine.com/articles/2019/04/01/weighted-k-nn-classification.aspx>. [Último acceso: Mayo 2023].
- [53] «Pandas,» [En línea]. Available: <https://pandas.pydata.org/>. [Último acceso: Mayo 2023].
- [54] «NumPy,» [En línea]. Available: <https://numpy.org/>. [Último acceso: Mayo 2023].
- [55] «Matplotlib,» [En línea]. Available: <https://matplotlib.org/>. [Último acceso: Mayo 2023].
- [56] «Scikit-Learn,» [En línea]. Available: <https://scikit-learn.org/stable/>. [Último acceso: Mayo 2023].
- [57] «Scikit-Learn - KNeighborsRegressor,» [En línea]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html>. [Último acceso: Mayo 2023].
- [58] «Scikit-Learn - SVM,» [En línea]. Available: <https://scikit-learn.org/stable/modules/svm.html>. [Último acceso: Mayo 2023].
- [59] R. Pupale, «Towards Data Science,» 16 Junio 2018. [En línea]. Available: <https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989>. [Último acceso: Mayo 2023].

- [60] V. Explained, «YouTube,» 2022. [En línea]. Available: https://www.youtube.com/watch?v=_YPSrckx28. [Último acceso: Mayo 2023].
- [61] R. Gandhi, «Towards Data Science,» 07 Junio 2018. [En línea]. Available: <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>. [Último acceso: Mayo 2023].
- [62] S. w. J. Starmer, «YouTube,» 2020. [En línea]. Available: <https://www.youtube.com/watch?v=efR1C6CvhmE>. [Último acceso: Mayo 2023].
- [63] D. Wilimitis, «Towards Data Science,» 12 Diciembre 2018. [En línea]. Available: <https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f>. [Último acceso: Mayo 2023].
- [64] «Scikit-Learn - SVC,» [En línea]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>. [Último acceso: Mayo 2023].
- [65] J. Brownlee, «Machine Learning Mastery,» 27 Marzo 2020. [En línea]. Available: <https://machinelearningmastery.com/multi-output-regression-models-with-python/>. [Último acceso: Mayo 2023].
- [66] «Scikit-Learn - MultiOutputRegressor,» [En línea]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.multioutput.MultiOutputRegressor.html>. [Último acceso: Mayo 2023].
- [67] «Scikit-Learn - RegressorChain,» [En línea]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.multioutput.RegressorChain.html>. [Último acceso: Mayo 2023].
- [68] «Scikit-Learn - SVR,» [En línea]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html>. [Último acceso: Mayo 2023].
- [69] «Scikit-Learn - LinearSVR,» [En línea]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVR.html>. [Último acceso: Mayo 2023].
- [70] «Scikit-Learn - NuSVR,» [En línea]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.NuSVR.html>. [Último acceso: Mayo 2023].

Anexo 1. Manual ROBOMAP

MANUAL DE USUARIO ROBOMAP

ROBOMAP es una plataforma robótica desarrollada por el Instituto Universitario de Cibernética, Empresa y Sociedad (IUCES) en la línea de investigación de Sistemas de Posicionamiento en Interiores o **IPS** (del inglés *Indoors Positioning System*).



Figura 41. Plataforma robótica ROBOMAP

La metodología actual que utiliza el IUCES en esta línea de investigación es la que se conoce como Análisis de escena o **Fingerprinting**. Se basa en la captura y estudio de la Intensidad de Señal Recibida o **RSS** (del inglés *Received Signal Strength*) de diferentes emisores de señal en dos situaciones diferentes, la fase de calibrado y la fase de posicionamiento. La primera, también conocida como fase *offline*, consiste en un estudio previo del entorno en el que se creará una base de datos con la que contrastar la información obtenida durante la segunda, también llamada fase *online*.

El proceso de captura de datos durante el análisis del entorno puede llegar a consumir bastante tiempo y, además, se deben tener en cuenta diversos factores que pueden afectar a la emisión y recepción de las señales, como son la temperatura y humedad del ambiente, la posición y orientación del dispositivo receptor, entre otros. Con la idea de **automatizar** este proceso nace ROBOMAP.

Este proyecto se ha ido trabajando por módulos a medida que diferentes miembros del IUCES junto con estudiantes, del grado de Ingeniería Informática en la Universidad de Las Palmas de Gran Canaria (ULPGC), han hecho sus aportaciones como parte de sus **Prácticas Externas** y/o **Trabajos de Fin de Carrera**. De esta manera, surge la necesidad de crear este documento, una guía en la que se describe brevemente su diseño y cómo realizar una campaña de recogida de datos con éxito.

1. DISEÑO

ROBOMAP está estructurado en diferentes módulos que dotan a la plataforma de diferentes funcionalidades. Todos ellos, se comunican entre sí utilizando la **arquitectura cliente-servidor** con el **protocolo UDP** (del inglés *User Datagram Protocol*). De manera resumida, se describen a continuación:

- **Módulo ROBOMAP (MR):** Los compañeros Alberto Ramos Sánchez y José Carlos Rodríguez dotaron a la plataforma de movimiento omnidireccional y, con el uso de balizas ultrasónicas para determinar su posición, indicarle una ruta a seguir dentro del espacio designado.
- **Módulo Meteorológico (MM):** David Suárez Suárez desarrolló, utilizando una placa Arduino y una serie de sensores, un sistema de toma de datos de la luminosidad, temperatura y humedad del ambiente en la zona de trabajo. Estos valores son importantes ya que pueden influir en la emisión y recepción de las señales Bluetooth.
- **Módulo de Adquisición de Datos BLE (MAD-BLE):** Los compañeros Yousuf Boutahar El Maachi e Iru Nervey Navarro Alejo incorporaron en la plataforma un receptor de señales provenientes de diferentes emisores o *beacons* BLE (del inglés *Bluetooth Low Energy*).
- **Módulo Orientador del Receptor BLE (MOR-BLE):** También desarrollado por David Suárez Suárez, permite poner el dispositivo receptor BLE en un soporte capaz de rotar, con la finalidad de registrar las señales recibidas desde diferentes ángulos.
- **Módulo Controlador (MC):** Yousuf e Iru, diseñaron e implementaron lo que se puede denominar el director de la plataforma. Este, durante una campaña de recogida de datos, es el encargado de comunicarse con las demás partes, indicándoles cuándo activarse, cómo operar y dónde guardar los datos almacenados a lo largo de todo el proceso.

- **Módulo de alertas (MA):** Fabián A. Beirutti Pérez, implementó una nueva funcionalidad a la plataforma, un sistema de alertas que consiste en unos mensajes que muestran al usuario el resultado de diferentes controles de errores y el estado funcional durante una campaña. Además, se diseñó una simple GUI para la inspección de las alertas, llamada ROBOMAP Inspector.

2. PREPARACIÓN PREVIA

Actualmente, se necesitan una serie de equipos y herramientas software para iniciar el proceso de captura. Antes de realizar una campaña de recogida de datos, debemos comprobar que disponemos de todo el material y que esté debidamente cargado y configurado.

2.1 Ordenador principal

El ordenador principal es el centro de operaciones de ROBOMAP, esto es, lugar donde se inicializarán y controlarán todos los procesos en una campaña de captura de datos. A este equipo se conectará el Hub USB para permitir la comunicación con el resto de los dispositivos de ROBOMAP.

Durante la redacción de esta guía, es un portátil cedido por el IUCES que contiene toda la información y código relativo a cada uno de los módulos en “C:/IPS”. Entonces, deberemos:

- Disponer de los archivos de programa y/o *scripts* correspondientes a todos los módulos (MR es en C/C++ y los demás en Python).
- Comprobar que los archivos estén ubicados dentro de las rutas especificadas de los ficheros de configuración de cada módulo (modificar las rutas si corresponde).

2.2 Ordenador auxiliar

El equipo auxiliar, ordenador portátil o no, será en el que nos apoyemos para realizar una campaña. Aquí, iniciaremos los programas de las balizas ultrasónicas y la de creación y comunicación de rutas para la plataforma. Entonces, deberemos:

- Tener instalado el programa de creación de rutas (herramienta de diseño Blender y el *add-on* desarrollado por Alberto).
- Tener instalado el software de control de las balizas ultrasónicas de Marvelmind.

2.3 Resto del material

Además de los ordenadores, principal y auxiliar, debemos comprobar que todos los módulos que conforman ROBOMAP disponen de todos sus componentes y, según corresponda, que tengan batería, estén conectados y/o configurados correctamente:



Figura 42. Baliza de ultrasonido de Marvelmind



Figura 43. Baliza BLE de Kontakt.io



Figura 44. Router CISCO

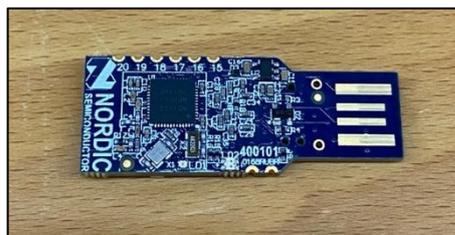


Figura 45. Dongle Bluetooth

El router de la Figura 4, configurado previamente por un compañero del IUCES, crea una red local llamada “lynksis”, a la que conectaremos los ordenadores para establecer la comunicación entre ellos. Este no hubo necesidad de modificarlo a pesar de que a veces tardada en iniciar la red.

Hay que destacar que estas fueron las herramientas utilizadas durante la redacción de este documento. Esto no implica que sea la única opción, existen alternativas para alguno de los elementos, por ejemplo, se disponía de balizas BLE de otras dos compañías diferentes.

IMPORTANTE: Desactivar el modo ahorro de batería y modo de suspensión de pantalla para evitar posibles problemas con la ejecución de código.

3. REALIZAR UNA CAMPAÑA

De manera resumida, los pasos a seguir para realizar una campaña de captura de datos son:

1. Comprobar que se detectan las balizas de ultrasonido y se reciben señales de las BLE.
2. Encender el router CISCO y conectar los equipos, principal y auxiliar, a su red.
3. Configurar y distribuir las balizas BLE y de ultrasonido.
4. Crear o cargar la escena y la ruta a seguir.
5. Iniciar servidores MAD-BLE, MOR-BLE, MM y MA.
6. Iniciar servidor ROBOMAP en equipo principal y cliente ROBOMAP en equipo auxiliar.
7. Si se quiere seguir la campaña, iniciar ROBOMAP Inspector.
8. Iniciar MC.

3.1 Balizas ultrasonido

La herramienta software de [Marvelmind](#), también conocida como Dashboard, nos permitirá monitorizar y configurar las **balizas de ultrasonido**. Toda la información necesaria para la instalación del programa la encontramos en el apartado de descargas de la web oficial. En principio, con descargar la última versión nos bastará.



Figura 46. Baliza de ultrasonido y receptor de Marvelmind

Exceptuando dos balizas que se posicionan en la parte superior de ROBOMAP, al resto, nos referiremos como estáticas, ya que las distribuiremos dentro del espacio de prueba designado y no las moveremos hasta finalizar.

El receptor va conectado por cable al ordenador auxiliar y, las dos balizas estáticas, una vez posicionadas en la plataforma, se conectan por cable al Hub USB.

Toda la información necesaria sobre el Dashboard, así como, la imagen mostrada a continuación, donde se describen las partes que conforman la interfaz del programa, lo podemos encontrar en el [manual digital](#).

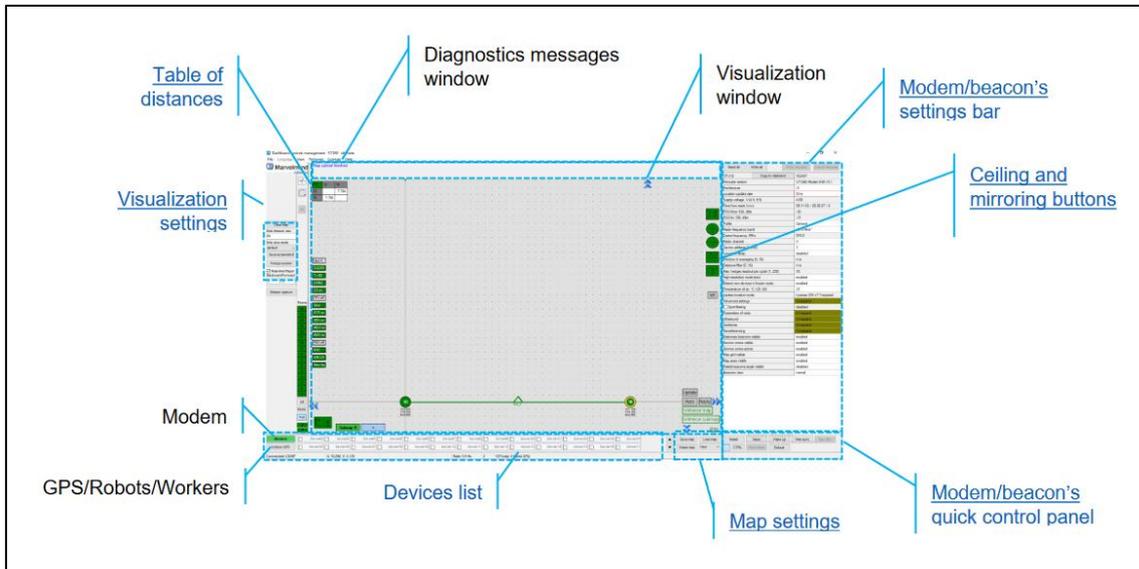


Figura 47. Vista general del Dashboard de Marvelmind

Comprobar la carga de las balizas es simple, debemos conectarla por cable al ordenador donde tengamos el software instalado, abrir el programa y verificar que el voltaje sea un valor dentro del rango indicado. Cuanto más cerca al mínimo, menor será el porcentaje de la batería y viceversa.

Supply voltage, V (3.50..4.30)	4.18
--------------------------------	------

Figura 48. Ejemplo de lectura del voltaje de una baliza

Para activar como poner en modo reposo las balizas, deberemos ir a la lista que aparece en la parte inferior de la pantalla, allí, al hacer doble clic en una de ellas cambiará su estado y se pondrá de color verde para las estáticas y azul para las móviles (en inglés *hedge*).

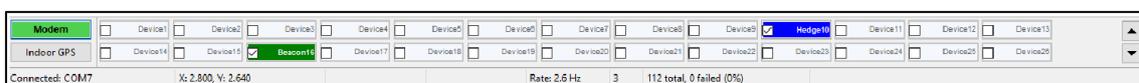


Figura 49. Lista de balizas en el Dashboard de Marvelmind

3.2 Kio Setup Manager

Las **balizas BLE** de [Kontakt.io](https://kontakt.io) se pueden monitorizar y configurar a través de una aplicación móvil denominada **Kio Setup Manager**. Sin embargo, para hacer uso de la aplicación necesitaremos iniciar sesión en Kio Cloud (pedir las credenciales a los tutores Alexis y/o Gabriele).

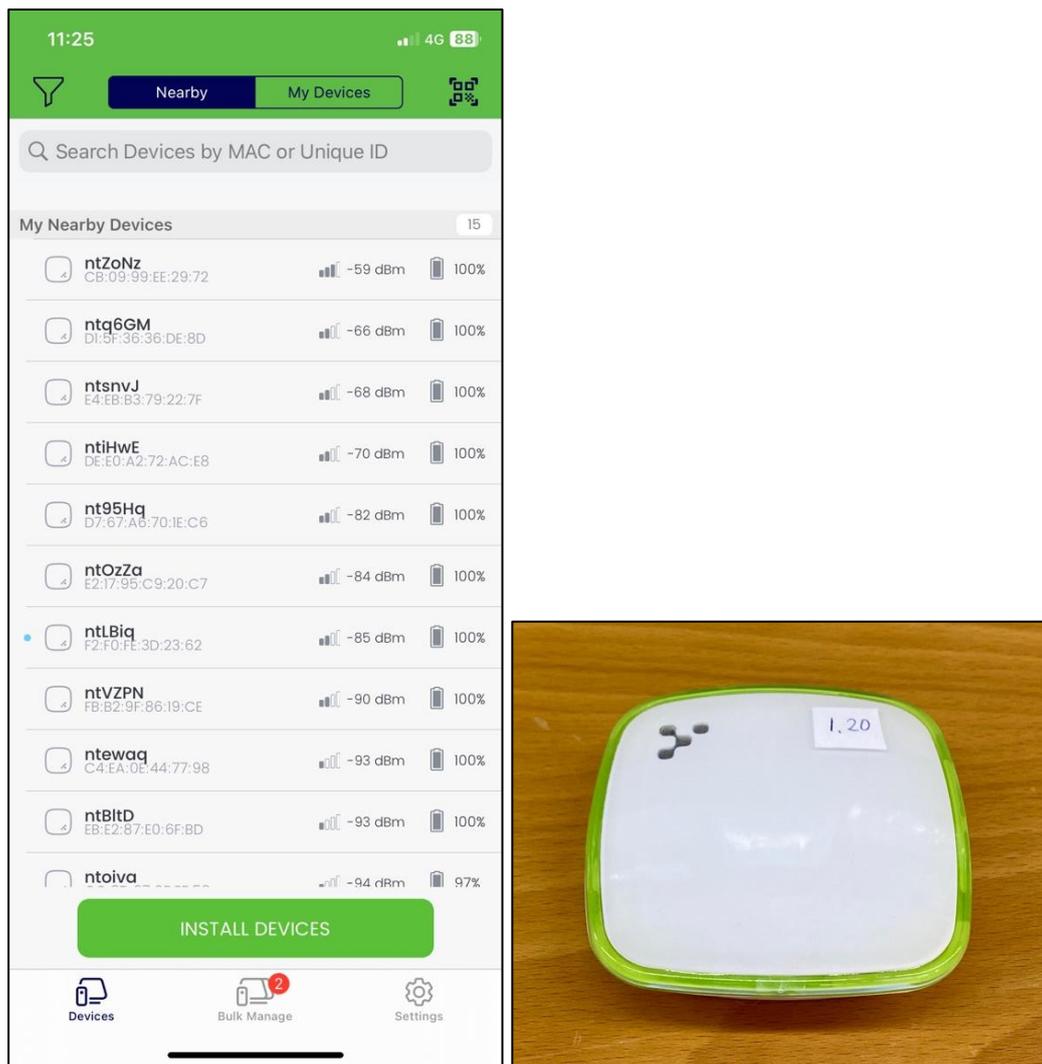


Figura 50. Aplicación móvil Kio Setup Manager y baliza BLE de Kontakt.io

Una vez tengamos la sesión iniciada y el Bluetooth del móvil esté activado, podremos ver los dispositivos vinculados a nuestra cuenta y aquellos de los que recibimos señales. Por último, pinchando en cada uno de ellos, accederemos a un menú más detallado donde podremos configurarlos a nuestro gusto.

3.3 Blender

La herramienta software [Blender](#), nos permite diseñar y modelar objetos en 3D. Utilizando una versión superior a la 2.8 y con la extensión “Robotcontrol editor toolbox” desarrollada por el compañero Alberto Ramos, podremos diseñar escenarios, rutas y controlar de forma remota a ROBOMAP.

Encontraremos información relevante sobre la instalación de la extensión, así como, pequeñas explicaciones sobre el manejo de la herramienta en su memoria de TFG y en el repositorio:

<https://github.com/bertoramos/blender-editor>

Una vez tengamos todo instalado correctamente podremos diseñar o importar una escena, que no es más que una representación digital del espacio físico donde realizaremos la campaña. Se hace básicamente para ubicar a la plataforma dentro del espacio designado y así detectar posibles colisiones con antelación. Para importar una escena, es tan simple como abrir el archivo .blend que lo contenga.

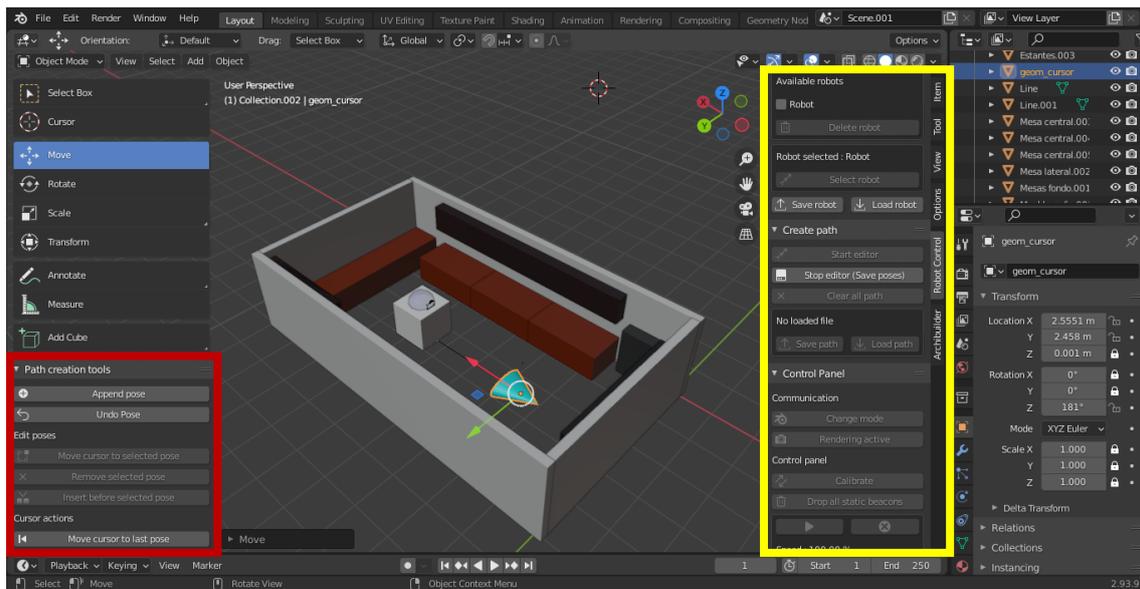


Figura 51. Menús y funcionalidades de la herramienta Robotcontrol en Blender

Al presionar la tecla ‘N’ de nuestro teclado deberíamos tener un menú desplegable igual al que vemos en el recuadro amarillo de la imagen anterior, que corresponde con una captura del software Blender con la escena que representa a la biblioteca del IUCES. Y, si navegamos hasta la pestaña “Robot Control”, accederemos a dicha herramienta y podremos crear un robot.

Podremos modificar un poco los campos salvo las direcciones IP y los puertos, que sí deben ser comunes a los indicados en los archivos de configuración del equipo principal en ROBOMAP. A continuación, deberemos seleccionar el robot que queramos de los que nos salgan listados y podremos entonces utilizar las funciones deseadas, como diseñar una ruta o usar el control remoto.

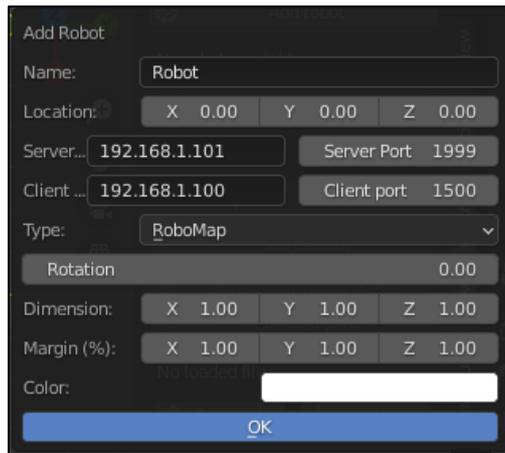


Figura 52. Ventana de creación de robot de la herramienta Robotcontrol en Blender

Para acceder al modo de edición y/o creación de rutas tendremos que seleccionar un robot y darle al botón “Start editor”. Nos debería salir el menú desplegable del recuadro rojo de la Figura 11 y ser visible el triángulo de color cian de la Figura 13. Una vez hayamos desplazado y/o cambiado la orientación del cursor, si queremos guardar esa posición utilizaremos la opción “Append pose” del recuadro rojo.

Para establecer una conexión con la plataforma desde Blender, debemos ejecutar el código en C/C++ del módulo ROBOMAP en el ordenador principal y, a continuación, clicar el botón del recuadro azul que aparece en la siguiente figura.

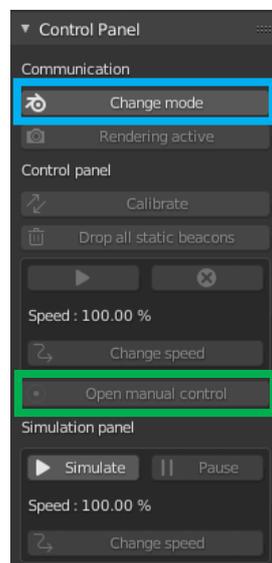


Figura 53. Panel de control de la herramienta Robotcontrol en Blender

Para manejar manualmente de manera remota a ROBOMAP, hay que establecer primero la conexión descrita en el párrafo anterior y, al clicar en el botón del recuadro verde, “Open manual control”, controlaremos el movimiento de la plataforma mediante el teclado.

Otra aspecto importante para tener en cuenta, es que tras establecer conexión con el MR, si pulsamos en el botón “Calibrate”, que se encuentra en el menú “Control Panel”, podremos ajustar la posición de las balizas de ultrasonido estáticas en la escena virtual.

3.4 R+ Manager

Con la herramienta software R+ Manager podremos configurar las características de cada uno de los motores DYNAMIXEL que conforman a la plataforma. [Manual en formato digital](#).

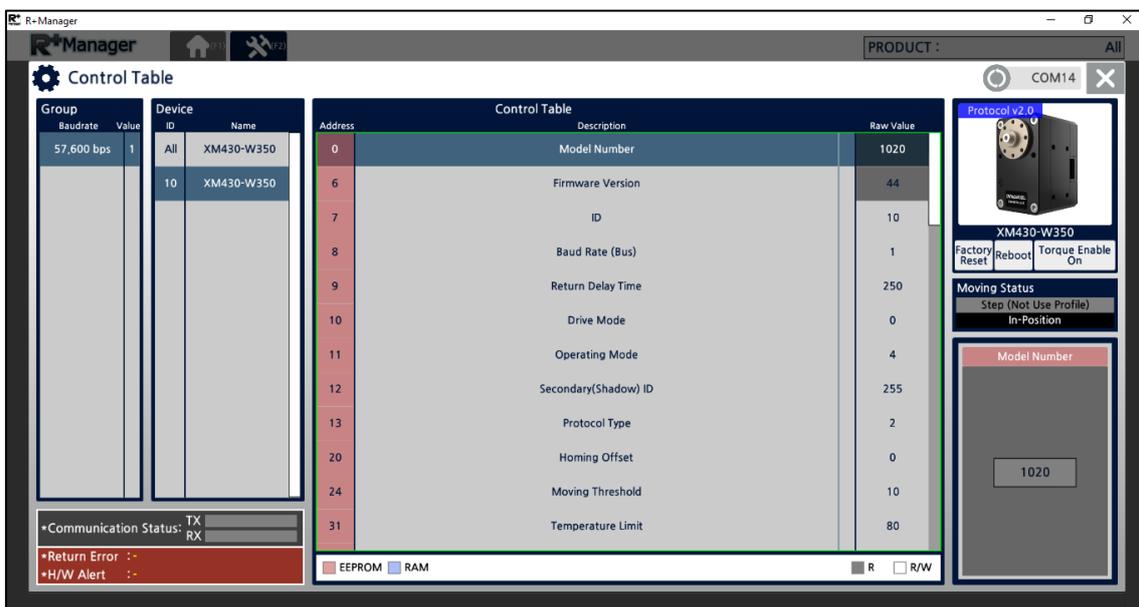


Figura 54. Ventana de configuración de los motores DYNAMIXEL en R+Manager

3.5 ROBOMAP Inspector

Es una interfaz gráfica desarrollada con PySimpleGUI que permite ver las alertas generadas durante una campaña, además, si establecemos la conexión con la plataforma una vez está ya allá comenzado, se puede solicitar el historial y ver las alertas previamente generadas en otro campo para no mezclar, las recibidas en directo y las antiguas.

Además, gracias a la librería pyinstaller, se consigue tener ejecutables como un programa más de ordenador, evitando así tener Python instalado o ejecutar los archivos desde la terminal. Por el momento solo se tiene el programa para Windows (.exe) pero es tan sencillo obtenerlo para Mac como ejecutar el siguiente comando:

```
“pyinstaller --noconsole --onfile nombre_del_archivo.py”
```

Se definió una alerta como un mensaje estructurado en dos partes: el nivel o importancia y la descripción. La primera indica la relevancia de la segunda. Además, cada nivel tiene asociado un color para aumentar la visibilidad de las alertas.

Color	Nivel	Significado
Verde	Ok	Todo correcto.
Amarillo	Warning	Algo puede estar mal, comprobar el cuerpo de la alerta.
Rojo	Error	Algo no salió como se esperaba.
Azul	Ack	Reconocimiento, por ejemplo, la conexión con el servidor.

Luego, en la carpeta “C:\IPS>alertsModule\” está el código y el ejecutable previamente generado de la interfaz y, en el archivo Servidor_MA se describen las funcionalidades de almacenamiento de las alertas y el código para las peticiones de conexión y envío del historial.

Por otro lado, en “C:\IPS\controladorModule\Service_Alerts” están los diferentes tests ejemplo elaborados, donde, cada uno de ellos genera una alerta según el resultado de las pruebas, solo cambia el cuerpo de la alerta en cada función.

Por último, la invocación de los tests o generación de alertas se hace desde el MC, tal y como si fuera un servicio más.

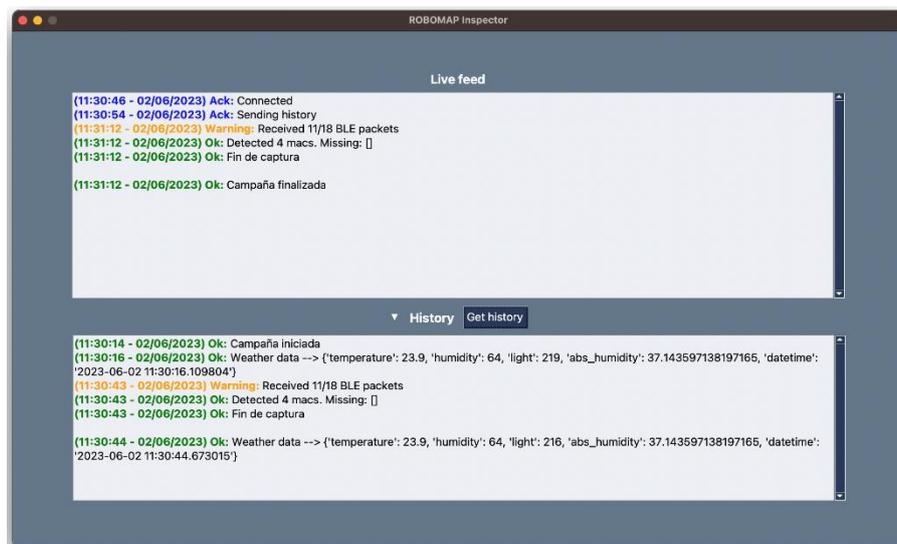


Figura 55. Interfaz gráfica ROBOMAP Inspector.

3.6 Ejecución de código

Los programas que describen el comportamiento de ROBOMAP deberemos ejecutarlos desde terminales diferentes.

- Servidores MAD-BLE, MOR-BLE y MM

“C:\IPS\controladorModule\desplegarServidores\despliegueServidores.py”

- Servidor MR en equipo principal

“C:\IPS\robomapModule\Release\robotips5_msvc2019.exe”

- Módulo Controlador

“C:\IPS\controladorModule\controlador.py”

4. POST-PROCESADO DE DATOS

Una vez finalizada la campaña para los puntos de referencia y los aleatorios, habrá que tratar la información almacenada en la base de datos (BBDD), lo que se conoce como post-procesado de datos, para ser utilizada más adelante con los algoritmos que estimarán la posición del objeto en cuestión.

Al momento de redactar este documento existen una serie de programas, en Python y Java, que obtendrán la información de la herramienta SQLite y la volcarán en archivos con formato “.csv” y “.xlsx”.

Con Python, extraeremos la información de la BBDD y por cada punto de referencia, crearemos diferentes archivos .csv que corresponden a la captura de señales BLE en cada una de las orientaciones indicadas al principio de la campaña. Se hará el mismo proceso con los puntos aleatorios que, aunque conocemos su ubicación al trasladar a ROBOMAP a dicha posición, será desconocido para el algoritmo de posicionamiento.

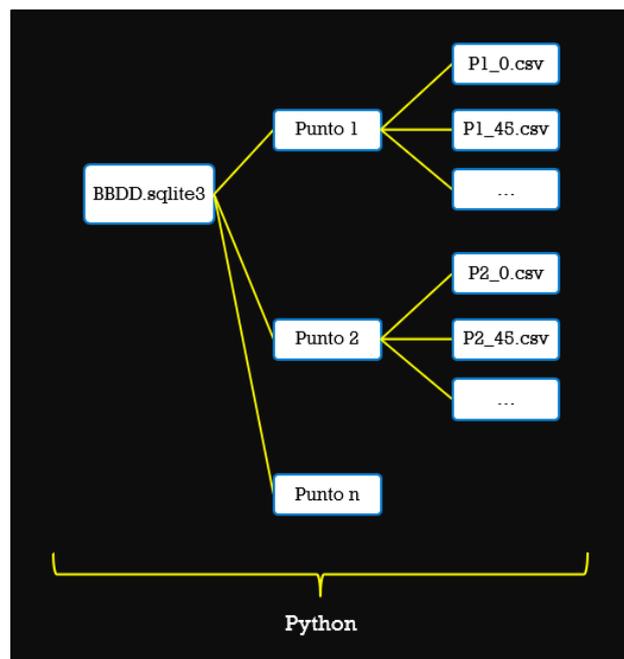


Figura 56. Estructura del procesamiento de datos con el programa en Python

Luego, con el programa descrito en Java, ubicaremos toda la información obtenida en un punto, en un único archivo .xlsx, el cual, estará dividido en diferentes hojas para separar las capturas de señales BLE según canal, protocolo y orientación.

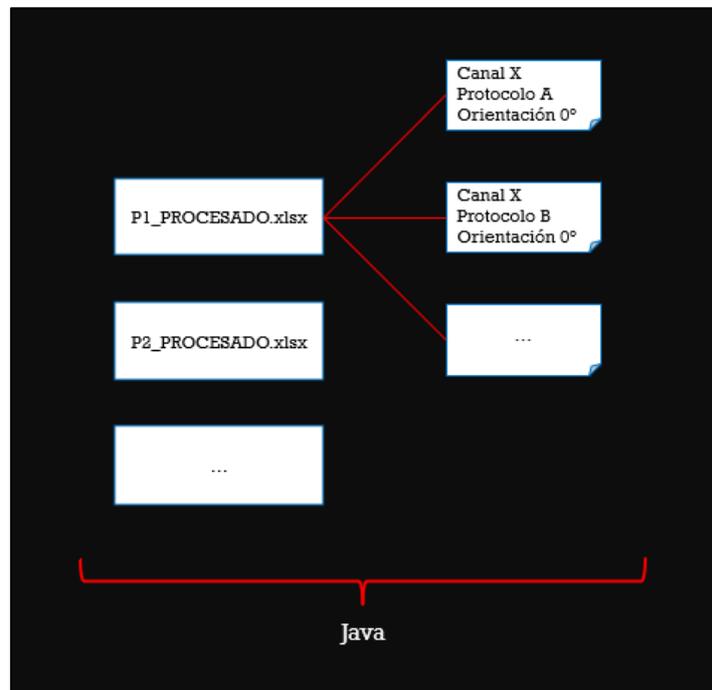


Figura 57. Estructura del procesamiento de datos con el programa en Java

4.1 Pasos

La carpeta “ProcesadoDatos” aloja todos los archivos del proyecto Java que realiza parte del post-procesado. En el mismo directorio, hay una carpeta llamada “Ejecutar_Programa” en la que encontraremos los ficheros necesarios para el resto de los pasos.

1. En la misma ubicación del archivo “Export_Files.py”, guardaremos las correspondientes bases de datos en formato .sqlite3.
2. En un fichero llamado “macs.txt” añadiremos las direcciones físicas de los beacons utilizados en la campaña.
3. En los archivos llamados “Config_Ref_Points.conf” y “Config_Ale_Points.conf” pondremos los parámetros de configuración usados durante la campaña.
4. Desde una terminal ejecutar el fichero “Export_Files.py”.