



UNIVERSIDAD  
DE MÁLAGA



**ESCUELA DE INGENIERÍAS INDUSTRIALES**

**Ingeniería Mecánica, Térmica y de Fluidos**

**Ingeniería Mecánica**

# **TRABAJO FIN DE MÁSTER**

**Sistema embarcado para el monitoreo y diagnóstico del  
estado del hilo de contacto en las infraestructuras del  
Metro de Málaga**

Máster en Ingeniería en Mecánica Avanzada

Autor: Rafael Jesús Luque Aguilar

Tutor: Sergio Postigo Pozo

Cotutor Montserrat Simarro Vega

MÁLAGA, Febrero de 2025





---

## RESUMEN

Este proyecto presenta el desarrollo de un dispositivo sensorizado integrado en la parte superior de las unidades ferroviarias del metro de Málaga, diseñado para recolectar información detallada sobre el estado de las instalaciones, con especial enfoque en la interacción pantógrafo-catenaria. El dispositivo incorpora varios sensores avanzados: un sensor de arco para detectar la presencia de arcos eléctricos, una unidad IMU que permite el registro de aceleraciones de la unidad, un módulo GPS para geolocalización y una cámara de alta definición que captura video continuo del estado actual de la catenaria. Gracias a una conexión remota, el dispositivo facilita la recolección y transmisión de datos de forma rápida y eficiente.

La implementación de este sistema sensorizado busca optimizar las tareas de mantenimiento en infraestructuras ferroviarias mediante la identificación proactiva de áreas críticas en la catenaria, agilizando el análisis del desgaste y las anomalías en el cable conductor. Con esta solución tecnológica, se mejora la efectividad de las inspecciones y se contribuye a la fiabilidad y seguridad del servicio ferroviario.

La motivación de este trabajo surge de la necesidad de obtener información acerca de las instalaciones de Metro de Málaga y contribuir a la búsqueda de anomalías y puntos de desgaste crítico del hilo de contacto. Todo ello, con el objetivo de realizar un estudio más en profundidad acerca de cómo afectan los distintos parámetros claves de la instalación e intentar de corregir esos puntos de desgaste modificando dichos parámetros.

**PALABRAS CLAVE:** ferroviario, pantógrafo-catenaria, mantenimiento predictivo, sensorización



---

## ABSTRACT

This project presents the development of a sensorized device integrated onto the roof of Málaga metro railway units, designed to gather detailed information about the state of infrastructure, with a particular focus on pantograph-catenary interaction. The device includes several advanced sensors: an arc sensor to detect electrical arcs, an IMU unit to record the train unit's accelerations, a GPS module for geolocation, and a high-definition camera that captures continuous video of the catenary's current state. Thanks to a remote connection, the device facilitates quick and efficient data collection and transmission.

The implementation of this sensorized system seeks to optimize railway infrastructure maintenance tasks by proactively identifying critical areas on the catenary, speeding up the analysis of wear and detecting anomalies in the contact wire. This technological solution enhances the effectiveness of inspections, contributing to the reliability and safety of railway services.

The motivation behind this project stems from the need to gather information about the installations of Metro de Málaga, aiding in the identification of anomalies and critical wear points on the contact wire. This aims to enable a more in-depth study of the impact of key installation parameters and explore adjustments to mitigate these points of wear by modifying said parameters.

**KEYWORDS:** railway, pantograph-catenary, predictive maintenance, sensorization



Sistema embarcado para el monitoreo y diagnóstico del estado del hilo de contacto en  
las infraestructuras del Metro de Málaga





## ÍNDICE

<b>1. ANTECEDENTES</b>	<b>6</b>
<b>2. OBJETIVOS</b>	<b>7</b>
<b>3. ESTADO DEL ARTE</b>	<b>8</b>
3.1. Pantógrafo	8
3.2. Catenaria	9
3.3. Interacción Pantógrafo-Catenaria	9
3.4. Desgaste en la interacción	10
3.5. Aparición de arcos eléctricos	10
3.6. Tareas de mantenimiento convencionales	11
3.7. Dispositivos de inspección de desgaste	11
<b>4. VISIÓN GENERAL DEL DISPOSITIVO</b>	<b>13</b>
4.1. Descripción del proyecto	13
4.2. Conexión y Seguridad del Dispositivo	14
<b>5. COMPONENTES Y CARACTERÍSTICAS DEL DISPOSITIVOS</b>	<b>16</b>
5.1. Mini Pc	16
5.2. Microcontroladores	17
5.3. Sensor de Arco	18
5.4. Cámara HD	20
5.5. Unidad de medición inercial (IMU)	22
5.6. Sistema de posicionamiento global (GPS)	22
<b>6. PROGRAMACIÓN</b>	<b>24</b>
6.1. Preparación del PC	24
6.2. Programa general	27
6.3. Programa de captura de arcos	30
6.4. Programa de captura de IMU y GPS	31
6.5. Programa de puerta de acceso al dispositivo	35
6.6. Control remoto	36
<b>7. FABRICACIÓN</b>	<b>38</b>
7.1. Fabricación mecánica	38
7.2. Conexiones eléctricas y cableado	43
<b>8. ENSAYOS</b>	<b>47</b>
8.1. Instalación del dispositivo	47



---

8.2. Realización del ensayo _____	48
8.3. Mantenimiento del dispositivo _____	50
<b>9. RESULTADOS _____</b>	<b>51</b>
<b>10. FUTURAS Y PASADAS ACTUALIZACIONES _____</b>	<b>55</b>
<b>11. REFERENCIAS _____</b>	<b>58</b>
<b>ANEXO A CÓDIGO GENERAL _____</b>	<b>59</b>
<b>ANEXO B CÓDIGO CAPTURA DE ARCOS _____</b>	<b>67</b>
<b>ANEXO C CÓDIGO PUERTA DE ACCESO _____</b>	<b>69</b>
<b>ANEXO D CÓDIGO CAPTURA DE IMU Y GPS _____</b>	<b>73</b>



## ÍNDICE DE FIGURAS

<i>Figura 1. Pantógrafo ferroviario en contacto con cable de alta tensión. (Ref. 8)</i>	8
<i>Figura 2. Dispositivos de inspección de desgaste</i>	12
<i>Figura 3. Dispositivo colocado en una unidad de Metro de Málaga</i>	15
<i>Figura 4. Esquema representativo con los distintos elementos del dispositivo</i>	14
<i>Figura 5. Detalle de los componentes que incorpora el dispositivo</i>	17
<i>Figura 6. Sensibilidad del sensor de arco en función del ángulo proporcionada por el fabricante</i>	19
<i>Figura 7. Sensor de arco eléctrico UV de la marca slgux utilizado en el dispositivo</i>	20
<i>Figura 8. Cámara HD de la marca Arducam utilizada en el dispositivo</i>	21
<i>Figura 9. Linterna incorporada al dispositivo para mejorar la visibilidad</i>	21
<i>Figura 10. Esquema del sistema operativo y herramientas de programación utilizadas</i>	25
<i>Figura 11. Esquema de flujo de trabajo del programa general</i>	29
<i>Figura 12. Esquema del flujo de trabajo del programa que captura los arcos eléctricos</i>	31
<i>Figura 13. Interfaz gráfica para el control de los elementos del dispositivo</i>	32
<i>Figura 14. Esquema del flujo de trabajo del control de la página web</i>	34
<i>Figura 15. Esquema del flujo de trabajo de la captura los datos de los sensores</i>	34
<i>Figura 16. Interfaz gráfica para la puerta de acceso al dispositivo</i>	35
<i>Figura 17. Dispositivo embarcado en detalle</i>	39
<i>Figura 18. Vista en planta del dispositivo</i>	40
<i>Figura 19. Detalle de los anclajes que sujetan el dispositivo con la unidad</i>	41
<i>Figura 20. Esquema del sistema de alimentación del dispositivo</i>	43
<i>Figura 21. Esquema de comunicación entre los elementos del dispositivo</i>	45
<i>Figura 22. Placa electrónica del dispositivo</i>	45
<i>Figura 23. Detalle de la parte trasera de la placa electrónica del dispositivo</i>	46
<i>Figura 24. Detalle del uso de batería para configuración el día de la instalación</i>	47
<i>Figura 25. Comprobación de la correcta visualización el día de la instalación</i>	48
<i>Figura 26. Comparativa de la vista entre las dos alturas de trabajo del pantógrafo</i>	49
<i>Figura 27. Dispositivo tras cierto tiempo instalado en una unidad</i>	50
<i>Figura 28. Muestra de arco eléctrico capturado por la cámara</i>	51
<i>Figura 29. Muestra del mapa de calor generado para localización de zonas conflictivas</i>	52
<i>Figura 30. Arco eléctrico visto desde el sensor</i>	53
<i>Figura 31. Representación de las aceleraciones registradas por IMU en los 3 ejes principales</i>	53
<i>Figura 32. Representación de las señales de GPS obtenidas por el dispositivo en un mapa real</i>	54
<i>Figura 33. Representación de la velocidad proporcionada por el GPS del dispositivo</i>	54
<i>Figura 34. Comparativa entre distintas versiones del dispositivo</i>	56



Sistema embarcado para el monitoreo y diagnóstico del estado del hilo de contacto en las infraestructuras del Metro de Málaga





# MEMORIA



## 1. ANTECEDENTES

Este proyecto se desarrolla en el marco del programa de apoyo “Proyectos de generación de conocimiento”, cuyo objetivo es la formación de personal investigador en etapa predoctoral vinculado a dichos proyectos, dentro del Programa Estatal para Impulsar la Investigación Científico-Técnica y su Transferencia, según el Plan Estatal de Investigación Científica, Técnica y de Innovación 2021-2023.

El objetivo principal es desarrollar un modelo matemático que permita predecir el desgaste que se produce en la interacción pantógrafo-catenaria. Para ello, se han diseñado dispositivos específicos tanto para la recolección directa de datos como para la simulación de condiciones reales. Esta memoria documenta el desarrollo de uno de estos dispositivos, destinado a localizar áreas críticas de desgaste en la infraestructura.

El proyecto está dividido en ocho paquetes de trabajo, distribuidos en cuatro áreas principales: trabajo en túnel, en laboratorio, en simulaciones y en modelado. El proceso comienza con la identificación de las zonas más propensas al desgaste en las instalaciones del metro de Málaga mediante la colocación de un dispositivo sensorizado en uno de los vagones. Este dispositivo recopila datos en cada trayecto, permitiendo anticipar las áreas con mayor desgaste en el tiempo. Una vez delimitadas estas zonas, se procede a medir el estado actual del cable de catenaria en dichas áreas críticas.

Simultáneamente, en el laboratorio se realizan ensayos modales para parametrizar el comportamiento dinámico del pantógrafo [1], además de simulaciones teóricas de desgaste. Con los datos obtenidos sobre desgaste y comportamiento dinámico, se elabora un modelo inicial de la interacción pantógrafo-catenaria. Este modelo, junto con las condiciones de infraestructura recogidas inicialmente, permite proyectar valores de desgaste para un periodo de tiempo determinado.



## 2. OBJETIVOS

El presente trabajo tiene como propósito desarrollar un dispositivo sensorizado capaz de recopilar datos críticos sobre el estado de las instalaciones ferroviarias en la interacción pantógrafo-catenaria. Dado que el diseño del dispositivo ha sido dinámico y sujeto a modificaciones conforme avanzaba el proyecto, algunos objetivos han cambiado de prioridad, lo que ha derivado en múltiples actualizaciones a lo largo de su desarrollo. Estas actualizaciones incluyen la incorporación de componentes adicionales y la optimización de elementos originales para cumplir con nuevas necesidades detectadas en el proceso.

Específicamente, los objetivos de este proyecto son:

- Recolectar información sobre los arcos eléctricos que ocurren en la instalación de la catenaria.
- Obtener datos visuales detallados del estado de la instalación mediante la captura continua de video.
- Recopilar información adicional sobre las aceleraciones y la ubicación geográfica de la unidad ferroviaria en la que está montado el dispositivo.
- Establecer una conexión remota fiable que permita la transmisión de datos sin necesidad de conexión física.

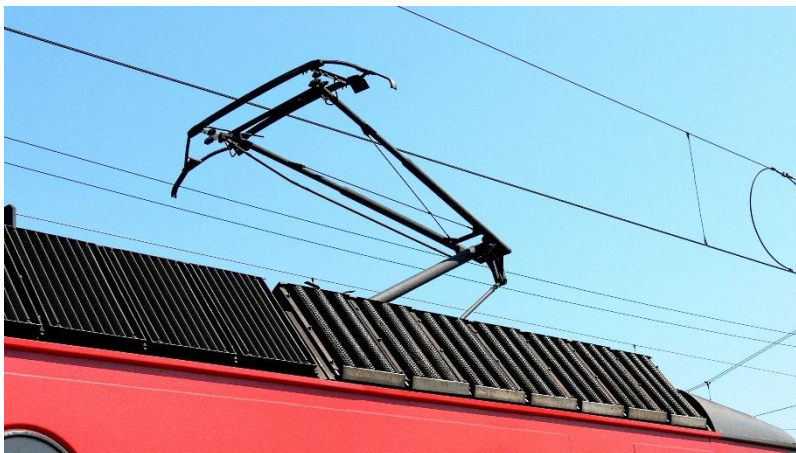
Estos objetivos guían el diseño y las mejoras implementadas, buscando optimizar la eficacia del dispositivo en la obtención de datos para el mantenimiento predictivo y preventivo de las infraestructuras ferroviarias.

### 3. ESTADO DEL ARTE

Para entender el problema que se aborda con la fabricación del dispositivo embarcado para la detección de zonas de desgaste de hilo conductor, es fundamental revisar los conceptos básicos que permiten comprender el proyecto en su totalidad.

#### 3.1. Pantógrafo

Un pantógrafo ferroviario es un dispositivo mecánico utilizado en los trenes eléctricos para captar la energía eléctrica de una línea aérea de contacto y transferirla al sistema de tracción del tren y sistemas auxiliares. Este mecanismo permite que los trenes funcionen eficientemente sin depender de una fuente de energía a bordo, como un motor de combustión o una máquina de vapor. En la Figura 1 se puede observar la morfología de uno de estos pantógrafos situados en la parte superior de la unidad ferroviaria.



*Figura 1. Pantógrafo ferroviario en contacto con cable de alta tensión. (Ref. 8)*

El pantógrafo está compuesto principalmente por una estructura que actúa como soporte de los componentes encargados de captar la energía. La estructura incluye una serie de brazos articulados y segmentos montados en la parte superior del tren. Estos brazos son ajustables, permitiendo mantener un contacto adecuado con la línea aérea de contacto. La cabeza del pantógrafo, la parte que toca la catenaria, es esencial para asegurar una conexión eléctrica estable sin dañar el cable conductor.

A medida que el tren se desplaza, el pantógrafo se mueve a lo largo de la línea aérea, recolectando energía eléctrica que puede ser de corriente continua o alterna, en función del sistema de electrificación de la red ferroviaria. La cabeza del pantógrafo está formada habitualmente por dos bandas de contacto o pastillas, fabricadas en materiales como grafito,



cobre o una combinación de ambos. Este es el componente que establece el contacto directo con la catenaria y, en consecuencia, el que experimenta el mayor desgaste debido al uso y las condiciones de operación.

El accionamiento del pantógrafo se basa en un ajuste controlado de sus brazos superiores contra la línea aérea. Durante el desplazamiento del tren, un mecanismo, generalmente neumático, controla electrónicamente la presión y posición del pantógrafo para asegurar un contacto continuo y seguro. Este sistema garantiza que el pantógrafo capte la energía necesaria de manera eficiente, la cual se transmite a los motores para la propulsión del tren y al resto de sistemas.

### **3.2. Catenaria**

La catenaria es la línea aérea que transmite transmitir la energía eléctrica a las locomotoras. Consiste en un conjunto de cables conductores que se encuentran suspendidos sobre las vías ferroviarias y que sirven como el principal medio de transmisión de energía eléctrica. Estos cables se mantienen en posición mediante soportes específicos que varían según el tipo de catenaria utilizado.

Los sistemas de catenaria funcionan en tensiones que generalmente oscilan entre 600 V y 3 kV en corriente continua y entre 15 kV y 25 kV en corriente alterna. La catenaria suele actuar como el polo positivo, mientras que las vías hacen las veces de polo negativo, completando así el circuito eléctrico.

Existen distintos tipos de catenaria en función del entorno donde se utilizan. Para las instalaciones al aire libre, es común el uso de catenarias flexibles, que se conforman de un cable suspendido sostenido por una serie de anclajes y tirantes. En cambio, en instalaciones subterráneas es habitual el uso de catenarias rígidas, que consisten en un perfil metálico continuo dispuesto a lo largo de la instalación.

### **3.3. Interacción Pantógrafo-Catenaria**

El contacto mecánico y eléctrico entre el pantógrafo, que se encuentra en la parte superior de los trenes eléctricos, y la catenaria o línea aérea de contacto, que suministra la energía necesaria para el funcionamiento del tren, se denomina interacción pantógrafo-catenaria. Esta interacción es fundamental para la transmisión continua y eficiente de corriente eléctrica, permitiendo que el tren mantenga su potencia y velocidad.



Durante esta interacción, el pantógrafo, mediante sus bandas de contacto o mesillas, roza de manera controlada la catenaria mientras el tren se desplaza. La presión del pantógrafo sobre la catenaria debe ser lo suficientemente constante para asegurar una buena transmisión de energía sin interrupciones, evitando al mismo tiempo un desgaste excesivo en ambos componentes. Sin embargo, factores como la velocidad del tren, el estado de la catenaria y las condiciones ambientales pueden afectar esta conexión, generando fricción, desgaste, y en algunos casos, arcos eléctricos en el punto de contacto. Estos efectos pueden impactar la durabilidad de los materiales y la seguridad de la operación, haciendo esencial su estudio y monitoreo en la ingeniería ferroviaria.

### **3.4. Desgaste en la interacción**

El desgaste en la interacción pantógrafo-catenaria está influenciado por varios factores, pero este trabajo se enfoca en uno específico: la aparición de arcos eléctricos [2, 3]. A medida que el sistema opera, los materiales de contacto, generalmente el grafito de las mesillas del pantógrafo y el cobre del hilo de contacto, se erosionan progresivamente. El diseño prioriza el desgaste de las mesillas del pantógrafo, ya que su reemplazo es más sencillo y económico que el del hilo de contacto completo. Sin embargo, esto no impide que el cable de la catenaria también sufra deterioro con el tiempo.

### **3.5. Aparición de arcos eléctricos**

A medida que los elementos de la instalación se desgastan, la aparición de arcos eléctricos aumenta. Esto sucede porque las superficies de contacto desarrollan irregularidades con el desgaste. Durante el deslizamiento, el contacto entre pantógrafo y catenaria se vuelve inestable, similar al comportamiento de un neumático en mal estado sobre una carretera llena de baches e inclinaciones: el hilo de contacto adquiere desniveles, y el pantógrafo presenta irregularidades en su perfil. Al desplazarse a cierta velocidad, estos desniveles provocan una separación momentánea en el contacto, generando arcos eléctricos cuya intensidad dependerá tanto de la magnitud de la separación como de la demanda de corriente del tren en ese instante.

Además, la aparición de arcos eléctricos aumenta el desgaste en los materiales de contacto, por lo que es un efecto que se retroalimenta así mismo. En aquellas zonas donde hay un desgaste leve, comienzan la aparición de arcos eléctricos que incrementan el desgaste en esa misma zona. Por lo que en aquellas zonas donde aparecen pequeños arcos, es normal que al paso del tiempo se vayan incrementando tanto la energía como la frecuencia de los arcos propiciando a un problema de desgaste.



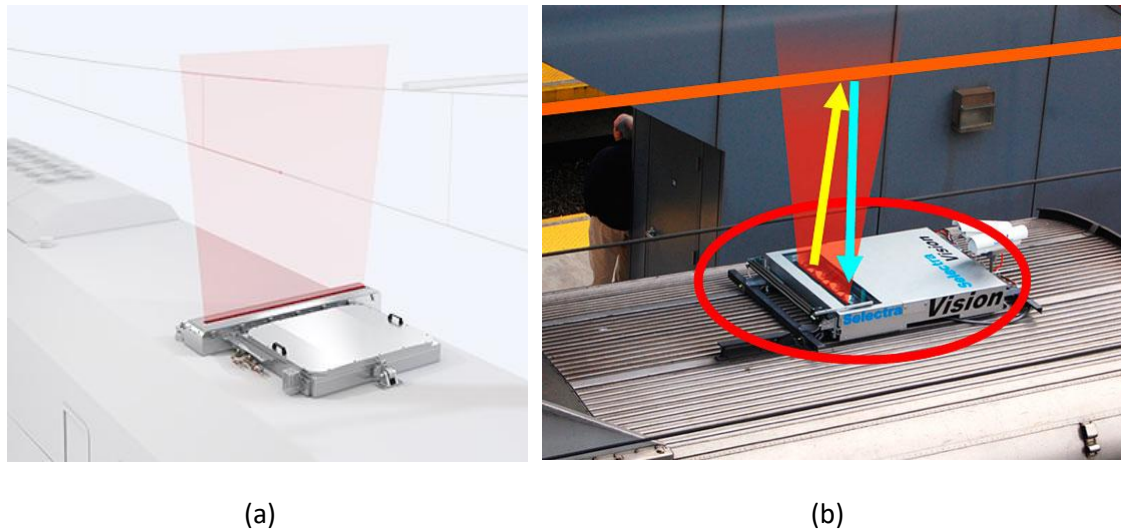
### 3.6. Tareas de mantenimiento convencionales

Actualmente, las infraestructuras ferroviarias monitorean el desgaste del hilo de contacto de forma manual. Esto implica que un operario debe situarse a la altura de la catenaria y, con un calibre, registrar el desgaste en puntos kilométricos específicos de la instalación. Sin embargo, antes de proceder, es necesario cortar la tensión en la catenaria por seguridad, y el operario debe acceder a estos puntos mediante una carretilla especializada. Este método de mantenimiento es costoso y poco eficaz, ya que solo permite inspeccionar puntos aislados de la infraestructura sin cubrirla en su totalidad, lo cual hace prácticamente inviable realizar una revisión completa en una sola noche de mantenimiento.

Por ello, contar con un dispositivo montado en la parte superior de las unidades que mida los arcos eléctricos puede simplificar el mantenimiento. Con solo una pasada de la unidad en funcionamiento, es posible identificar las zonas de mayor desgaste en la instalación, optimizando tiempo y costos de mantenimiento y proporcionando un diagnóstico más completo de la infraestructura.

### 3.7. Dispositivos de inspección de desgaste

Aunque la mayoría de los dispositivos actuales se centran en detectar el desgaste de los raíles en infraestructuras ferroviarias, también existen empresas y equipos de investigación que han desarrollado tecnologías para identificar zonas de desgaste en el hilo de contacto. Entre estas empresas destacan *SelectraVision* y *Mermec*, que afirman poder medir milímetros de desgaste del hilo de contacto mientras la unidad está en funcionamiento [4, 5]. Aunque se dispone de poca información sobre las tecnologías específicas que utilizan, sus sistemas están diseñados para operar en condiciones de trabajo. Sus dispositivos se presentan en la Figura 2.



**Figura 2.** Dispositivos de inspección de desgaste. (a) Esquema del dispositivo de la empresa Mermec. (b) Dispositivo real de la empresa. (Ref. 4, 5)

En el ámbito de la investigación, se emplean diversas técnicas para detectar el desgaste en el hilo de contacto. Una de estas técnicas es la detección de arcos eléctricos mediante detectores de rayos ultravioleta, que permiten identificar directamente la presencia de estos arcos. Otros equipos utilizan cámaras termográficas para registrar las variaciones de temperatura que provocan los arcos eléctricos, mientras que algunos emplean cámaras estándar y luego procesan las imágenes capturadas para identificar áreas de desgaste.

Es importante señalar que la mayoría de los equipos de inspección actuales son de gran tamaño y presentan una complejidad considerable para su implementación en una unidad, ya sea por la cantidad de sensores que integran o por la falta de opciones para conexión remota.

Después de varias búsquedas, no se ha identificado ningún dispositivo de bajo costo que permita la detección temprana de zonas de desgaste y que ofrezca conexión remota. Por lo tanto, el objetivo de este trabajo es diseñar un dispositivo que cumpla con todas estas características, cubriendo una necesidad que hasta ahora no ha sido resuelta.



#### 4. VISIÓN GENERAL DEL DISPOSITIVO

El dispositivo desarrollado en este proyecto tiene como propósito la recolección de información sobre el estado de la infraestructura ferroviaria en las unidades de Metro de Málaga, con un enfoque en la interacción pantógrafo-catenaria y la identificación de posibles zonas de desgaste en el hilo de contacto. Este sistema busca anticipar los daños en la infraestructura y optimizar las tareas de mantenimiento mediante un monitoreo constante y automatizado, eliminando la necesidad de inspecciones manuales extensivas. En la Figura 3 se muestra un esquema general de las características del dispositivo.

##### 4.1. Descripción del proyecto

Desde el comienzo, el diseño se centró en la creación de una estructura compacta, similar a una caja de mínimo tamaño, que pudiera ser instalada en la parte superior de una de las unidades del metro (Figura ). A lo largo del proyecto, esta idea general se ha ido refinando, adaptándose a los requisitos de la instalación y asegurando la capacidad del dispositivo para funcionar de manera autónoma en un entorno ferroviario desafiante.

El dispositivo tiene como objetivo la detección y el registro de arcos eléctricos que se producen en el contacto entre el pantógrafo y la catenaria. Estos arcos son señales tempranas de desgaste en el hilo de contacto y pueden proporcionar información crítica sobre el estado de la infraestructura. Para esta tarea, se han implementado dos sistemas de detección en paralelo:

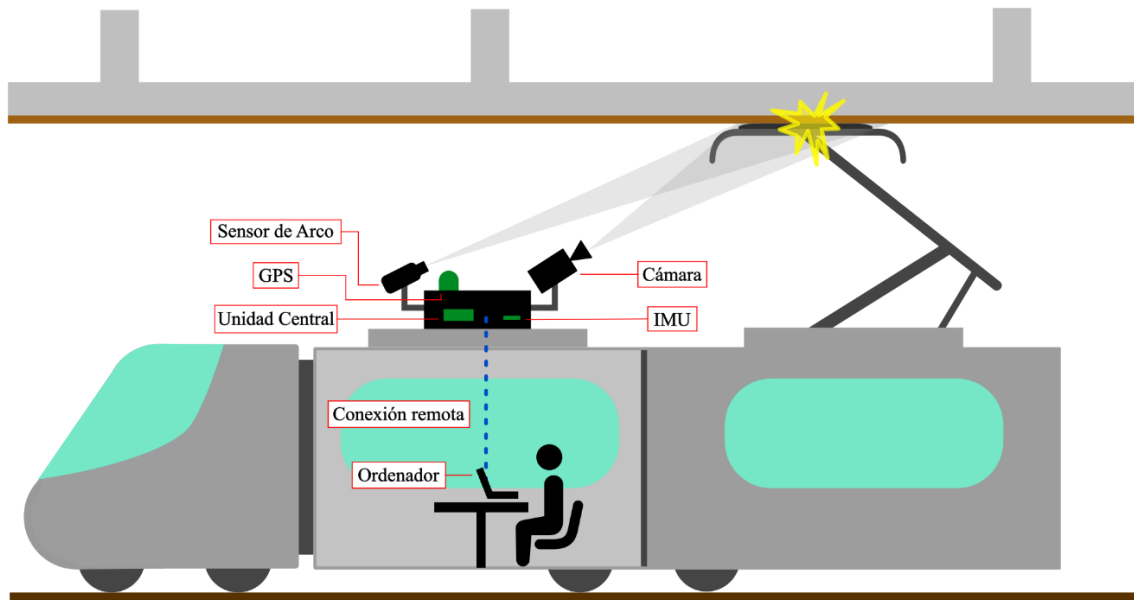
**1. Sensor de arcos eléctricos:** Este sensor está diseñado específicamente para capturar la aparición de arcos en tiempo real. Los arcos, al ser eventos de muy corta duración, requieren un sensor con alta frecuencia de muestreo para evitar la pérdida de datos importantes y con alta sensibilidad para recoger arcos en un amplio espectro de energía.

**2. Cámara de alta definición:** Para complementar la información proporcionada por el sensor de arcos, se ha integrado una cámara HD que graba el estado de la instalación. La combinación de las imágenes con los datos del sensor permite un análisis visual y un cruce de información que mejora la detección de zonas de desgaste.

Gran parte de la red de Metro de Málaga se encuentra en tramos subterráneos, por lo que se ha incorporado una linterna de alta intensidad que proporciona iluminación adicional. Esto permite que la cámara capture imágenes claras incluso en zonas con poca luz, asegurando una visibilidad adecuada del estado del hilo de contacto.

Además de los sistemas de detección de arcos y video, el dispositivo está equipado con un sensor IMU y un GPS. El sensor IMU permite registrar las aceleraciones y vibraciones que experimenta la unidad, lo que ayuda a comprender el contexto en el que se producen los arcos eléctricos. Por su parte, el GPS facilita la geolocalización de cada evento registrado en la superficie de la instalación, permitiendo un mapeo exacto de las zonas afectadas a lo largo de la línea.

Todos estos datos, tanto los captados por los sensores como los visuales, son gestionados por un conjunto de microprocesadores que, a su vez, están conectados a un ordenador central ubicado dentro del dispositivo. Este ordenador actúa como el núcleo de procesamiento, almacenando la información y coordinando la captura simultánea de video y datos en tiempo real.



*Figura 3. Esquema representativo con los distintos elementos del dispositivo*

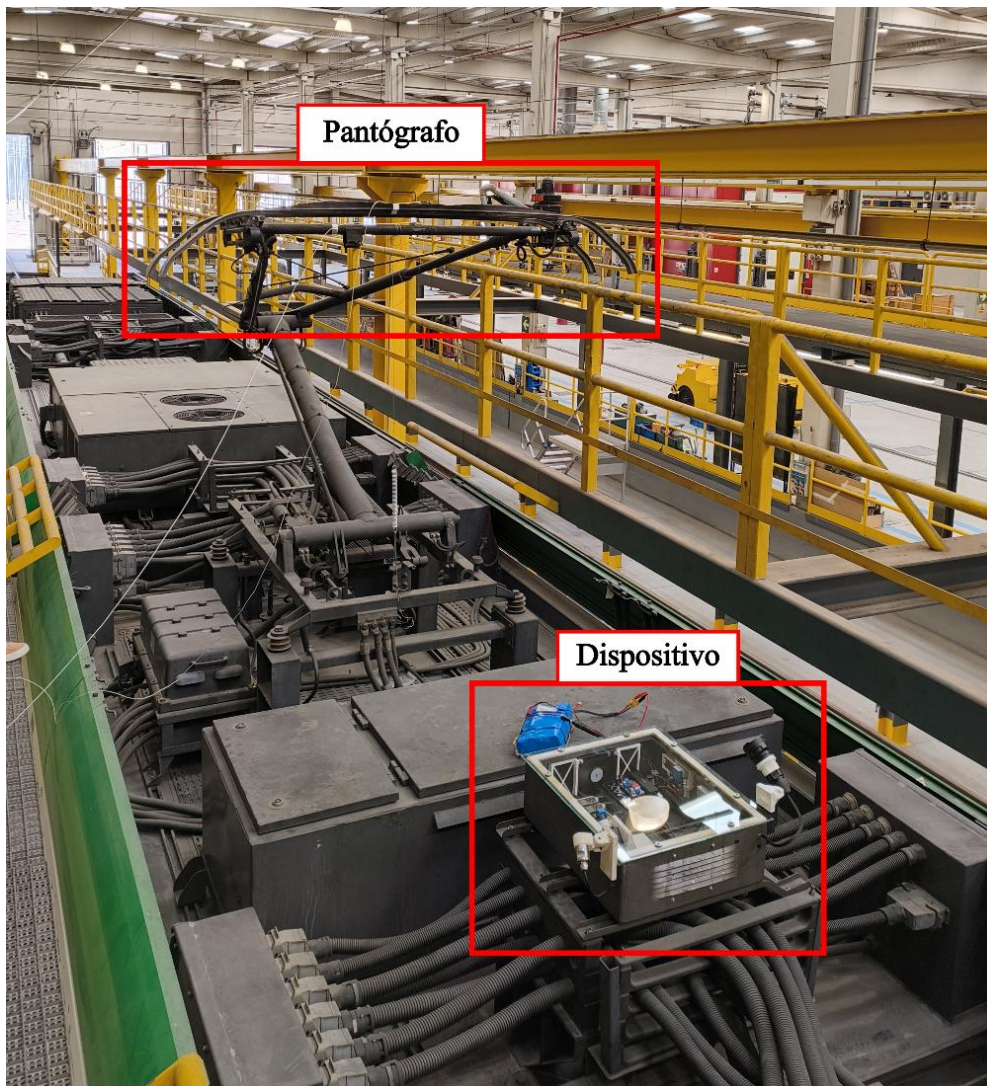
#### **4.2. Conexión y Seguridad del Dispositivo**

Una de las prioridades en el diseño del sistema ha sido facilitar su conexión y desconexión sin necesidad de cableado físico. Por esta razón, se ha implementado una conexión inalámbrica entre el ordenador central del dispositivo y el equipo de monitoreo remoto. Esta característica permite realizar los ensayos de manera sencilla, sin requerir una conexión física, lo cual simplifica el montaje y desmontaje del dispositivo sobre la unidad de metro.

Adicionalmente, la seguridad es un aspecto fundamental en el diseño del sistema. Para garantizar que no haya riesgos eléctricos innecesarios, el dispositivo cuenta con un

microcontrolador dedicado que actúa como puerta de acceso al suministro de energía. Este microcontrolador permanece en modo de espera, sin proporcionar tensión a los demás componentes, a menos que se inicie una sesión de ensayos. Solo una vez que se conecta el dispositivo y se otorgan los permisos necesarios, los demás elementos se ponen en tensión y están listos para comenzar el ensayo.

De esta manera, el sistema asegura que todos los componentes se encuentran apagados cuando no están en uso, reduciendo el riesgo de fallos o accidentes y optimizando la seguridad en cada sesión de ensayo. Este enfoque de diseño proporciona una solución confiable y segura para la recolección de datos, permitiendo un monitoreo continuo de la infraestructura ferroviaria sin comprometer la integridad del sistema o la seguridad del personal de mantenimiento.



**Figura 4.** Dispositivo colocado en una unidad de Metro de Málaga



## 5. COMPONENTES Y CARACTERÍSTICAS DEL DISPOSITIVO

En este apartado se realiza una descripción detallada de los principales componentes que conforman el dispositivo diseñado. Cada uno de los elementos que componen el sistema ha sido seleccionado y adaptado para cumplir con los objetivos del proyecto. A continuación, se explican las características, funcionalidades y el papel que desempeña cada componente, desde el ordenador central hasta los diversos sensores y microcontroladores, con el fin de ofrecer una visión clara y completa del dispositivo en su conjunto.

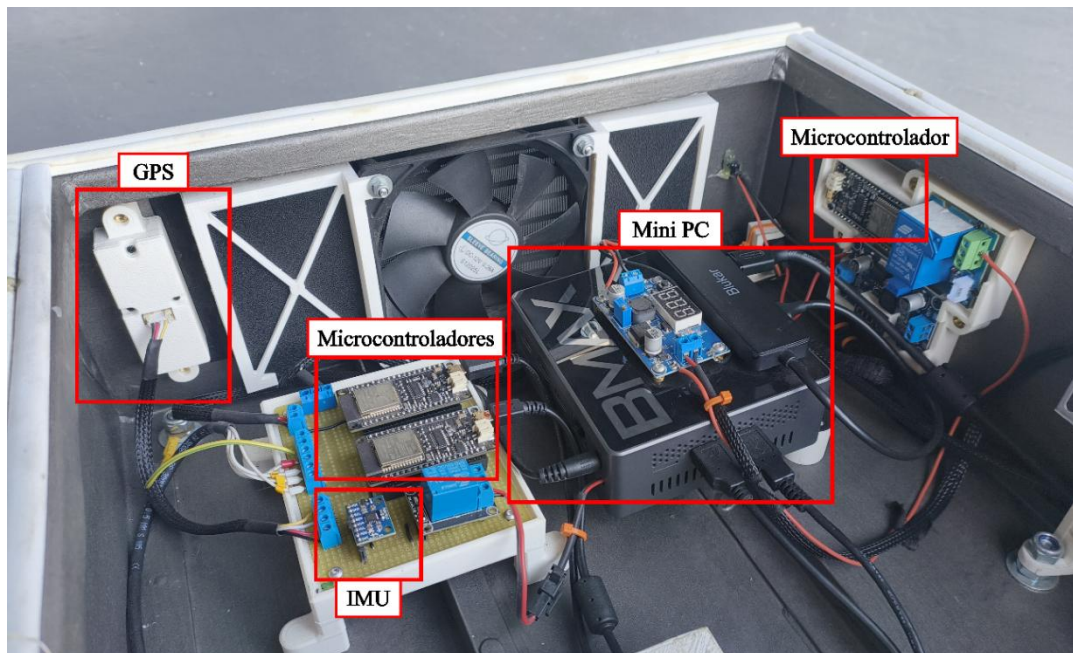
### 5.1. Mini Pc

El PC seleccionado para el dispositivo es el modelo *BMAX B2 Pro* (Figura 3), un mini PC que destaca por su rendimiento y tamaño compacto. Este equipo cuenta con un procesador *Intel Celeron N4000*. El procesador está acompañado de 8 GB de memoria RAM y un almacenamiento SSD de 256 GB, lo que permite una ejecución eficiente de las tareas sin comprometer el consumo energético.

El *BMAX B2 Pro* también incluye gráficos integrados *Intel UHD 600*, que proporcionan una calidad visual adecuada para tareas básicas de procesamiento de imágenes o video. Además, tiene varias opciones de conectividad, como WiFi de doble banda, Bluetooth 5.0, puertos USB 3.0 y 2.0, dos puertos HDMI, y un puerto Ethernet RJ45 para garantizar una conexión estable y versátil.

Este mini PC se destaca por su tamaño reducido, con dimensiones de 12.5 x 11.2 x 4.4 cm y un peso de solo 210 g, lo que lo convierte en una opción ideal para el dispositivo de tamaño compacto que se requiere en el proyecto. Además, su capacidad de expansión para incorporar una SSD adicional, le da flexibilidad para ampliar el almacenamiento si es necesario.

El *BMAX B2 Pro* se ha mostrado como una solución eficaz, económica y de alto rendimiento para actuar como el cerebro del dispositivo, gestionando la información de todos los sensores y componentes conectados, y permitiendo su monitoreo y control en tiempo real.



*Figura 3. Detalle de los componentes que incorpora el dispositivo*

## 5.2. Microcontroladores

Los microcontroladores del dispositivo cumplen dos funciones principales: recopilar datos de los sensores (sensor de arco, IMU, y GPS) y servir como controladores de encendido y apagado de otros componentes del sistema. El sistema emplea tres microcontroladores que se pueden observar en la Figura 3, cada uno con responsabilidades específicas:

1. El primer microcontrolador está dedicado exclusivamente a la recolección de datos del sensor de arco eléctrico. Dado que los arcos tienen una duración extremadamente corta, en el rango de milisegundos, es crucial capturar esta información con una alta frecuencia. Por ello, tras varios ensayos, se determinó que un microcontrolador específico se encargaría de esta tarea para evitar la sobrecarga del sistema y asegurar una captura de los eventos de arco. Este microcontrolador tiene una capacidad de muestreo de hasta 4 kHz.

2. El segundo microcontrolador maneja la recolección de datos de la IMU y el GPS, los cuales no requieren una frecuencia de muestreo tan alta como el sensor de arco. Además, este microcontrolador tiene la función adicional de controlar el encendido y apagado del ordenador y de la linterna, permitiendo gestionar estos elementos de manera eficiente sin sobrecargar el sistema principal.

3. El tercer microcontrolador actúa como una puerta de acceso que regula la alimentación de todos los componentes del dispositivo. Permanece encendido en todo momento y es el



encargado de activar o desactivar la tensión del sistema completo. Esta función de control central garantiza que el dispositivo solo consuma energía cuando se necesita realizar un ensayo, manteniendo el sistema en un estado seguro cuando no está en uso.

Los microcontroladores elegidos para este dispositivo son los ESP32, una serie de microcontroladores muy versátiles y eficientes que han ganado popularidad por su potencia, conectividad y costo accesible. El ESP32 es un microcontrolador de doble núcleo, esta capacidad de procesamiento lo hace adecuado para tareas que requieren alta frecuencia de muestreo y recolección de datos en tiempo real, como es el caso del sensor de arco eléctrico que necesita capturar eventos de corta duración.

Entre las principales características del ESP32, destaca su conectividad WiFi integrada y Bluetooth de bajo consumo, lo cual facilita el envío y transmisión de datos sin necesidad de cables, ideal para el diseño compacto y de monitoreo remoto requerido en este proyecto.

El ESP32 también soporta una gran variedad de protocolos de comunicación, como UART, SPI, I2C, y CAN, lo que lo hace compatible con múltiples sensores, incluyendo los de este proyecto, como la IMU y el GPS. Esto simplifica la integración de diversos sensores al dispositivo y asegura que cada microcontrolador puede comunicarse eficazmente con otros componentes.

Otra de las razones para elegir el ESP32 es su facilidad de programación y compatibilidad con el entorno de desarrollo *Arduino IDE*, además de su robusta comunidad de soporte, lo que facilita el proceso de desarrollo y pruebas. Estas características, junto con su bajo consumo energético y tamaño compacto, lo convierten en una opción ideal para aplicaciones de monitoreo en campo, donde se requiere un control autónomo y eficiente.

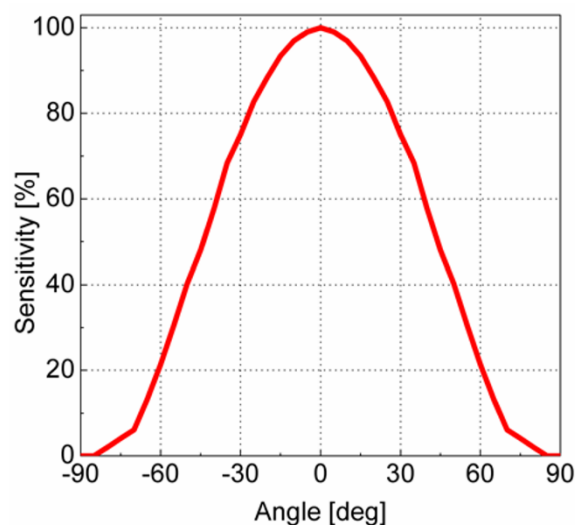
### **5.3. Sensor de Arco**

Los sensores de arco eléctrico ultravioleta están diseñados para detectar arcos eléctricos entre el pantógrafo y el hilo de contacto en sistemas ferroviarios. Este dispositivo es impermeable y utiliza un fotodiodo UV con un filtro especial para captar la radiación ultravioleta que se genera durante un arco, eliminando las interferencias de la luz solar. Esto es posible gracias a que el sensor está tarado para detectar únicamente la frecuencia de radiación que corresponde con el cobre del hilo de contacto. Cuenta con un tiempo de respuesta de 30  $\mu$ s, ideal para la captura de eventos de corta duración. Esto permite identificar y localizar defectos en el hilo de contacto conforme a la norma EN 50317 [6].

La norma EN 50317 establece los procedimientos de medición y evaluación de la interacción mecánica y eléctrica entre pantógrafo y catenaria en sistemas ferroviarios. En el caso específico de la captación de arcos eléctricos, la norma define los métodos para medir la frecuencia, intensidad, y duración de estos arcos, que pueden indicar problemas de contacto entre pantógrafo y catenaria.

En concreto, algunas de las recomendaciones que se indican para la toma correcta de medición de arcos eléctricos se detallan a continuación. Para los hilos de contacto de cobre o aleaciones de cobre que se suelen emplear en las instalaciones, el dispositivo debe de trabajar un espectro de longitud de onda de concreto entre los 220 nm y 329 nm.

Otras recomendaciones son, estar lo suficientemente cerca del pantógrafo para tener una buena sensibilidad, siempre inferior a 5 metros para el caso de la instalación de Metro de Málaga. Estar colocado lo más cerca posible del eje longitudinal del vehículo ya que, como se puede observar en la Figura 4, la sensibilidad del sensor se ve muy afectada si el arco no se produce en la zona central del mismo [7].



**Figura 4.** Sensibilidad del sensor de arco en función del ángulo proporcionada por el fabricante. (Ref. 7)

Por último, apuntar hacia el frotador y colocar el sensor en la parte trasera del pantógrafo según la dirección de la marcha. Este último requisito, no se pudo cumplir para toda la instalación de Metro de Málaga, ya que las unidades disponen de dos cabezas en ambos extremos y no hacen un recorrido cerrado, sino de ida y vuelta. Por tanto, el sensor al ir ubicado en una posición fija, en un viaje de ida el sensor si está ubicado en la posición correcta (en la parte trasera de la dirección de la marcha), mientras que en un viaje de vuelta el sensor no ha cambiado de posición, pero el sentido de la marcha sí. Una solución a esto sería colocar dos

sensores de arcos en ambas ubicaciones, pero tras ver los resultados obtenidos tras los ensayos se pudo comprobar que únicamente existe una pequeña disminución de la intensidad de los arcos, pero se siguen pudiendo monitorizar.

El sensor empleado es un sensor de arco ultravioleta de la marca *sglux* (Figura 5), el cual cumple con las especificaciones requeridas por la normativa EN 50317 para la detección de arcos eléctricos. Este sensor proporciona una salida de voltaje de 0 a 5 V, que puede ser capturada por los microcontroladores instalados, y está calibrado en el rango de longitud de onda adecuado para detectar arcos eléctricos, asegurando así una captura de los eventos UV generados por la interacción pantógrafo-catenaria.



**Figura 5.** Sensor de arco eléctrico UV de la marca *sglux* utilizado en el dispositivo

#### 5.4. Cámara HD

La cámara HD seleccionada en este proyecto cumple la función de validar la información recolectada por el sensor de arco, aportando además detalles adicionales sobre el estado de la infraestructura. Se ha optado por una cámara de alta definición para asegurar imágenes claras y detalladas, esenciales para observar el contacto pantógrafo-catenaria.

La cámara empleada es el módulo de cámara USB de 12MP 477P con lente CS de 6mm de *Arducam* que se muestra en la Figura 6. Este modelo cuenta con un sensor de imagen *Sony 477P* de 12 megapíxeles, proporcionando alta calidad y una instalación sencilla dentro del dispositivo diseñado. La elección de una lente adecuada ha sido crucial para cubrir el rango de movimiento

del pantógrafo a lo largo de la instalación, ya que en ciertas áreas se extiende completamente, mientras que en otras se encuentra casi contraído. Esto asegura que se capture de forma nítida y el comportamiento completo del sistema, sin importar su posición en la infraestructura.



**Figura 6.** Cámara HD de la marca Arducam utilizada en el dispositivo

Tras unos primeros ensayos, se observó la que la iluminación era insuficiente para poder obtener videos de calidad, por tanto, en los siguientes ensayos se incorporó al sistema de cámaras una linterna (Figura 7) para facilitar y mejorar la calidad de los videos.



**Figura 7.** Linterna incorporada al dispositivo para mejorar la visibilidad



### **5.5. Unidad de medición inercial (IMU)**

Para medir las aceleraciones de la unidad que transporta el dispositivo, se ha integrado un sensor inercial de bajo coste. El propósito principal de este sensor es obtener datos de los momentos clave de arranque y parada de la unidad, puntos en los que la demanda de energía entre el pantógrafo y la catenaria se incrementa considerablemente. Estos momentos son críticos, ya que permiten observar cómo afecta la interacción de energía y ayudan a identificar patrones específicos en los datos. Idealmente, sería beneficioso contar también con sensores de voltaje e intensidad para complementar la información sobre el impacto de los arcos eléctricos y obtener una evaluación completa de la calidad de la energía suministrada.

El sensor inercial elegido es el MPU-6050, que aparece en la Figura 3, que es altamente compatible con los microcontroladores instalados. Esto facilita tanto su implementación como su programación, además, ofrece una tolerancia adecuada para las tareas que va a desempeñar. El MPU-6050 mide aceleración y rotación, y es ampliamente usado en aplicaciones donde se requiere un equilibrio entre bajo costo y buen rendimiento. Su integración permite registrar no solo la aceleración sino también la inclinación de la vía, lo cual puede ser útil para analizar factores que influyen en el comportamiento del pantógrafo.

Otra característica importante es la capacidad del sensor de proporcionar datos sobre la temperatura ambiental. Esto resulta especialmente relevante en Málaga, donde las temperaturas elevadas en verano pueden causar sobrecalentamiento en el dispositivo, ya que está expuesto en la parte superior de la unidad. El monitoreo de la temperatura ayuda a identificar cualquier riesgo térmico y a tomar medidas adecuadas para proteger la funcionalidad del sistema en condiciones adversas.

### **5.6. Sistema de posicionamiento global (GPS)**

Para la localización de la unidad, se ha optado por utilizar un módulo GPS, el cual envía las coordenadas exactas de la ubicación de la unidad siempre que haya cobertura satélite suficiente. Este sensor debe cumplir con las mismas características que el sensor IMU, es decir, debe ser fácil de programar, integrar en el sistema y tener un costo bajo. El principal inconveniente de este sensor es que, cuando la unidad se encuentra en las secciones subterráneas de la instalación, no se obtiene información sobre su ubicación debido a la falta de señal satelital.



Por lo tanto, el GPS no se utiliza para localización en tiempo real, sino más bien para realizar un seguimiento del recorrido de la unidad, ya que la Línea 1 del metro de Málaga sale a la superficie, mientras que la Línea 2 permanece siempre en infraestructura subterránea. Al combinar estos datos de ubicación con los datos de aceleración del sensor IMU, se puede identificar qué línea está recorriendo la unidad en un momento dado. En el futuro, se planea realizar un estudio más detallado para poder contabilizar el número de pasadas que realiza cada unidad y las instalaciones en las que se encuentra. Este parámetro es crucial para alimentar adecuadamente los modelos de desgaste en los que se está trabajando

El GPS empleado en el dispositivo es el modelo GPS Air 530 de la marca *Grove-Seeed Studio* (Figura 3), un módulo de alta integración y alto rendimiento diseñado para aplicaciones de posicionamiento. Este módulo soporta múltiples sistemas de navegación satelital, incluidos GPS, Beidou, Glonass, Galileo, QZSS y SBAS, lo que lo hace adecuado para una amplia variedad de aplicaciones de posicionamiento. El GPS Air 530 puede recibir señales de más de seis satélites simultáneamente, lo que le permite trabajar de manera rápida incluso en condiciones de señal débil. Además, su consumo de energía es muy bajo, con solo  $31\mu\text{A}$ , y cuenta con interfaces de hardware y software compatibles para su integración con otros módulos. Su margen de error en el posicionamiento es de solo 10 metros, lo que asegura una alta exactitud en la localización.

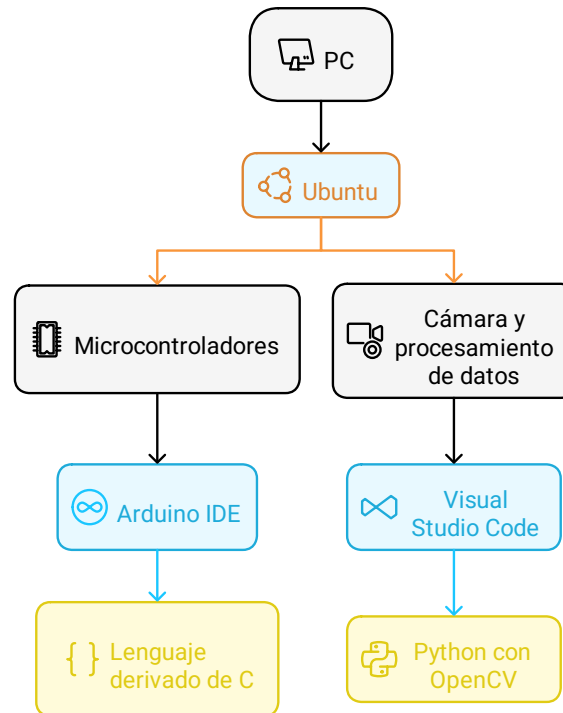


## 6. PROGRAMACIÓN

En ese apartado, se tiene como objetivo detallar los aspectos clave relacionados con los códigos utilizados en el desarrollo del dispositivo. A través de la programación, se logra gestionar el funcionamiento de los diferentes componentes que componen el sistema, tales como los microcontroladores, sensores y la comunicación inalámbrica. Además, se abordan las estructuras y lógicas de programación empleadas para garantizar la correcta recolección de datos y su transmisión en tiempo real, así como el control de encendido y apagado de los diferentes elementos del dispositivo. En este capítulo se explicarán los enfoques adoptados y las decisiones técnicas que permitieron integrar los distintos módulos de forma eficiente y coherente dentro del sistema.

### 6.1. Preparación del PC

Dentro de la programación del PC, se destacan dos aspectos clave. El primero es la elección del sistema operativo, en este caso *Ubuntu*, que se seleccionó debido a su estabilidad, facilidad de uso en entornos de desarrollo y compatibilidad con los diferentes programas y herramientas necesarias para la gestión del dispositivo. El segundo aspecto es la programación del código, que se ha dividido en dos áreas. Para el desarrollo general, se ha empleado el lenguaje *Python* junto con el editor *Visual Studio Code*, utilizado para la captura y almacenamiento de datos. En cuanto a los microcontroladores, se utilizó el editor *Arduino IDE* y su propio lenguaje, basado en *C*, para la programación de los sensores y la configuración de los distintos componentes. La Figura 8 muestra un esquema tanto del sistema operativo empleado como de las herramientas y lenguajes de programación empleados en el dispositivo.



**Figura 8.** Esquema del sistema operativo y herramientas de programación utilizadas

### 6.1.1. Sistema operativo

El sistema operativo seleccionado para el ordenador central es *Ubuntu 24.04.1 LTS*, elegido principalmente por su eficiencia en el uso de recursos y por minimizar las tareas en segundo plano. En las primeras versiones, se usó *Windows 10*, pero surgieron problemas al reconectar el dispositivo tras largos períodos de inactividad, ya que *Windows* realizaba actualizaciones automáticas y procesos innecesarios que ralentizaban el rendimiento. *Ubuntu*, en cambio, mantiene un uso mínimo de recursos tras una desconexión prolongada, permitiendo un rendimiento estable y eficiente.

Además, *Ubuntu* es un software libre con una sólida seguridad, amplia personalización, y una comunidad activa que proporciona soporte. La flexibilidad y estabilidad del sistema hacen que sea ideal para entornos de prueba en dispositivos embebidos, garantizando un entorno optimizado y libre de interferencias externas.

### 6.1.2. Arduino IDE

Para la programación de los microcontroladores ESP32 se ha utilizado el software *Arduino IDE 1.8.19*, el cual es ampliamente empleado para programar distintos tipos de placas de



microcontroladores. Este entorno de desarrollo utiliza un lenguaje derivado de C, que facilita la escritura de código compatible con las ESP32. Sin embargo, para utilizarlo correctamente en estas placas, es necesario instalar el gestor de tarjetas ESP32, ya que el IDE de *Arduino* por defecto solo incluye compatibilidad con placas *Arduino*.

Una vez realizado este sencillo proceso de instalación, el programa queda listo para compilar y cargar el código en los microcontroladores ESP32. *Arduino IDE* también incluye una función para monitorizar la comunicación serial entre la placa y el dispositivo al que está conectada, lo cual es esencial para la depuración de código y la verificación de la correcta ejecución de los programas. Esta comunicación serial actúa como el principal medio para identificar y solucionar errores en los programas desarrollados.

### **6.1.3. Python con Visual Studio Code**

El programa general ha sido desarrollado utilizando *Python 3.11.0* junto con la librería *OpenCV*. *Python* ha sido elegido debido a su versatilidad, su amplia gama de librerías, y la disponibilidad de abundante información, lo cual facilita la creación de programas de alta calidad y optimización en diferentes aplicaciones.

*Python*, con la librería *OpenCV*, permite realizar procesamiento de imágenes y videos mediante funciones avanzadas, ideales para capturar, manipular y analizar contenido visual en tiempo real. *OpenCV* en *Python* es una opción potente por su capacidad para manejar múltiples formatos de imágenes y videos, permitiendo filtros, detección de bordes, seguimiento de objetos, entre otras aplicaciones de visión artificial. Esto lo convierte en una excelente herramienta para desarrollos en sistemas de monitoreo visual o en aplicaciones de control de calidad visual.

Además, se ha utilizado *Visual Studio Code* como editor principal. Este entorno es un editor de código fuente, ampliamente utilizado por su interfaz intuitiva y soporte de diversas extensiones. Integra características avanzadas como autocompletado, depuración en tiempo real, y un amplio catálogo de extensiones, que facilitan el trabajo con múltiples lenguajes de programación, incluyendo *Python*. En este caso, *Visual Studio Code* permite una experiencia de desarrollo más fluida, ofreciendo resaltado de sintaxis, administración de paquetes y un terminal integrado que facilita la ejecución y revisión del código de manera eficiente.



## 6.2. Programa general

Este programa es el que está desarrollado en *Python* para captura las imágenes de la cámara mediante la biblioteca de *OpenCV*, recibir los datos de los sensores por medio de los puertos series de los microcontroladores y almacenarlos en un archivo con el nombre del ensayo.

El programa se organiza en varias secciones que realizan funciones específicas para grabar video, leer datos de los sensores conectados a través de puertos seriales, y mostrar esta información en tiempo real mediante una interfaz gráfica. De forma esquemática la Figura 9 muestra el flujo de trabajo del programa para una mejor comprensión.

1. **Configuración inicial:** Al comenzar, el programa carga una serie de bibliotecas y establece parámetros clave. Entre estos parámetros están las rutas de archivos donde se guardarán los datos grabados y las configuraciones de video, como la velocidad de fotogramas y el tamaño de estos. También se inicializan variables para controlar el estado de grabación, los puertos seriales y otras funciones.
2. **Gestión de fotogramas:** Se crea una clase específica para gestionar la captura y almacenamiento de video. Esta clase permite que los fotogramas se capturen y almacenen en un buffer antes de ser procesados y escritos en el archivo de video, asegurando que los fotogramas se mantengan a una frecuencia establecida, minimizando pérdidas y manteniendo la sincronización.
3. **Ajuste y configuración de cámaras:** Se configura una cámara para capturar imágenes en un formato y resolución específicos. La cámara establece parámetros que permiten captar video de alta calidad, adecuado para análisis detallados de objetos en movimiento como el pantógrafo en este caso. Si ocurre un problema al iniciar la cámara, el programa lo detecta e imprime un mensaje de error.
4. **Interacción con el puerto serial:** El programa contiene dos funciones principales para leer datos desde dos puertos seriales diferentes, que corresponde con los dos microcontroladores que reciben la información de los sensores externos. Cada función lee de forma continua los datos recibidos en el puerto respectivo y, si está activada la grabación, estos datos se guardan en archivos separados para su posterior análisis.
5. **Ejecutando hilos paralelos:** Cada puerto serial se gestiona en un hilo paralelo. Esto permite que el programa lea datos de ambos puertos simultáneamente sin que una tarea interfiera



con la otra. Los hilos aseguran que el flujo de datos de cada sensor se procese en tiempo real, incluso mientras se está grabando el video.

6. **Interfaz gráfica:** Para permitir el control del programa en tiempo real, se usa una interfaz gráfica que permite al usuario iniciar y detener la grabación de video mediante botones. También se incluye un indicador visual que muestra el estado de grabación, lo cual facilita el monitoreo y control de las actividades de captura de datos y video.
7. **Procesamiento de video en tiempo real:** A medida que la cámara captura fotogramas, el programa añade información en la pantalla del video, como la hora actual, y ajusta el color y posición del texto para que sea legible en diferentes áreas del video. Esto asegura que los datos de tiempo sean visibles en el video grabado, proporcionando un contexto valioso para cada fotograma grabado.
8. **Generación de nombres de archivos:** Cada vez que se inicia una nueva sesión de grabación, el programa genera nombres únicos para los archivos de texto y video. Estos nombres incluyen la fecha y hora actuales, lo cual facilita la organización y acceso a los archivos creados.
9. **Ejecución del programa principal:** En el bloque principal, el programa inicializa la interfaz gráfica y configura los puertos seriales para empezar a recibir datos. También configura la cámara y, en un bucle continuo, responde a las interacciones del usuario en la interfaz gráfica. La grabación se inicia y detiene mediante la interfaz, controlando el flujo de fotogramas y datos que se guardan en los archivos.
10. **Terminación y limpieza:** Cuando el usuario cierra la interfaz gráfica o detiene el programa, este se asegura de cerrar los puertos seriales, detener los hilos paralelos, liberar los recursos de la cámara y cerrar las ventanas abiertas. De esta manera, se libera toda la memoria utilizada y se previenen posibles errores en futuras ejecuciones del programa.

Este desglose ilustra cómo el programa organiza y coordina la captura de video, la lectura de datos seriales, y la interacción con el usuario en un flujo continuo y estructurado. Cada componente del código está diseñado para integrarse de manera que mantenga el programa en un estado estable y listo para recopilar y almacenar datos importantes de forma eficiente.

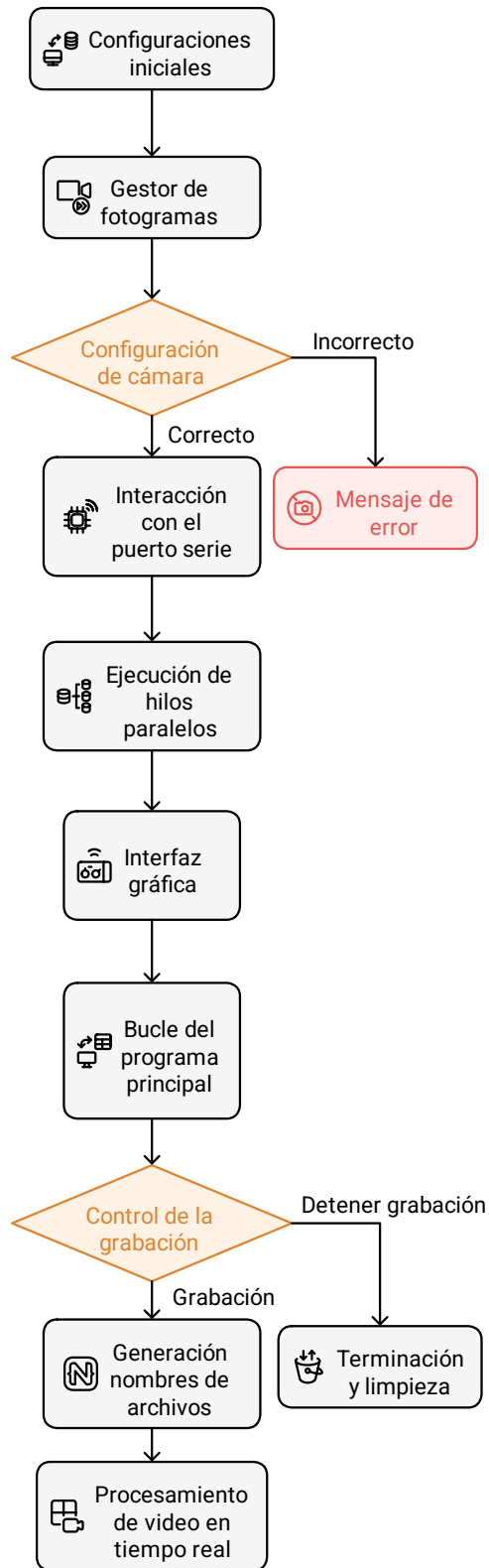


Figura 9. Esquema de flujo de trabajo del programa general

### 6.3. Programa de captura de arcos

El siguiente programa es el de captura de arcos, la peculiaridad que tiene este programa es que se desea captura a una muy alta frecuencia para poder tener con detalle cada uno de los arcos que se producen en la instalación, en concreto queremos trabajar en los 4 KHz. Para ello se han llevado a cabo distintas pruebas tanto con el sensor como con el programa, para evaluar si las repuesta que se obtiene es la esperada.

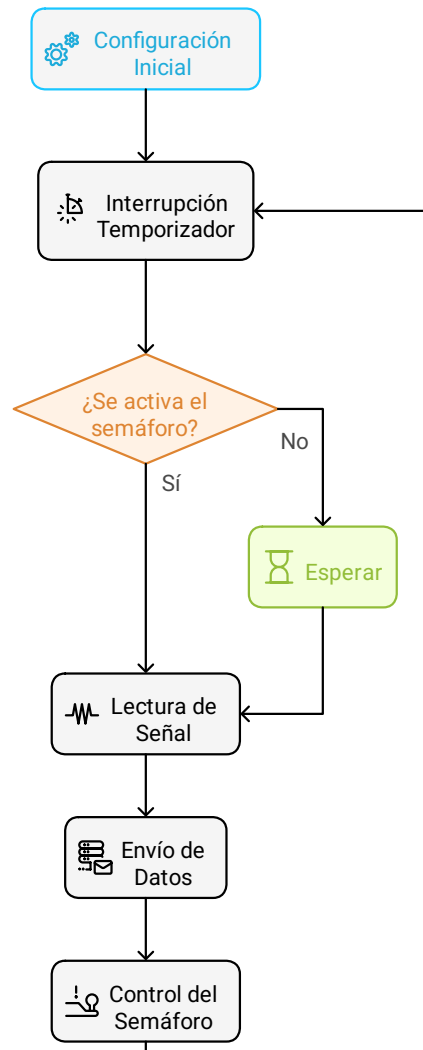
Este programa para un microcontrolador ESP32 utiliza un temporizador hardware para medir la frecuencia de muestreo de un valor analógico y enviar estos datos a la interfaz serie. De forma esquemática la Figura 10 representa el flujo de trabajo del programa para una mejor comprensión.

**1. Configuración inicial:** Primero, se establece la comunicación serie con una alta tasa de transmisión para una rápida transferencia de datos. A continuación, el temporizador se configura con la frecuencia específica de muestreo que se desea realizar el ensayo. Se configura el temporizador para que ajuste la frecuencia base de 80 MHz para sincronizarla con el intervalo deseado, lo cual también asegura que el temporizador actúe de forma periódica y continua.

**2. Interrupción del temporizador:** Se utiliza una rutina de servicio de interrupción (ISR) que se activa cada vez que el temporizador alcanza el intervalo deseado. Dentro de esta ISR, se incrementa un contador para llevar registro de las interrupciones y se activa una señal de semáforo que permite al bucle principal saber cuándo realizar la tarea de lectura del valor analógico.

**3. Bucle principal:** En el bucle principal, se comprueba el estado de la señal generada en la ISR. Si la señal está activada, se realiza la lectura del voltaje en el pin analógico especificado, que registra el estado del sensor o el valor de un circuito conectado. Luego, esta lectura se envía al puerto serie, junto con el contador de la ISR que permite observar la exactitud de las lecturas.

**4. Control del semáforo:** Después de realizar la lectura y el envío de datos, el programa apaga la señal de semáforo, preparándose para la siguiente interrupción. Esto garantiza que las lecturas analógicas ocurran solo cuando la ISR lo permite, evitando lecturas innecesarias y manteniendo la frecuencia de muestreo controlada.



**Figura 10.** Esquema del flujo de trabajo del programa que captura los arcos eléctricos

En resumen, este código permite que el ESP32 capture datos analógicos de manera regular y eficiente, enviándolos por el puerto serie en intervalos definidos, mientras que el temporizador asegura que estas lecturas se realicen con una frecuencia controlada y constante.

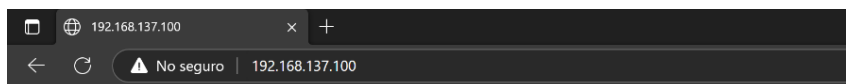
#### 6.4. Programa de captura de IMU y GPS

El siguiente código es el encargado de la recogida de los datos proporcionados por el IMU y el GPS. Este código utiliza el mismo sistema de temporización que se emplea para la captura de los arcos, sin embargo, se emplea a una frecuencia mucho menor. Por lo general, los datos se recogen a una frecuencia de 20 Hz. Además, este programa al ser mucho más ligero permite realizar cosas en paralelo sin que existan problemas en la recogida de datos. De esta forma, el

programa crea un servidor web a la que se accede durante los ensayos y que permite controlar el apagado y encendido tanto del PC como de la linterna. A continuación, se detalla el funcionamiento del programa en detalle.

**1. Conexión y red:** El dispositivo ESP32 se conecta a una red Wi-Fi usando una IP fija. Esto facilita que otros dispositivos puedan acceder a él directamente, siempre en la misma dirección de red mediante una conexión local. Una vez conectado, el ESP32 establece un servidor web, que es una página en internet. Este servidor funciona en el puerto fijo, donde los usuarios pueden enviar comandos para interactuar con el dispositivo, en concreto controlar el PC y la linterna.

**2. Página web de control:** La página web que el ESP32 presenta, tiene botones que permiten al usuario activar o desactivar dos relés conectados al dispositivo como se muestra en la Figura 11. Cuando el usuario hace clic en un botón, el ESP32 recibe el comando, cambia el estado del relé correspondiente (enciende o apaga) y envía una respuesta visual al usuario. Además, el relé dedicado a la linterna tiene programado en código un temporizador para apagarse automáticamente después de un corto período, imitando una señal de pulsación breve, ya que es la forma que tiene la linterna de apagarse y encenderse.



## Controles dentro de la caja

Encender ordenador:

Encender linterna:

*Figura 11. Interfaz gráfica para el control de los elementos del dispositivo*

### 3. Sensores y recolección de datos:

**Sensor de Arco Eléctrico:** Este sensor mide la intensidad de los arcos eléctricos a través de una señal de voltaje que se recoge mediante uno de los pines del microcontrolador. El ESP32 convierte esta señal de voltaje en valores que puede registrar y analizar.

**IMU:** Este sensor mide aceleración y rotación en tres direcciones, capturando el movimiento del dispositivo. Con estos datos, se calcula el ángulo de inclinación, lo que permite



analizar el balance y movimiento del sistema. La IMU también proporciona datos de temperatura. Todos estos datos son registrados por parte del microcontrolador

**GPS:** El módulo GPS proporciona la ubicación en coordenadas dadas en latitud y longitud, altura, la hora en formato UTC, el número de satélites a los que está conectado el GPS, la calidad de la señal y la velocidad de desplazamiento. Estos datos se actualizan cada segundo, permitiendo al ESP32 rastrear la ubicación del dispositivo en tiempo real.

**4. Procesamiento de datos del IMU:** La aceleración y la rotación que la IMU mide son combinadas para calcular a posteriori los ángulos de inclinación. Para ello, se emplea un filtro complementario que utiliza la aceleración para las mediciones iniciales y los datos de rotación para actualizar el ángulo.

**5. Alarmas y registro de datos:** El ESP32 está configurado para activar su sistema de recopilación de datos en intervalos regulares, gracias a una alarma que se activa periódicamente. Cada vez que se activa, el ESP32 toma los datos más recientes de todos los sensores y los imprime en el puerto serie, una conexión que permite observar en una pantalla de computadora todo lo que está ocurriendo. En el registro, cada entrada incluye la hora, coordenadas GPS, temperatura y los ángulos de inclinación.

**6. Ejecución continua y respuestas:** Mientras el ESP32 recopila y registra los datos, el servidor web permanece disponible, permitiendo que el usuario controle los relés en cualquier momento. Esta estructura permite la supervisión continua del estado de los sensores y el control remoto del dispositivo desde la web.

En resumen, el ESP32 realiza dos tareas simultáneamente: recopilar y procesar datos de los sensores y ofrecer una interfaz de control mediante el servidor web. De una forma simplificada la Figura 12 y Figura 13 muestran los flujos de trabajos que hay programados en este microcontrolador. Este sistema permite supervisar el estado del dispositivo en tiempo real, así como controlar ciertos aspectos de este a través de la red.

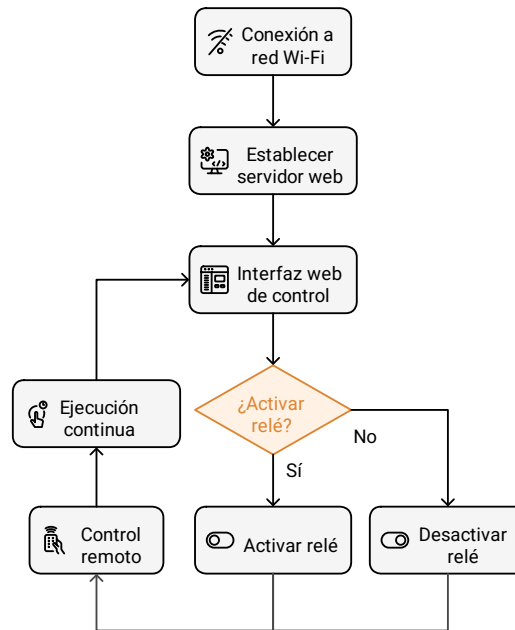


Figura 12. Esquema del flujo de trabajo del control de la página web

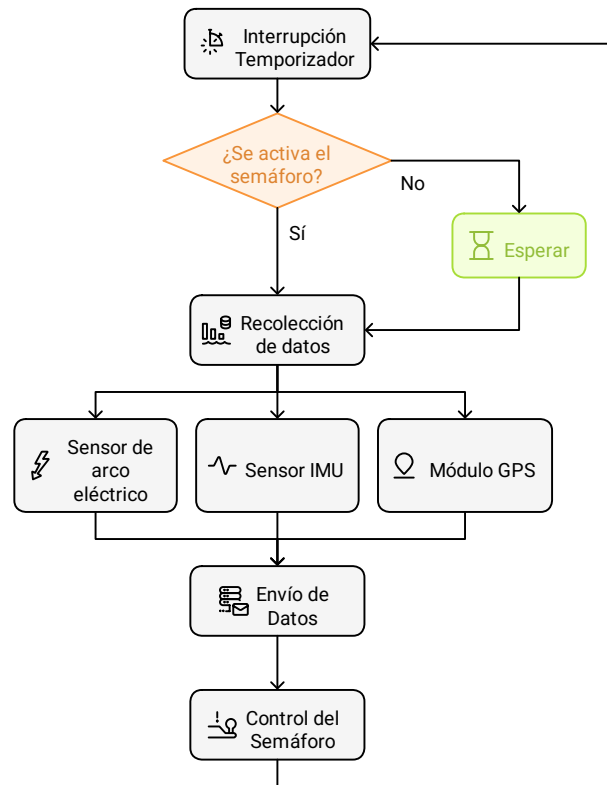


Figura 13. Esquema del flujo de trabajo de la captura los datos de los sensores

## 6.5. Programa de puerta de acceso al dispositivo

Este programa utiliza el ESP32 para crear un sistema de control de un relé a través de una interfaz web accesible mediante una red Wi-Fi. Este microcontrolador es el que actúa de puerta de acceso principal al dispositivo, es por tanto el dispositivo que siempre está encendido esperando a que le llegue las instrucciones correctas para conectarse. A continuación, se detalla el funcionamiento paso a paso:

**1. Definición del entorno y variables de red:** El ESP32 se configura para conectarse a una red Wi-Fi específica con una dirección IP fija, configurada manualmente. Esta IP, junto con los datos de puerta de enlace y la máscara de subred, permite que el dispositivo esté accesible de manera remota en la misma red local. La configuración fija garantiza que el dispositivo siempre tenga la misma IP.

**2. Inicialización del servidor web:** En el programa, se inicia un servidor web en el puerto estándar. Este servidor se encarga de manejar las solicitudes de los usuarios que interactúan con la interfaz web. La IP asignada y el puerto serán los puntos de acceso para que cualquier navegador en la misma red pueda interactuar con el ESP32.

**3. Interfaz básica para el control de relé:** Al acceder a la IP del ESP32 en un navegador, el servidor web muestra una interfaz sencilla compuesta de un título y dos botones de encendido o apagado (Figura 14), que permiten controlar el estado del relé. Esto se logra generando una página desde el ESP32, donde cada botón está vinculado de forma específica para que se envíe la orden de encender o apagar el relé.



*Figura 14. Interfaz gráfica para la puerta de acceso al dispositivo*

**4. Manejo de solicitudes del usuario para encender o apagar el relé:** Cuando el usuario presiona un botón en la página web, se envía una solicitud a una URL específica. El servidor web recibe esta solicitud y la interpreta para saber qué acción realizar.



**Encendido:** Si el servidor detecta que la solicitud es para encender el relé, activa el pin de salida correspondiente, configurado previamente en el programa, lo que permite el flujo de corriente para encender el dispositivo completo.

**Apagado:** En el caso de una solicitud de apagado, el servidor corta la corriente en el pin de salida, haciendo que todo el dispositivo se quede sin tensión.

**5. Redirección a la Página Principal Después de Cada Acción:** Una vez que el ESP32 procesa la solicitud y cambia el estado del relé, envía una respuesta de redirección para que el usuario regrese automáticamente a la página principal en su navegador. Así, el usuario siempre vuelve a la interfaz de control principal tras cada acción.

**6. Bucle Principal de Ejecución:** En el ciclo principal del programa, el ESP32 está en constante espera de solicitudes de los usuarios. Cada vez que detecta una nueva solicitud, la procesa inmediatamente a través del servidor web, ejecutando la acción indicada ya sea apagar o encender el sistema.

De esta manera, el programa permite controlar el encendido del dispositivo de manera remota desde cualquier navegador en la red local, utilizando una interfaz web sencilla. El uso de IP fija asegura que el dispositivo sea siempre accesible en la misma dirección, y el servidor web en el ESP32 se encarga de gestionar la interacción con el usuario.

### 6.6. Control remoto

Actualmente, el control remoto del dispositivo se realiza mediante la aplicación de escritorio remoto de Chrome. Para configurarla, es necesario instalarla tanto en el PC del dispositivo como en el PC que se usará para la conexión. Después de completar la instalación, se realiza una primera conexión para establecer las credenciales de seguridad, lo que permite que, siempre que ambos equipos estén conectados a una red WiFi con acceso a internet, se puedan enlazar de manera remota de forma sencilla y sin complicaciones.

Se ha optado por este método de control remoto, en lugar de una conexión local, debido a las incompatibilidades entre los sistemas operativos de ambos ordenadores. Mientras el dispositivo ejecuta *Ubuntu*, el ordenador de conexión utiliza *Windows 11*. Tras probar varios métodos de conexión local y enfrentar numerosos problemas, se decidió emplear una aplicación de acceso remoto a través de internet como *TeamViewer*, *AnyDesk* o *Chrome Remote Desktop*. Tras comparar varias de estas alternativas, *Chrome Remote Desktop* demostró ser la más estable y



---

fiable, permitiendo conexiones fluidas y sin inconvenientes, lo cual ha facilitado significativamente la realización de los ensayos.

Por otro lado, la transferencia de archivos desde el dispositivo al ordenador requiere una solución adicional, ya que *Chrome Remote Desktop* limita el tamaño de los archivos a un máximo de 500 MB. Dado que los archivos de video de los ensayos suelen exceder este límite, se ha implementado una cuenta de *Google Drive* en el dispositivo, donde se cargan directamente todos los archivos de los ensayos. Esta medida permite almacenar los datos en la nube, desde donde pueden descargarse en el equipo deseado, facilitando el acceso y la gestión de los archivos de mayor tamaño.



## 7. FABRICACIÓN

Se han empleado diferentes técnicas de fabricación dependiendo del tipo de componente a construir. Estas se dividen en dos bloques principales:

**Fabricación mecánica:** Incluye la creación de la estructura principal del dispositivo, los elementos de anclaje con la unidad y los elementos auxiliares. Estos últimos son necesarios para asegurar la correcta sujeción de los sensores y otros componentes del sistema. Para la fabricación del dispositivo se han empleado tanto técnicas tradicionales como el plegado y soldado de chapa como técnicas más actuales como la impresión 3D.

**Conexiones eléctricas y cableado:** Comprende el diseño y desarrollo del esquema eléctrico de alimentación del dispositivo, así como la elaboración de placas de conexionado similares a una PCB, pero realizadas a mano. Estas placas se han fabricado para facilitar el cableado de los distintos sensores, permitiendo su comunicación eficiente con los microcontroladores y asegurando una integración robusta entre los componentes eléctricos y electrónicos del sistema.

### 7.1. Fabricación mecánica

Este apartado a su vez se puede subdividir en dos secciones diferenciadas, cada una centrada en aspectos clave de la fabricación mecánica del dispositivo. Por una parte, la fabricación del chasis y anclajes con la unidad y, por otra parte, la fabricación de componentes auxiliares. Ambas secciones son fundamentales para el ensamblaje completo del dispositivo, asegurando tanto su robustez mecánica como la correcta disposición de todos sus elementos funcionales.

#### 7.1.1. Chasis del dispositivo

En esta sección se aborda la construcción del chasis del dispositivo, diseñado para proporcionar estabilidad y soporte estructural. También incluye los anclajes necesarios para asegurar una fijación firme del dispositivo a la unidad donde será instalado, garantizando su funcionalidad en condiciones de operación reales.

El diseño inicial del dispositivo parte de una idea clave: minimizar sus dimensiones, especialmente la altura. Este enfoque busca hacerlo discreto y evitar riesgos de colisión con elementos situados por encima de la unidad. Además, se prioriza la protección frente a condiciones climáticas adversas, como lluvia o calor extremo. Para ello, los sensores sin certificación de resistencia al agua se alojan dentro del dispositivo, mientras que los resistentes,

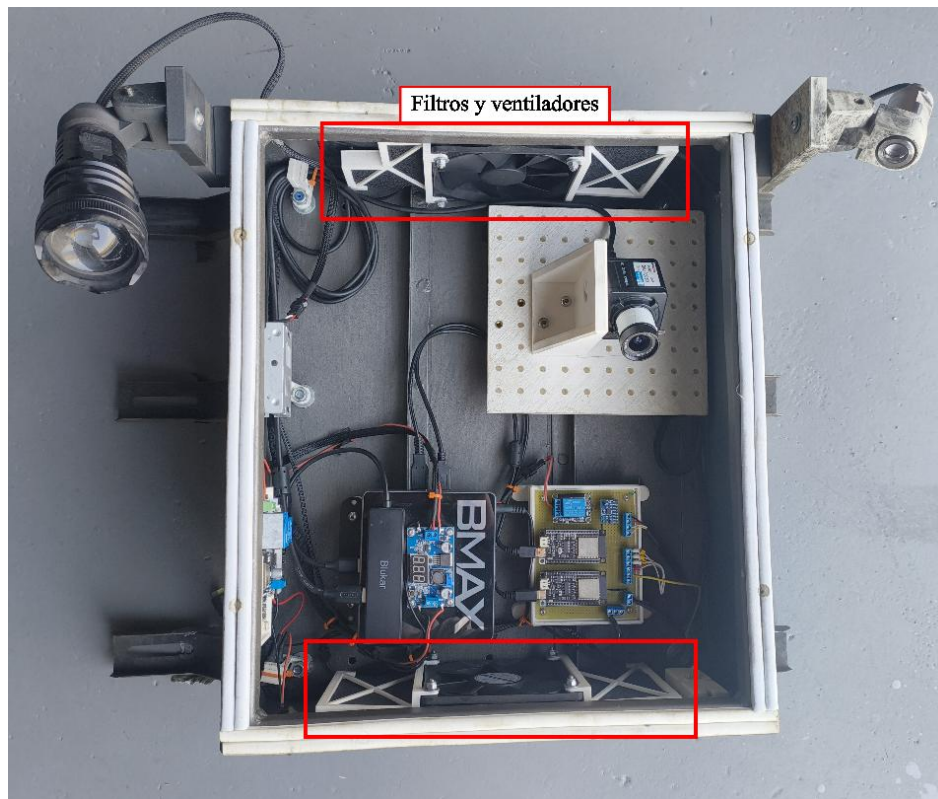
como la linterna y el sensor de arco, quedan expuestos. Además, el sensor de arco necesita ser colocado en el exterior debido a que cualquier tipo de pantalla delante del sensor actúa como filtro para las longitudes de onda del rango del espectro electromagnético en el que trabaja. La Figura 15 muestra en detalle el dispositivo donde se pueden observar estas características.



**Figura 15.** *Dispositivo embarcado en detalle*

Un caso particular es la cámara de video, ubicada dentro del dispositivo. Para asegurar su funcionalidad, se ha fabricado un cristal a medida que permite una visión óptima mientras protege el equipo y facilita una inspección visual del estado general del dispositivo. En los primeros ensayos esta tapa era de policarbonato, pero debido a que se rayaba con gran facilidad tras su limpieza, apareciendo marcas en los vídeos. Por ello, se sustituyó el policarbonato por el vidrio templado para mejorar la calidad de los videos.

Por otro lado, la interacción pantógrafo-catenaria genera carbonilla que podría afectar el funcionamiento interno. Para prevenir esto, se han instalado filtros que se observan en la Figura 16 en los conductos de ventilación que limitan la entrada de partículas. Estas partículas al ser de grafito conducen bien la electricidad por lo tanto hay que evitar que entren y se pongan en contacto con cualquier tipo de conexión eléctrica.

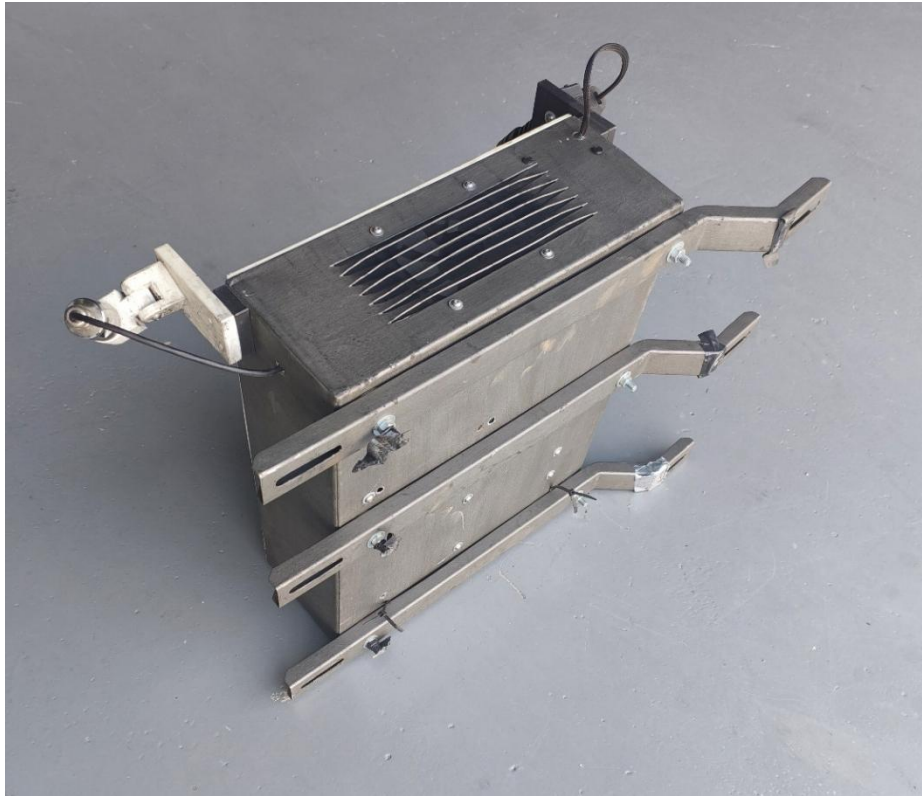


**Figura 16.** Vista en planta del dispositivo

En cuanto a la instalación, los anclajes diseñados, basados en información proporcionada por Metro de Málaga, incluyen tolerancias para adaptarse a ligeras diferencias entre las unidades. Estos anclajes incorporan acoplamientos elásticos que absorben vibraciones, minimizando su impacto en las grabaciones realizadas y garantizando la estabilidad operativa del dispositivo. Estos anclajes se muestran con más detalle en la Figura 17.

El proceso de fabricación del chasis comienza con el corte de la chapa, una operación fundamental para dar forma inicial al material. Se utilizó maquinaria especializada, en el caso de este dispositivo una prensa de corte para chapas. El corte asegura que las piezas tengan dimensiones exactas, ajustándose a los planos de diseño.

Posteriormente, se realizó el plegado de la chapa para crear las estructuras tridimensionales necesarias. Este proceso se llevó a cabo en una plegadora, una máquina que aplica presión controlada sobre la chapa para doblarla en ángulos específicos. Durante esta etapa, es crucial alinear correctamente el material para evitar deformaciones y garantizar que las dimensiones finales sean exactas. Además de tener claro el orden de los procesos para obtener el resultado final que se desea.



*Figura 17. Detalle de los anclajes que sujetan el dispositivo con la unidad*

Una vez que se obtuvo la forma deseada, se procedió a la soldadura, uniendo las distintas piezas para dotarlas de rigidez y solidez estructural. Estas técnicas ofrecen gran resistencia y calidad en uniones metálicas. Durante la soldadura, es esencial mantener una temperatura controlada y una alineación perfecta de las piezas para evitar distorsiones debidos a los cambios de temperatura.

Por último, el chasis se sometió a un proceso de pintura protectora para prevenir la corrosión y mejorar su durabilidad. Antes de pintar, la superficie se limpió y se trató con imprimación anticorrosivas, asegurando que la pintura se adhiriera de manera uniforme y proporcione una protección adecuada contra el óxido. Este acabado también aporta un aspecto estético y profesional al chasis terminado.

### **7.1.2. Fabricación de componentes auxiliares**

Se refiere a la elaboración de piezas de menor tamaño, pero de igual importancia. Estas piezas están diseñadas para facilitar la colocación de los diversos elementos y sensores dentro del entorno del dispositivo, contribuyendo a una integración ordenada y eficiente de los componentes.



Las piezas diseñadas para el dispositivo fueron creadas utilizando software especializado en modelado 3D, permitiendo adaptarlas específicamente a los elementos de la caja y asignarles ubicaciones concretas dentro del sistema. La fabricación se llevó a cabo mediante impresión 3D, utilizando materiales como ABS y nylon reforzado con policarbonato. Ambos materiales destacan por sus propiedades técnicas: el ABS es conocido por su alta resistencia al impacto, durabilidad y tolerancia a temperaturas moderadamente altas, mientras que el nylon con policarbonato proporciona una excelente resistencia mecánica, estabilidad térmica y durabilidad en entornos exigentes.

Para integrar estas piezas de manera firme al dispositivo, se colocaron insertos metálicos en las partes plásticas. Estos insertos ofrecen una rosca metálica que permite fijarlas al chasis mediante tornillos, asegurando una unión rígida y duradera.

Entre los componentes más relevantes fabricados con este método se encuentran los soportes para el sensor de arco, la linterna, y la cámara, así como una plataforma ajustable para esta última. Los soportes del sensor de arco y la linterna son prácticamente idénticos, diseñados no solo para sostener los dispositivos, sino también para permitir su ajuste, asegurando que apunten correctamente a las zonas de interés. Estos soportes se ubican en el exterior del dispositivo, aprovechando las certificaciones de resistencia al agua que poseen.

El soporte y la plataforma de la cámara, en particular, tienen un diseño especial. Su objetivo es garantizar que el campo de visión de la cámara abarque todo el rango de movimiento del pantógrafo. Dado que las unidades ferroviarias presentan pequeñas variaciones entre sí, la plataforma fue diseñada inicialmente con un rango amplio de ajuste para adaptarse a estas diferencias. Aunque en las primeras fases se desconocían los márgenes exactos necesarios, hoy se dispone de información suficiente para reducir estos ajustes a parámetros más específicos, optimizando el diseño para hacerlo más compacto y eficiente.

Además de los soportes previamente mencionados, la técnica de impresión 3D también se ha utilizado para fabricar otros elementos esenciales del dispositivo. Entre estos se encuentran los soportes para las placas de conexión, que garantizan una distribución ordenada y segura del cableado; el soporte destinado al PC, diseñado para proteger y fijar adecuadamente el equipo principal; y los soportes para los relés de corriente, necesarios para el control eléctrico del sistema.

Asimismo, se crearon soportes específicos para alojar el módulo GPS en una posición estratégica que garantice una óptima recepción de la señal. Por último, también se diseñaron y fabricaron los anclajes que fijan los filtros de aire al dispositivo.

## 7.2. Conexiones eléctricas y cableado

En este apartado se detalla el proceso llevado a cabo para el conexionado eléctrico de los componentes, abarcando tanto la alimentación como la transmisión de datos.

El sistema de alimentación se inicia con la tensión proporcionada por la unidad, que es de corriente continua y oscila alrededor de los 24 V. Dicha tensión llega primero a un punto de control o "puerta de acceso", formada por un microcontrolador y un relé. Para alimentar el microcontrolador, se utiliza un convertidor DC-DC que regula la tensión de entrada de 24 V a 5 V, apta para poder alimentar al microcontrolador. Posteriormente, el relé gestiona la activación del suministro eléctrico hacia el resto del sistema. La tensión de 24 V, a través de otro regulador, se transforma en 12 V para alimentar el PC, los electroventiladores y el sensor de arco, conformando el sistema general de alimentación del dispositivo. Por último, el suministro de eléctrico de la linterna es proporcionada por otro convertidor de 24 V a 3,7 V que es la tensión de trabajo de la linterna. De una forma esquemática y resumida la Figura 18 muestra un resumen de la alimentación del dispositivo.

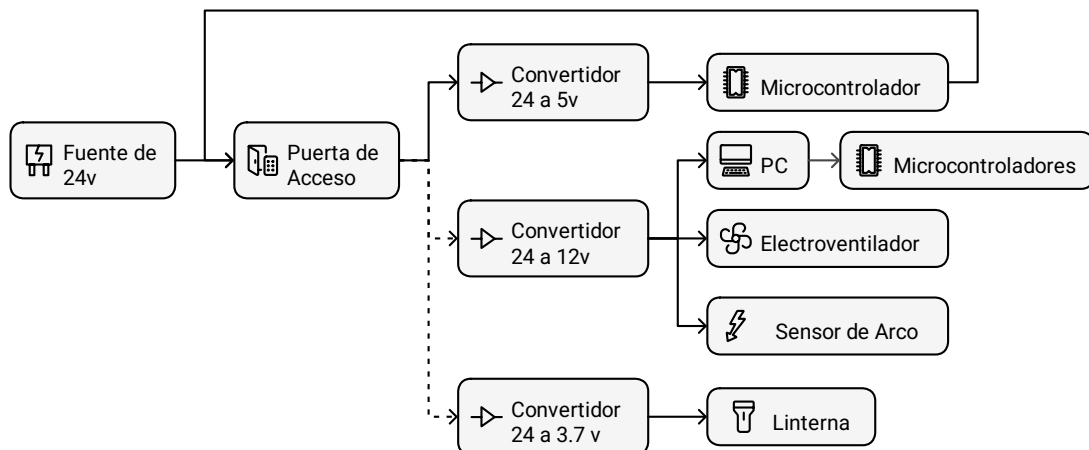


Figura 18. Esquema del sistema de alimentación del dispositivo



En cuanto a la comunicación de datos, se emplea una placa de conexiones diseñada específicamente para integrar todos los elementos. Los microcontroladores ESP32 están conectados al PC mediante un puerto USB, permitiendo simultáneamente su alimentación y la transmisión de datos. La comunicación entre dispositivos se realiza utilizando el protocolo UART de comunicación a través de este puerto serie, asegurando un intercambio de información eficiente y estable.

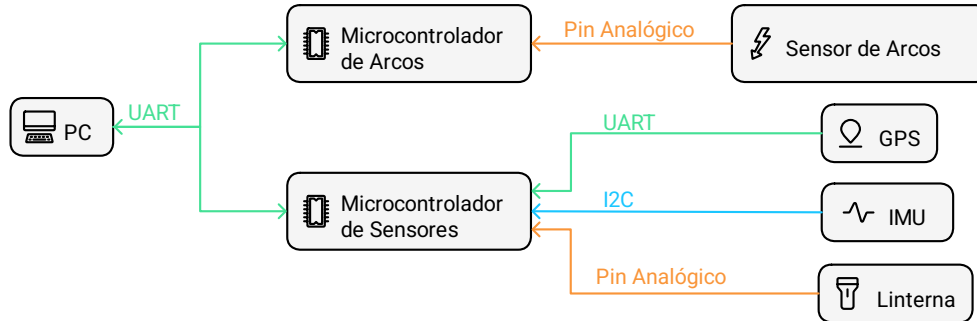
A continuación, se describe el proceso de recolección y transmisión de datos desde los sensores del dispositivo, junto con los protocolos de comunicación empleados para asegurar una transmisión adecuada. La Figura 19 muestra un esquema con los distintos protocolos de comunicación que se han empleado para cada componente.

El sensor de arco emite una señal analógica con valores entre 0 y 5 V, representando la intensidad del arco eléctrico. Esta señal se captura mediante uno de los pines analógicos del microcontrolador. Los datos recolectados son enviados al PC utilizando el protocolo UART a través del puerto serie, donde son gestionados y almacenados.

La información del IMU y el GPS se gestiona mediante un segundo microcontrolador. El GPS utiliza el protocolo UART para enviar sus datos. Este microcontrolador admite hasta tres canales UART simultáneos, asignando uno al GPS y otro para la comunicación con el PC. Adicionalmente, el GPS requiere alimentación eléctrica de 3.3V y conexión a tierra común con el microcontrolador, estableciendo así dos conexiones adicionales. Tanto la tierra como la alimentación proceden directamente del microcontrolador al que está conectado.

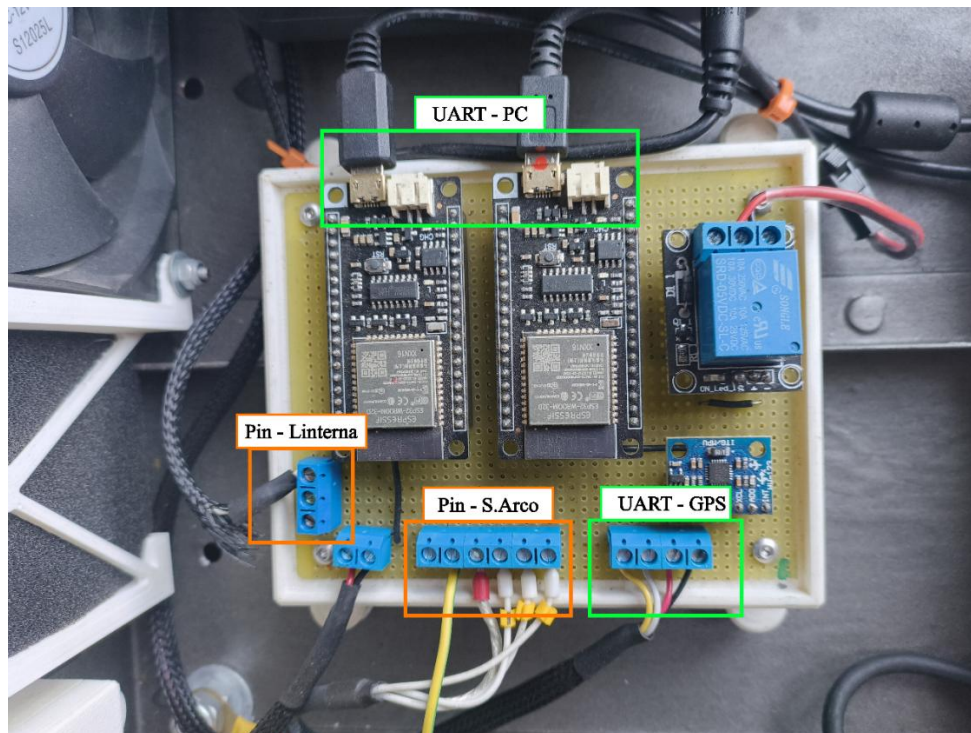
El IMU, por su parte, utiliza el protocolo de comunicación I2C, compatible con el microcontrolador ESP32, que está diseñado para operar con este estándar. De igual forma que con el GPS, es necesario alimentar el IMU y compartir una tierra común por lo que hay que añadir dos conexiones extras. Los datos de ambos sensores (GPS e IMU) son procesados y también transmitidos al PC mediante UART para su almacenamiento y análisis.

Por último, este microcontrolador controla el encendido y apagado tanto del PC como de la linterna. Para ello, gestiona un relé encargado del PC y envía una señal de pulso para activar o desactivar la linterna, cerrando así el ciclo de control y comunicación dentro del dispositivo.



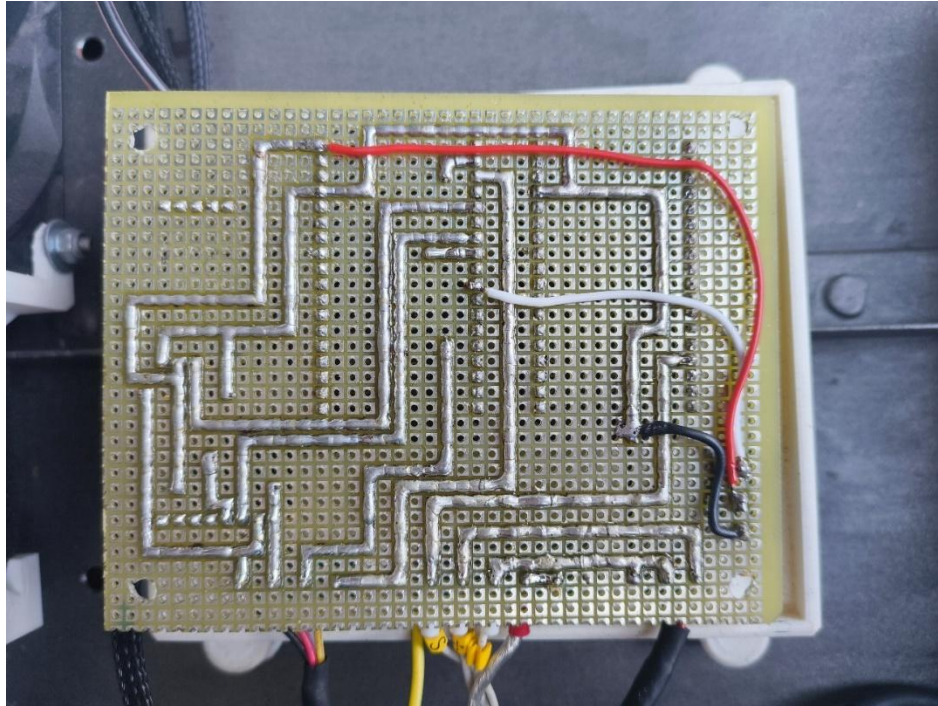
**Figura 19.** Esquema de comunicación entre los elementos del dispositivo

Para establecer la comunicación física entre los distintos elementos mediante los protocolos correspondientes, se ha diseñado y fabricado desde cero una placa electrónica, la cual se muestra en la Figura 20. La instalación de los componentes a controlar se llevó a cabo previamente, seguida de su correspondiente conexionado. La mayoría de las conexiones se realizaron mediante un carril de estaño, como se observa en la Figura 21, mientras que en aquellos casos donde el acceso resultaba complicado o se requirieron modificaciones posteriores, se emplearon cables para garantizar la correcta interconexión del sistema.



**Figura 20.** Placa electrónica del dispositivo

El resultado obtenido es un elemento de calidad considerable, a pesar de haber sido fabricado de forma manual, proporcionando además un alto grado de fiabilidad en la estabilidad de la conexión entre los componentes.



**Figura 21.** Detalle de la parte trasera de la placa electrónica del dispositivo

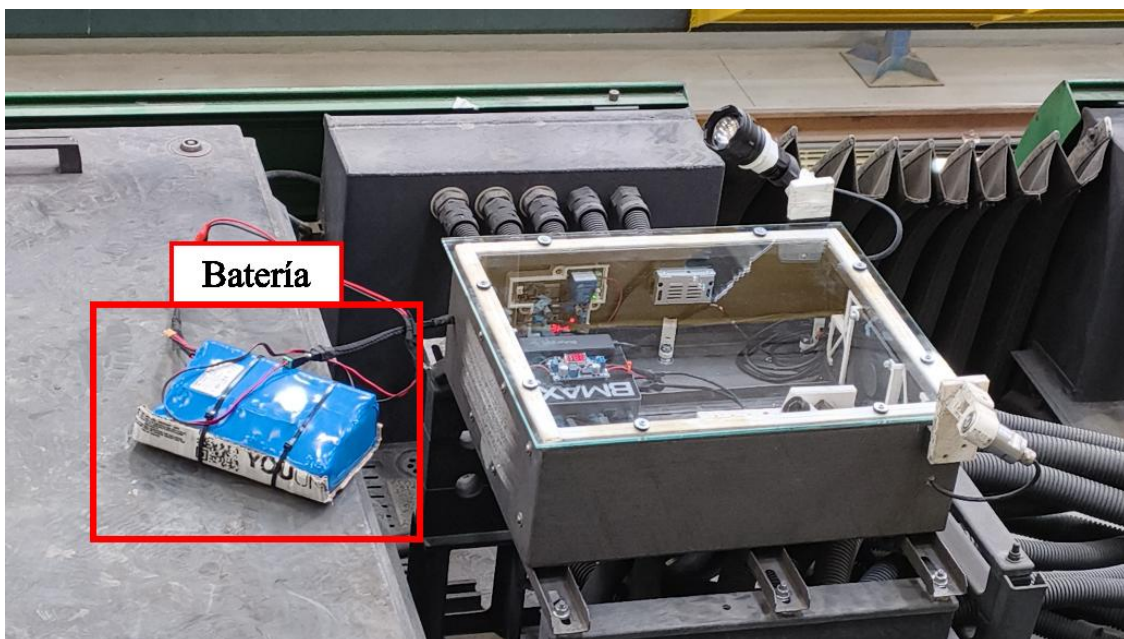
## 8. ENSAYOS

Con lo anteriormente explicado, se concluye la descripción de los componentes que integran el dispositivo, así como de su funcionalidad individual y conjunta. A partir de aquí, se aborda el propósito principal del sistema: la realización de ensayos.

### 8.1. Instalación del dispositivo

El primer paso consiste en instalar el dispositivo en una unidad del metro de Málaga. Para ello, se coordina una cita con el equipo de Metro de Málaga, que asigna una unidad en los talleres para facilitar la instalación. Utilizando los anclajes diseñados previamente según las especificaciones de los puntos de fijación permisibles, se asegura el montaje físico del dispositivo en la unidad.

Tras completar el montaje físico, se procede con el conexionado eléctrico. El dispositivo está equipado con un terminal externo para esta tarea, que requiere manejar tensiones de la unidad. Dado el riesgo asociado, se informa al operario de mantenimiento que debe realizar la conexión siguiendo los protocolos de seguridad establecidos por el fabricante de la unidad. Mientras se realiza esta conexión permanente, el día de la instalación se utiliza una batería externa como fuente de alimentación provisional como se observa en la Figura 22.



*Figura 22. Detalle del uso de batería para configuración el día de la instalación*

La batería permite verificar y calibrar los sensores, asegurando su correcta orientación hacia la zona de interacción del pantógrafo. Además, se realizan pruebas con la cámara para ajustar el enfoque y garantizar la captura de imágenes nítidas durante el ensayo, como se puede comprobar en la Figura 23. Estas verificaciones aseguran que el dispositivo está en perfecto estado y reduce la probabilidad de fallos durante la prueba final.



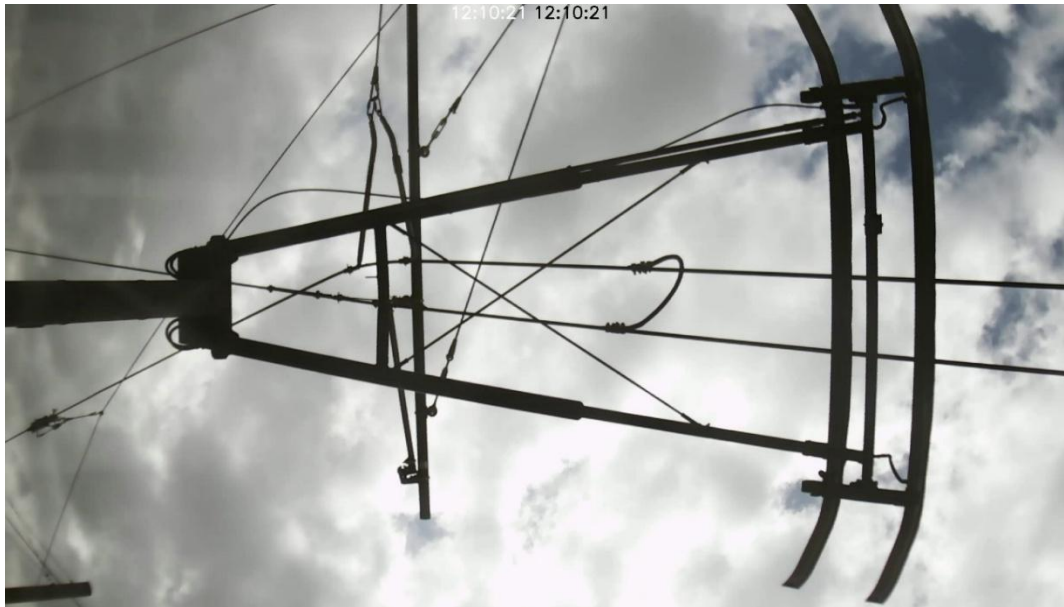
*Figura 23. Comprobación de la correcta visualización el día de la instalación*

## **8.2. Realización del ensayo**

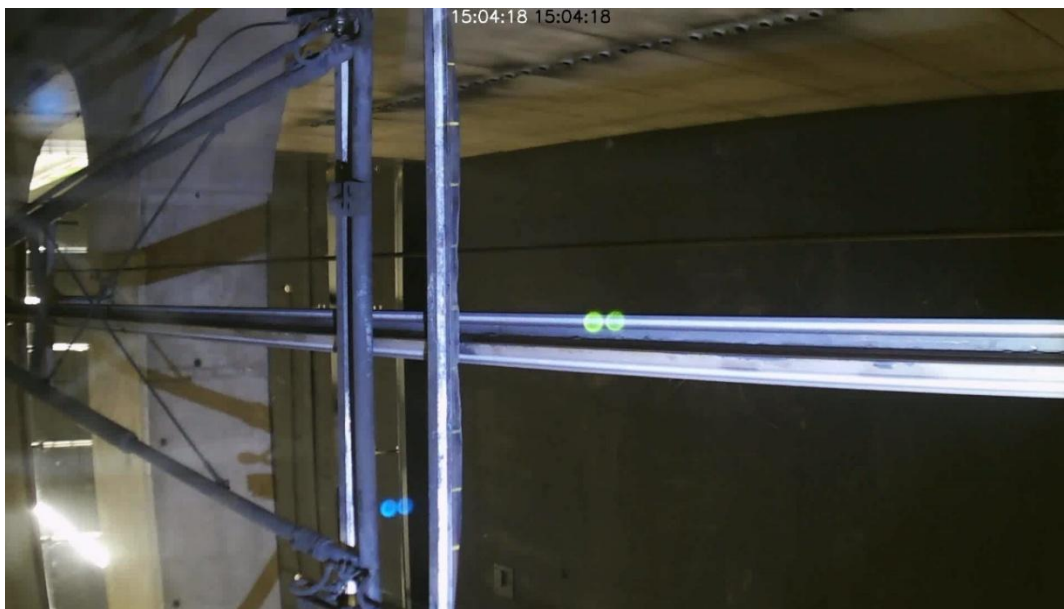
Una vez instalada la caja en una de las unidades, es necesario esperar a que esta unidad entre en funcionamiento en alguna de las líneas del metro. Cuando esto sucede, Metro de Málaga debe proporcionar información sobre su ubicación exacta para permitir una adecuada coordinación y abordar la unidad en una de las paradas correspondientes.

Al entrar en la unidad, es imprescindible seguir las instrucciones de manejo del dispositivo para evitar fallos y garantizar un funcionamiento correcto. Estas instrucciones detallan paso a paso las acciones a realizar y ofrecen soluciones para posibles imprevistos. El primer paso consiste en establecer la conexión con la puerta de acceso, lo que permite encender todos los componentes del sistema. Posteriormente, se procede a activar la linterna para asegurar una iluminación adecuada en el área de interés. Una vez realizados estos pasos, se establece una conexión remota con el ordenador integrado en el dispositivo. Desde allí, se ejecuta el programa de grabación y se lleva a cabo una prueba inicial para verificar que todo opera correctamente (Figura 24).

La metodología de grabación varía según las necesidades del ensayo. Normalmente, se realizan grabaciones completas en ambos sentidos de la línea, aunque en ocasiones se limitan a zonas específicas, ya sea para inspeccionar un tramo concreto o verificar el funcionamiento de elementos recién instalados.



(a)



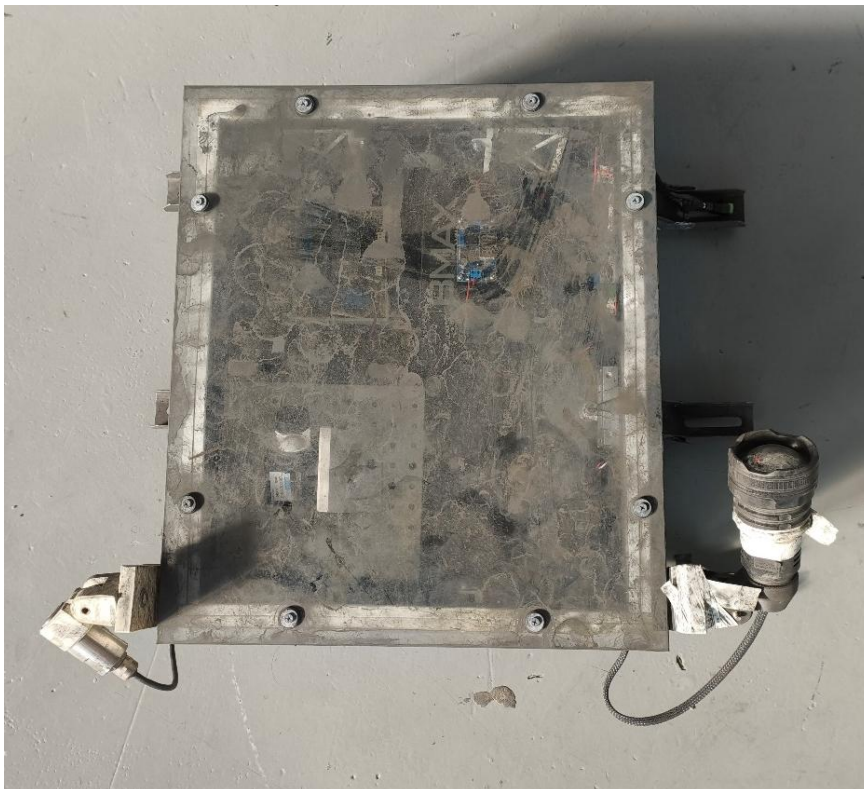
(b)

**Figura 24.** Comparativa de la vista entre las dos alturas de trabajo del pantógrafo. (a) Captura del vídeo en tramo de superficie. (b) Captura del vídeo en tramo de túnel

### 8.3. Mantenimiento del dispositivo

Dado que se trata de un equipo experimental, es fundamental realizar un mantenimiento periódico para garantizar su correcto funcionamiento y evitar que pequeños problemas se agraven con el tiempo. Uno de los aspectos clave es la limpieza regular de la superficie de vidrio de la parte superior, asegurando así la captura de vídeos con la mejor calidad posible. Como se puede observar en la Figura 25, si el dispositivo no se limpia con frecuencia se acaba acumulando la suciedad, impidiendo así una correcta ejecución de los ensayos. Además, es necesario revisar periódicamente los filtros del dispositivo y comprobar que se encuentran en buen estado, evitando la acumulación de suciedad en el interior de la caja.

Asimismo, es importante verificar el sellado entre el vidrio y la estructura de la caja, asegurando su hermeticidad ante condiciones climáticas adversas, como la lluvia. Este mantenimiento preventivo ayuda a prolongar la vida útil del dispositivo y a mantener la calidad de los datos recopilados durante los ensayos.



**Figura 25.** Dispositivo tras varios días instalado en una unidad

## 9. RESULTADOS

Los datos más relevantes obtenidos hasta ahora han sido los proporcionados por los vídeos capturados por la cámara del dispositivo. Estos vídeos ofrecen una perspectiva detallada de la infraestructura y del entorno, permitiendo un análisis exhaustivo de la evolución de las instalaciones respecto a la cantidad e intensidad de los arcos eléctricos detectados (Figura 26).



**Figura 26.** Muestra de arco eléctrico capturado por la cámara

Para el análisis, se revisan manualmente todas las grabaciones y se clasifican los arcos eléctricos observados entre cada parada. Se elabora un mapa de calor con una escala de colores: verde para las áreas sin arcos, amarillo para arcos de baja intensidad, y escalas crecientes de naranja, rojo y negro para aquellos de mayor magnitud. Este sistema permite identificar rápidamente las zonas más afectadas y prioritarias para mantenimiento. La Figura 27 muestra un trozo del mapa que ha generado gracias a la información obtenida por los videos.

Del análisis de los resultados de arcos se constata cómo se ha podido monitorizar la aparición de nuevos arcos en zonas concretas entre estaciones. Estos arcos pueden estar causados por diferentes motivos y su estudio es de interés para los administradores de infraestructuras. Localizar nuevas zonas de arcos es de gran interés para poder monitorizar la evolución de los desgastes de hilo así como permitir una mejor planificación de las actividades de mantenimiento futuras.

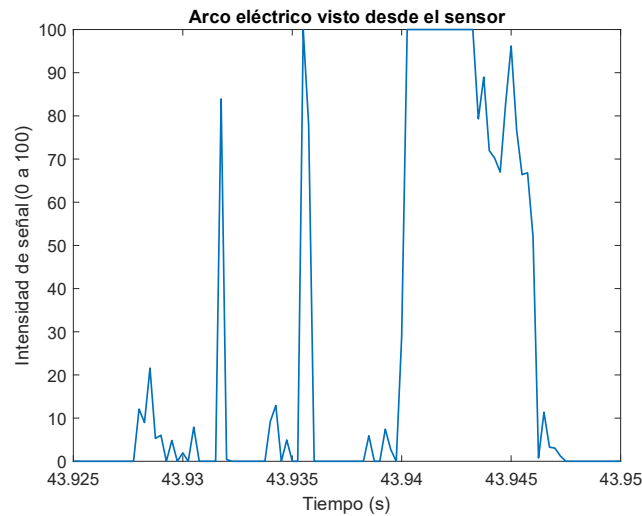
	Tramo 1	Tramo 2	Tramo 3
<b>Ensayo 1</b>		0:37	0:47
		0:45	
		1:01	
		1:12	
		1:27	
		1:36	
<b>Ensayo 2</b>		3:10	5:13
		3:53	5:15
		4:05	5:18
<b>Ensayo 3</b>		6:23	7:50
		6:28	
		6:30-6:35	
<b>Ensayo 4</b>		17:13	19:19
		17:48	19:23
		18:04	19:30-19:34
		18:09-18:17	19:53
			19:57

**Figura 27.** Muestra del mapa de calor generado para localización de zonas conflictivas

En cuanto a los datos proporcionados por el sensor de arco, se ha identificado una discrepancia significativa respecto a la información recogida por la cámara. A pesar de que el sensor debería proporcionar mayor información y detectar arcos eléctricos adicionales, no se ha logrado establecer una correlación consistente entre ambas fuentes de datos de forma exacta.

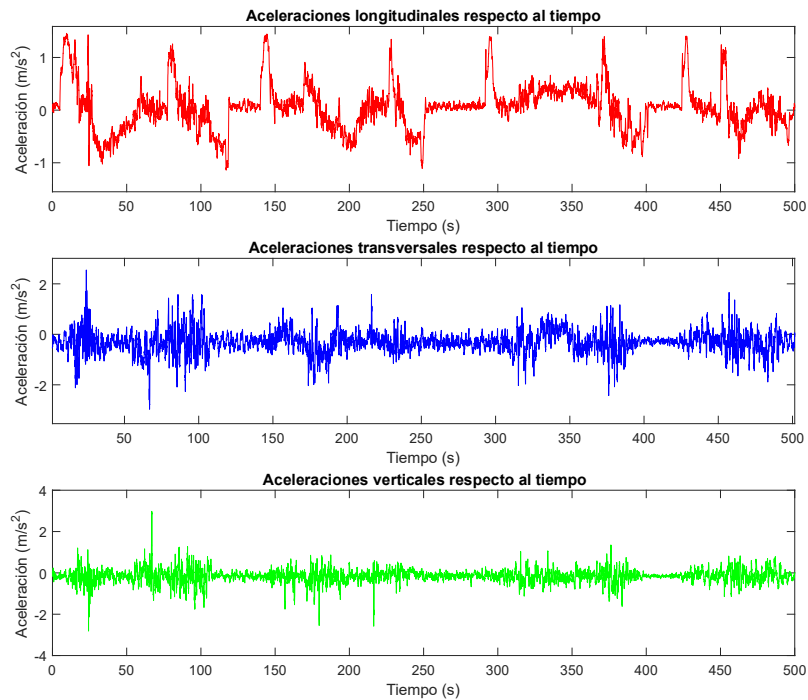
Los vídeos ofrecen evidencia visual directa de los arcos eléctricos observados, lo que les otorga mayor validez. Por el contrario, el sensor de arco parece limitado a detectar únicamente aquellos arcos de gran intensidad o que emiten la radiación de una forma directa al sensor. Este comportamiento puede explicarse por qué no se reconocen arcos de menor intensidad o aquellos situados en los extremos del pantógrafo. Aunque el sensor ha registrado múltiples arcos, no ha logrado capturar todos los que han sido identificados visualmente, limitando su utilidad en ciertos escenarios.

En la Figura 28 se puede ver una de estas utilidades, que es examinar con detalle aquellos arcos que el sensor si ha sido capaz de detectar. Las unidades que proporciona el sensor son adimensionales y representan únicamente la intensidad de energía que emite. Gracias a la elevada frecuencia de muestreo, es posible identificar eventos en los que el arco eléctrico no se mantiene de forma continua en el tiempo, sino que, en intervalos extremadamente cortos, alcanza valores máximos y mínimos. Este fenómeno resulta imperceptible mediante la grabación.



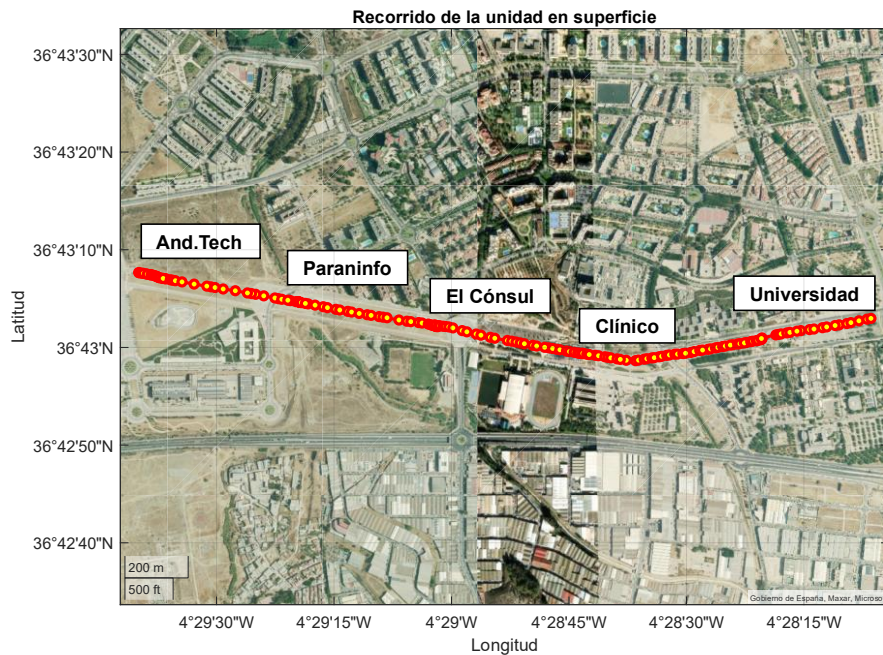
**Figura 28.** Arco eléctrico visto desde el sensor

En cuanto a los datos obtenidos del GPS y el IMU, se ha constatado que las lecturas son correctas y proporcionan información valiosa que puede relacionarse directamente con el mapa de calor de la infraestructura. Esto permite investigar posibles correlaciones entre la aparición de arcos eléctricos y el movimiento de la unidad. Como se puede apreciar en la Figura 29 en la aceleración longitudinal se puede observar con detalle aquellos momentos en los que la unidad se encuentra esperando o en movimiento.



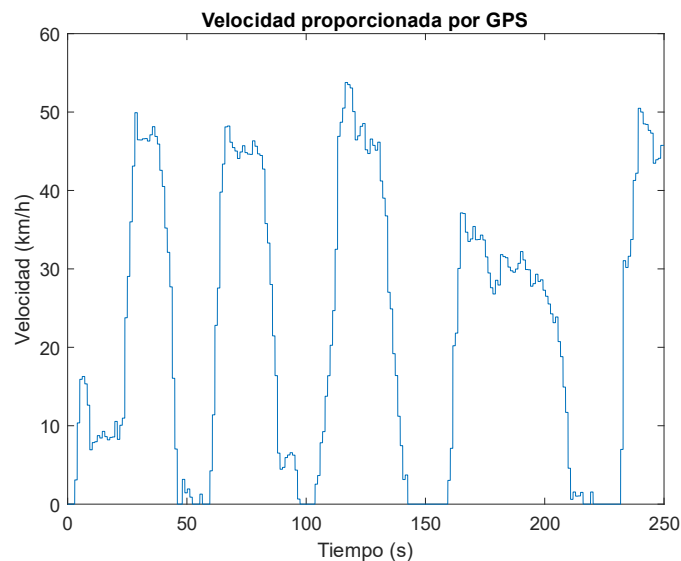
**Figura 29.** Representación de las aceleraciones registradas por IMU en los 3 ejes principales

La Figura 30 muestra en un mapa satelital el recorrido de la señal recibida por el GPS en uno de los ensayos, coincidiendo de forma exacta con el recorrido que realizan las unidades en las infraestructuras de Metro de Málaga, lo que permite asegurar la calidad de los datos obtenidos.



**Figura 30.** Representación de las señales de GPS obtenidas por el dispositivo en un mapa real

Por último, la señal de GPS también proporciona información sobre velocidad mostrada en la Figura 31, donde se observa claramente los tramos donde se encuentran las estaciones.



**Figura 31.** Representación de la velocidad proporcionada por el GPS del dispositivo



---

## 10. ACTUALIZACIONES PASADAS

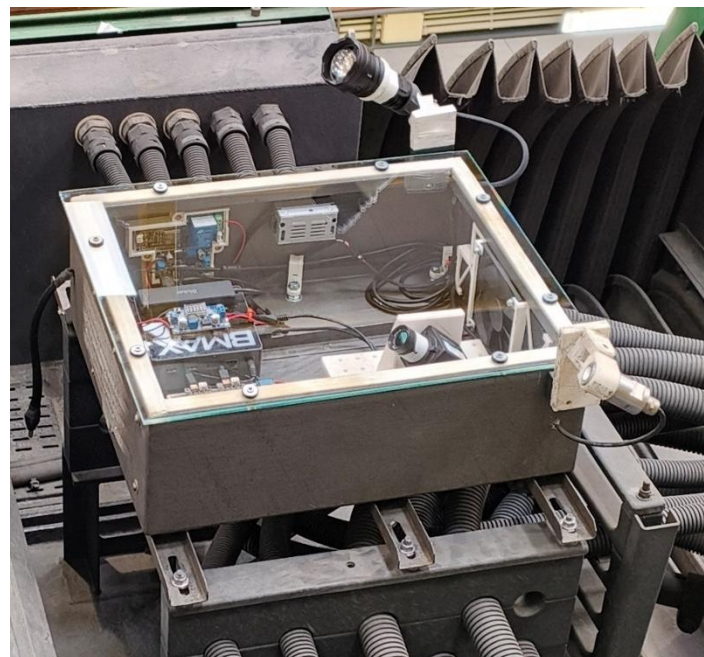
Es importante señalar que el dispositivo presentado en este proyecto no es un producto finalizado, sino una plataforma de investigación en constante desarrollo y evolución. La versión actual es el resultado de dos años de trabajo, pruebas y mejoras, donde se han descartado elementos ineficaces y optimizado aquellos que han demostrado ser funcionales.

Por ejemplo, en etapas iniciales, el dispositivo contaba con una cámara lateral destinada a localizar la unidad en túneles. Sin embargo, fue eliminada al comprobar que no aportaba información útil y afectaba negativamente al rendimiento del programa de grabación. Tampoco se incluía una linterna, lo que resultaba en imágenes oscuras que dificultaban la evaluación de detalles de la instalación. Otra propuesta descartada fue un sensor LiDAR, implementado para mapear el descentramiento de la catenaria, pero incapaz de operar a la velocidad del tren, lo que generaba datos incorrectos. Y la actualización más visible que es la sustitución de la tapa de policarbonato por vidrio. En la Figura 32 se muestra una comparativa de ambas versiones donde se puede observar la evolución del dispositivo.

Adicionalmente, numerosos retos en programación han dado lugar a un dispositivo progresivamente más robusto y eficiente. Sin embargo, sigue siendo una herramienta experimental sujeta a actualizaciones y ajustes tras cada ensayo.



(a)



(b)

**Figura 32.** Comparativa entre distintas versiones del dispositivo. (a) Versión con policarbonato, LiDar y sin linterna. (b) Última versión instalada

## 11. CONCLUSIONES Y TRABAJOS FUTUROS

En el presente proyecto se ha desarrollado un dispositivo sensorizado diseñado para la monitorización del estado de la infraestructura ferroviaria, con especial atención a la interacción entre el pantógrafo y la catenaria. Tras diversas pruebas y ajustes, el sistema ha demostrado ser una herramienta adecuada para la recopilación y análisis de datos, proporcionando información valiosa para la detección de posibles problemas en la instalación.

El dispositivo ha sido instalado en unidades ferroviarias del Metro de Málaga y sometido a múltiples ensayos en condiciones reales. Durante estas pruebas, se ha logrado identificar zonas de mayor desgaste en la catenaria a través de un sistema de grabación visual y la clasificación de arcos eléctricos mediante un mapa de calor. Si bien se han detectado limitaciones respecto al sensor de arco, las cámaras han proporcionado resultados suficientemente detallados para evaluar el estado de la infraestructura.

De cara al futuro, uno de los objetivos prioritarios es el control remoto total del dispositivo, permitiendo realizar ensayos sin necesidad de que un operador esté presente en la unidad. Esto eliminaría la dependencia de la coordinación con Metro de Málaga para conocer la ubicación del sistema, permitiendo supervisar remotamente la línea en uso y realizar pruebas sin intervención física, optimizando así los recursos y el tiempo.

A modo de resumen general se presenta la **Tabla 1** donde se muestran los componentes principales empleados y el objetivo que desempeñan dentro del dispositivo.

Componente	Objetivo
Mini Pc	Gestión principal de procesos
Microcontroladores	Gestión y transmisión de datos
Sensor de arco	Detección de arcos eléctricos
Cámara HD	Grabación de vídeos
Linterna	Mejora del entorno de grabación
IMU	Medición de aceleraciones
GPS	Localización de la unidad en superficie

**Tabla 1.** Listado de componentes principales del dispositivo



## 12. REFERENCIAS

- [1] M. Simarro, S. Postigo, J.A. Cabrera, J.J. Castillo, A procedure for validating rigid catenary models using evolutionary techniques. *Comput Struct.* 202;228,2019
- [2] Bucca, G.; Collina, A.; Tanzi, E. Experimental analysis of the influence of the electrical arc on the wear rate of contact strip and contact wire in a.c. system. In *Advances in Italian Mechanism Science*; Springer: Berlin/Heidelberg, Germany, 2016
- [3] T. Kziazzyk, E. Gavignet, P.-H. Cornuault, P. Baucour y D. Chamagne, “Review on Test Benches Studying Sliding Electrical Contact and Synthesis of Experimental Results”, *Energies*, vol. 16, n.º 3, p. 1294, 2023.
- [4] “Geometry & Contact Wire Wear”. Mermec Group. Accedido el 24 de febrero de 2025. [En línea]. Disponible: <https://www.mermecgroup.com/measurement-br-trains-e-systems/catenary-measurement/78/geometry-e-contact-wire-wear.php>
- [5] “measuring Catenary | on Vehicle | CAT-VW”. SelectraVision, Computer Vision System for Measuring, Inspection and Diagnostics. Accedido el 24 de febrero de 2025. [En línea]. Disponible: <https://www.selectravision.com/products-cat-vw.php>
- [6] Aplicaciones ferroviarias. Sistemas de captación de corriente. Requisitos y validaciones de medidas de la interacción dinámica entre el pantógrafo y las líneas aéreas de contacto., UNE-EN 50317:2012, 2012.
- [7] “UV-Arc | Sondas de medición UV | sglux”. sglux. Accedido el 24 de febrero de 2025. [En línea]. Disponible: <https://sglux.de/es/producto/uv-arc/>
- [8] “Pixabay”. Pixabay. Accedido el 13 de febrero de 2024. [En línea]. Disponible: <https://pixabay.com/es/photos/pant%C3%B3grafo-locomotora-electrica-2647175/>



---

# ANEXO A CÓDIGO GENERAL



```
# Cargar librerías externas
#-----
import PySimpleGUI as sg
import os
import serial
import threading
import datetime
import time
import numpy as np
import queue
import io
import cv2
#-----

# Definición de una clase para el control de los FPS
#-----
class BufferedVideoWriter:
    def __init__(self, filename, fourcc, fps, frameSize, maxBufferSize=25):
        self.fourcc=fourcc
        self.thread=None
        self.queue=queue.Queue()
        self.fps=fps
        self.maxBufferSize=maxBufferSize
        self.frameSize=frameSize
        self.filename=filename
    def start(self):
        if self.thread:
            return
        dt=1./self.fps
        def loop():
            writer=cv2.VideoWriter(filename=self.filename, fourcc=self.fourcc,
            fps=self.fps, frameSize=self.frameSize)
            t=time.time()
            self.queue.put((cv2.rectangle(self.frameSize, (0, 0),
            self.frameSize, (0, 0, 0), -1),t))
            while self.thread or self.queue.qsize():
                frame,ts=self.queue.get()
                if frame is None:
                    break
                while t<ts:
                    writer.write(frame)
                    t+=dt
                writer.release()
            self.thread=threading.Thread(target=loop)
            self.thread.start()
        def stop(self):
            if not self.thread:
                return
            self.write(None)
            t=self.thread
            self.thread=None
            t.join()

        def write(self, image):
            if self.thread and self.queue.qsize()<self.maxBufferSize:
                return self.queue.put((image,time.time()))
#-----
```



```
# Iniciciacion de variables
#-----

# Configuracion del puerto serie IMU y GPS
puerto_general = '/dev/ttyUSB1'
freq_puerto_general = 20

# Configuracion video de salida
fourcc = cv2.VideoWriter_fourcc(*'mp4v')
fps_1 = 20
frameSize_1 = (1920, 1080)

# Variables de control
recording = False
out_1 = None
estado_hilo = True

# Carpeta del lugar de guardado de la grabacion
carpeta_datos_generales =
r"/home/caja/NUCLEO_CAJA/DATOS/FICHEROS_DATOS_GENERAL"
carpeta_video_1= r"/home/caja/NUCLEO_CAJA/DATOS/VIDEO_PANTOGRAFO"

# Configuracion del puerto serie Arcos
puerto_arco = '/dev/ttyUSB0'
estado_hilo_arcos = True
carpeta_datos_arcos = r"/home/caja/NUCLEO_CAJA/DATOS/FICHEROS_DATOS_ARCOS"
#-----

#Funciones
#-----

# Función para leer y procesar los datos del puerto serial
#-----
def leer_puerto_serial_general():
    global estado_hilo
    global recording_general
    global datos

    recording_general = False
    datos = 0
    contador = 0

    while estado_hilo==True:
        try:
            #Leer datos del puerto serial
            datos = puerto_serial.readline().decode('utf-8').strip()

            contador += 1

            if recording_general == True and datos.startswith('\t'):
                fecha_actual = datetime.now().strftime("%H:%M:%S")[:-3]
                archivo_texto_general.write(datos + "\n")

        except UnicodeDecodeError:
            print("Ha ocurrido un fallo al realizar la lectura por puerto
            serie, volver a intentarlo.")
        except:
```



```
print("Ha ocurrido un fallo que se desconoce")
#-----

# Funcion que ejecuta el hilo y trata los datos del puerto serial
#-----
def hilo_puerto_serial():
    global archivo_texto_general
    global hilo_serial

    #Ejecuta el hilo del puerto serial general
    hilo_serial = threading.Thread(target=leer_puerto_serial_general)
    hilo_serial.start()
#-----

# Funcion para configuracion general de las camaras
#-----
def configuracion_camaras():
    global cap_1

    # Configuración de las cámaras
    cap_1 = cv2.VideoCapture(0)

    if not cap_1.isOpened():
        print("Error: No se pudo abrir la cámara.")
        return

    cap_1.set(cv2.CAP_PROP_BUFFERSIZE, 10)

    # Pantografo
    cap_1.set(cv2.CAP_PROP_FRAME_WIDTH, 1920)
    cap_1.set(cv2.CAP_PROP_FRAME_HEIGHT, 1080)
#-----

# Funcion para implementar texto blanco en el video
#-----
def mostrar_datos_en_video_blanco(frame, anchura, altura, altura_linea,
dato_recogido, texto="", color = (255, 255, 255)):
    font = cv2.FONT_ITALIC
    cv2.putText(frame, texto + str(dato_recogido), (anchura, altura), font,
altura_linea, (255, 255, 255), 2)
#-----

# Funcion para implementar texto negro en el video
#-----
def mostrar_datos_en_video_negro(frame, anchura, altura, altura_linea,
dato_recogido, texto="", color = (255, 255, 255)):
    font = cv2.FONT_ITALIC
    cv2.putText(frame, texto + str(dato_recogido), (anchura, altura), font,
altura_linea, (0, 0, 0), 2)
#-----

# Funcion para mostrar todos los datos en video
#-----
def mostrar_datos_completos(frame_1):
    escala_1 = float(1)
    altura_x = 800
    altura_y = 25
```



```
hora_actual = datetime.now().strftime("%H:%M:%S")

mostrar_datos_en_video_blanco(frame_1, altura_x, altura_y, escala_1,
hora_actual, '', 'Blanco')

altura_y_negra = altura_y
altura_x_negra = 950

mostrar_datos_en_video_negro(frame_1, altura_x_negra, altura_y_negra,
escala_1, hora_actual, '', 'Negro')
#-----

# Funcion para generar el nombre de los archivos
#-----
def generar_nombre_archivos(Tipo_Archivo):
    fecha_hora_actual = datetime.now().strftime("%Y-%m-%d_%H-%M-%S")

    if Tipo_Archivo == "Datos_Generales":
        nombre_archivo = f"datos_generales_{fecha_hora_actual}.txt"
        ruta_archivo = os.path.join(carpeta_datos_generales, nombre_archivo)
        return ruta_archivo
    elif Tipo_Archivo == "Datos_Arcos":
        nombre_video_0 = f"datos_arcos_{fecha_hora_actual}.txt"
        ruta_video_0 = os.path.join(carpeta_datos_arcos, nombre_video_0)
        return ruta_video_0
    elif Tipo_Archivo == "Video_1":
        nombre_video_1 = f"video_pantografo_{fecha_hora_actual}.avi"
        ruta_video_1 = os.path.join(carpeta_video_1, nombre_video_1)
        return ruta_video_1
#-----

# Funcion para generar las rutas de los videos
#-----
def rutas_videos():
    ruta_video_1 = generar_nombre_archivos("Video_1")
    ruta_archivos_generales = generar_nombre_archivos("Datos_Generales")
    ruta_archivos_arcos = generar_nombre_archivos("Datos_Arcos")
    return ruta_video_1, ruta_archivos_generales, ruta_archivos_arcos
#-----

# Funcion para crear la interfaz grafica
#-----
def create_gui():
    layout = [
        [sg.Button('Iniciar grabación', key='start'), sg.Button('Detener
grabación', key='stop'),
        sg.Text('Grabando', background_color='green', visible=False,
key='recording_indicator')],
        ]
    window_control = sg.Window('Control Video', layout, resizable=True,
location=(100, 100))
    return window_control
#-----
```



```
# Función para leer y procesar los datos del puerto serial
#-----
def leer_puerto_serial_arcos():
    global dato_arcos_1, dato_arcos_2
    global datos_arcos
    global estado_hilo_arcos
    global archivo_texto_arcos
    global recording_arcos
    global valid_value, last_valid_time
    global datos_buffer
    global buffer

    buffer = io.BytesIO()

    recording_arcos = False
    valid_value = False

    dato_arcos_1 = 0
    dato_arcos_2 = 0
    datos_arcos = 0

    last_valid_time = time.time()

    puerto_serial_arcos.reset_input_buffer()

    while estado_hilo_arcos==True:

        # Leer datos del puerto serial
        datos_arcos = puerto_serial_arcos.readline()

        if recording_general == True:
            archivo_texto_arcos.write(datos_arcos)
            buffer.write(datos_arcos)
#-----

# Funcion que ejecuta el hilo y trata los datos del puerto serial
#-----
def hilo_puerto_serial_arcos():
    global archivo_texto_arcos
    global hilo_serial_arcos

    hilo_serial_arcos = threading.Thread(target=leer_puerto_serial_arcos)
    hilo_serial_arcos.start()
#-----
```



```
#Codigo principal
#-----
if __name__ == "__main__":
    window_control = create_gui()

    # Abrir el puerto serial con una velocidad de transmisión de baudios
    determinada
    puerto_serial = serial.Serial(puerto_general, 115200)
    puerto_serial_arcos = serial.Serial(puerto_arco, 500000)

    # Configurar camara
    configuracion_camaras()

    #Ejecucion de hilos paralelos para cada puerto
    hilo_puerto_serial()
    hilo_puerto_serial_arcos()

    # Crea la ventana y permitir redimensionarlar
    cv2.namedWindow('Pantografo', cv2.WINDOW_NORMAL)

    #Bucle principal
    while True:
        event, values = window_control.read(timeout=5)

        if event == sg.WINDOW_CLOSED:
            break

        # Captura de imágenes de las cámaras
        ret_1, frame_1 = cap_1.read()

        if ret_1:
            mostrar_datos_completos(frame_1)

            # Actualización de las imágenes en la interfaz gráfica
            cv2.imshow('Pantografo', frame_1)

            # Control de la grabación
            if recording == True:
                if out_1 is None:
                    ruta_video_1, ruta_archivos_generales,
                    ruta_archivos_arcos = rutas_videos()
                    out_1 = BufferedVideoWriter(ruta_video_1, fourcc, fps_1,
                    frameSize_1)

                    archivo_texto_general = open(ruta_archivos_generales, "w")
                    archivo_texto_arcos = open(ruta_archivos_arcos, "wb")

                    recording_general = True

                    window_control['recording_indicador'].update(visible=True)
                    window_control['recording_indicador'].update(background_color='red')
                    out_1.start()
                    out_1.write(frame_1)
                else:
                    if out_1 is not None:
                        out_1.stop()
```

```
out_1 = None
recording_general = False
archivo_texto_general.close()
archivo_texto_arcos.close()
window_control['recording_indicator'].update(background_color='green')
window_control['recording_indicator'].update(visible=False)
)

# Control de los botones de la interfaz gráfica
if event == 'start':
    recording = True
elif event == 'stop':
    recording = False
cv2.waitKey(1)

# Liberación de las cámaras y cerrar la ventana
cap_1.release()
estado_hilo = False
estado_hilo_arcos = False

# Esperar a que el hilo acabe para continuar con el código principal
hilo_serial.join()
hilo_serial_arcos.join()

# Cerrar el puerto serial, la captura de video y el archivo de texto
puerto_serial.close()
puerto_serial_arcos.close()

# Cierre de la interfaz gráfica
window_control.close()
cv2.destroyAllWindows()
#-----
```



# **ANEXO B CÓDIGO CAPTURA DE ARCOS**



```
// Definicion de pines y variables necesarias
const int pin_voltaje = 35;
int lectura_voltaje; // Almacena la lectura analogica
uint16_t contador_isr = 0; // Contador de interrupciones

// Configuracion del temporizador
hw_timer_t *timer = NULL;
int16_t freq = 4000; // Frecuencia de muestreo en Hz
uint8_t isrTask = 0; // Semaforo para la tarea principal

// Interrupcion del temporizador
void IRAM_ATTR onTimer() {
    isrTask = 1;
    contador_isr++;
    if (contador_isr == freq) contador_isr = 0;
}

// Inicializacion de la comunicacion
void setup() {
    Serial.begin(500000);

    // Inicializacion del temporizador
    timer = timerBegin(0, 80, true);
    timerAttachInterrupt(timer, &onTimer, true);
    timerAlarmWrite(timer, 1000000 / freq, true);
    timerAlarmEnable(timer);
}

// Bucle principal
void loop() {
    if (isrTask == 1) {
        lectura_voltaje = analogRead(pin_voltaje); // Lectura del ADC
        Serial.printf("%.4d\t%.4d\n", lectura_voltaje, contador_isr);
        isrTask = 0; // Restablecer semaforo
    }
}
```



# **ANEXO C CÓDIGO PUERTA DE ACCESO**



```
#include <WiFi.h> // Librería para la conexión Wi-Fi
#include <WebServer.h> // Librería para el servidor web

// Define el pin del LED que se controlará de forma remota
#define LED_1 12

// Crea un servidor web que escucha en el puerto 80 (HTTP)
WebServer server(80);

// Configuración de red fija (IP estática)
IPAddress local_IP(192, 168, 137, 50); // Dirección IP del ESP32
IPAddress gateway(192, 168, 137, 1); // Puerta de enlace (router)
IPAddress subnet(255, 255, 255, 0); // Máscara de subred

// Función para manejar la página principal del servidor web
void handleRoot() {
  // Genera una página HTML con botones para encender y apagar el LED
  String html = "<html><body>";
  html += "<h1>Interruptor General Caja</h1>";
  html += "<p>Encender Caja: <a
href=\"/led1/on\"><button>Encender</button></a> ";
  html += "<a href=\"/led1/off\"><button>Apagar</button></a></p>";
  html += "</body></html>";

  // Envía la página HTML al cliente web
  server.send(200, "text/html", html);
}

// Función para encender el LED cuando se accede a la URL correspondiente
void handleLedOn() {
  // Obtiene la parte relevante de la URL para determinar qué LED encender
  String led = server.uri().substring(1, 5);

  // Si la solicitud es para el LED 1, lo enciende
  if (led == "led1") {
    digitalWrite(LED_1, HIGH);
  }

  // Redirige al usuario de vuelta a la página principal
  server.sendHeader("Location", "/");
  server.send(302);
}

// Función para apagar el LED cuando se accede a la URL correspondiente
void handleLedOff() {
  // Obtiene la parte relevante de la URL para determinar qué LED apagar
  String led = server.uri().substring(1, 5);

  // Si la solicitud es para el LED 1, lo apaga
  if (led == "led1") {
    digitalWrite(LED_1, LOW);
  }

  // Redirige al usuario de vuelta a la página principal
  server.sendHeader("Location", "/");
  server.send(302);
}
```



```
//Inicializacion
void setup() {
  Serial.begin(115200); // Inicializa la comunicación serial para depuración

  // Configura el pin del LED como salida y lo apaga inicialmente
  pinMode(LED_1, OUTPUT);
  digitalWrite(LED_1, LOW);

  // Configura la IP estática antes de conectarse a la red Wi-Fi
  if (!WiFi.config(local_IP, gateway, subnet)) {
    Serial.println("Configuración de IP estática fallida");
  }

  // Conéctate a la red Wi-Fi con SSID y contraseña especificados
  WiFi.begin("red", "contrasena");
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Conectando a la red Wi-Fi...");
  }

  // Mensajes de depuración en la consola serie
  Serial.println("Conexión Wi-Fi establecida");
  Serial.print("Dirección IP: ");
  Serial.println(WiFi.localIP());

  // Asigna las funciones de control de LED a las rutas del servidor web
  server.on("/", handleRoot);
  server.on("/led1/on", handleLedOn);
  server.on("/led1/off", handleLedOff);

  // Inicia el servidor web
  server.begin();
  Serial.println("Servidor web iniciado");

  delay(1000); // Pequeño retraso antes de entrar en el bucle principal
}

//Bucle principal
void loop() {
  // Atender las solicitudes entrantes del servidor web
  server.handleClient();
}
```



Sistema embarcado para el monitoreo y diagnóstico del estado del hilo de contacto en las infraestructuras del Metro de Málaga





# **ANEXO D CÓDIGO CAPTURA DE IMU Y GPS**



```
// Configuración del temporizador
//-----
hw_timer_t *timer = NULL;
int16_t freq = 20;           // Frecuencia de muestreo en Hz
uint8_t isrTask = 0;        // Semáforo para la tarea

void IRAM_ATTR onTimer() { // Interrupción del temporizador
    isrTask = 1;
}
//-----

// Interruptor WIFI
//-----
#include <WiFi.h>
#include <WebServer.h>

// Define los pines del LED
#define LED_1 12
#define LED_2 23

// Crea un servidor web en el puerto 80
WebServer server(80);

// Configuración de red fija
IPAddress local_IP(192, 168, 137, 100); // Cambia esta IP a la que desees
asignar
IPAddress gateway(192, 168, 137, 1);    // Cambia a la IP de tu gateway
IPAddress subnet(255, 255, 255, 0);    // Máscara de subred

// Genera la página HTML con dos botones para controlar los LED
void handleRoot() {
    String html = "<html><body>";
    html += "<h1>Controles dentro de la caja</h1>";
    html += "<p>Encender ordenador: <a
href=\"/led1/on\"><button>Encender</button></a> <a
href=\"/led1/off\"><button>Apagar</button></a></p>";
    html += "<p>Encender linterna: <a
href=\"/led2/on\"><button>Encender</button></a> <a
href=\"/led2/off\"><button>Apagar</button></a></p>";
    html += "</body></html>";
    // Envía la página HTML al cliente
    server.send(200, "text/html", html);
}

// Manejo del interruptor para encendido
void handleLedOn() {
    // Obtiene el número de LED a encender de la URL
    String led = server.uri().substring(1, 5);

    // Enciende el LED correspondiente
    if (led == "led1") {
        digitalWrite(LED_1, LOW);
        digitalWrite(LED_1, HIGH); // Encender Ordenador
        delay(200);
        digitalWrite(LED_1, LOW);
    } else if (led == "led2") {
```



```
digitalWrite(LED_2, HIGH);
digitalWrite(LED_2, LOW); // Encender linterna
delay(200);
digitalWrite(LED_2, HIGH);
}

// Redirige al usuario a la página principal
server.sendHeader("Location", "/");
server.send(302);
}

// Manejo del interruptor para apagado
void handleLedOff() {
// Obtiene el número de LED a apagar de la URL
String led = server.uri().substring(1, 5);

// Apaga el LED correspondiente
if (led == "led1") {
digitalWrite(LED_1, LOW);
digitalWrite(LED_1, HIGH); // Apagar Ordenador
delay(5000);
digitalWrite(LED_1, LOW);

} else if (led == "led2") {
digitalWrite(LED_2, HIGH);
digitalWrite(LED_2, LOW); // Apagar linterna
delay(2000);
digitalWrite(LED_2, HIGH);
}

// Redirige al usuario a la página principal
server.sendHeader("Location", "/");
server.send(302);
}
//-----

//GPS
//-----
#include <HardwareSerial.h>

HardwareSerial gpsSerial(2);
const uint8_t rx2Pin =16;
const uint8_t tx2Pin =17;

boolean mostrado_gps = false;
int counter;

int horas;
int minutos;
int segundos;
int gradosLat;
int minutosLat;
float segundosLat;
int gradosLon;
int minutosLon;
float segundosLon;
```



```
int pos; int stringplace = 0; int pos_1; int stringplace_1 = 0;
String nmea[15]; String nmea_1[15]; String GNGGA_msg; String GNVTG_msg;
String tiempo.UTC; String latitud; String longitud;

unsigned char buffer[64];
int count=0;
unsigned long ultima_ejecucion = 0;
//-----

//IMU
//-----
// Librerias I2C para controlar el mpu6050
// la libreria MPU6050.h necesita I2Cdev.h, I2Cdev.h necesita Wire.h
#include "I2Cdev.h"
#include "MPU6050.h"
#include "Wire.h"

MPU6050 sensor;

// Valores sin procesar del acelerometro y giroscopio en los ejes x,y,z
int16_t ax, ay, az;
int16_t gx, gy, gz;
int16_t temperature;

long tiempo_prev;
float dt;
float ang_x, ang_y;
float ang_x_prev, ang_y_prev;
//-----

//Inicializacion
//-----
void setup() {

    Serial.begin(115200);

    // Inicializacion del temporizador
    timer = timerBegin(0, 80, true);
    timerAttachInterrupt(timer, &onTimer, true);
    timerAlarmWrite(timer, 1000000/freq, true);
    timerAlarmEnable(timer);

    //Inicializacion IMU
    Serial2.begin(9600, SERIAL_8N1, rx2Pin, tx2Pin);
    Wire.begin(); //Iniciando I2C
    sensor.initialize(); //Iniciando el sensor

    // Valores de calibracion
    sensor.setXAccelOffset(4247);
    sensor.setYAccelOffset(796);
    sensor.setZAccelOffset(807);

    sensor.setXGyroOffset(111);
    sensor.setYGyroOffset(83);
    sensor.setZGyroOffset(177);
```



```
//Interruptor WIFI
PinMode(LED_1, OUTPUT);
pinMode(LED_2, OUTPUT);
digitalWrite(LED_1, LOW);
digitalWrite(LED_2, HIGH);

// Configura la IP fija antes de conectar a Wi-Fi
if (!WiFi.config(local_IP, gateway, subnet)) {
  Serial.println("Configuración de IP estática fallida");
}

WiFi.begin("red", "contrasena");
while (WiFi.status() != WL_CONNECTED) {
  delay(1000);
  Serial.println("Conectando a la red Wi-Fi...");
}
Serial.println("Conexión Wi-Fi establecida");
// Imprime la dirección IP asignada al dispositivo
Serial.print("Dirección IP: ");
Serial.println(WiFi.localIP());

// Registra las funciones de control de LED para las rutas correspondientes
server.on("/", handleRoot);
server.on("/led1/on", handleLedOn);
server.on("/led1/off", handleLedOff);
server.on("/led2/on", handleLedOn);
server.on("/led2/off", handleLedOff);

// Inicia el servidor web
server.begin();
Serial.println("Servidor web iniciado");

delay(1000);
}

// Bucle principal
//-----
void loop() {
if (isrTask == 1){

  // Wifi
  //-----
  server.handleClient();
  //-----

  // IMU
  //-----
  sensor.getAcceleration(&ax, &ay, &az);
  sensor.getRotation(&gx, &gy, &gz);

  dt = (millis()-tiempo_prev)/1000.0;
  tiempo_prev=millis();

  float ax_m_s2 = ax * (9.81/16384.0);
  float ay_m_s2 = ay * (9.81/16384.0);
  float az_m_s2 = az * (9.81/16384.0);
```



```
float gx_deg_s = gx * (250.0/32768.0);
float gy_deg_s = gy * (250.0/32768.0);
float gz_deg_s = gz * (250.0/32768.0);

//Calcular los angulos de inclinacion:
float accel_ang_x=atan(ax_m_s2/sqrt(pow(ay_m_s2,2) +
pow(az_m_s2,2)))*(180.0/3.14);
float accel_ang_y=atan(ay_m_s2/sqrt(pow(ax_m_s2,2) +
pow(az_m_s2,2)))*(180.0/3.14);

//Calcular angulo de rotación con giroscopio y filtro complemento
ang_x = 0.98*(ang_x_prev+(gx/131)*dt) + 0.02*accel_ang_x;
ang_y = 0.98*(ang_y_prev+(gy/131)*dt) + 0.02*accel_ang_y;

ang_x_prev=ang_x;
ang_y_prev=ang_y;

temperature = sensor.getTemperature();
float celsius = (float)temperature / 340.00 + 36.53;
//-----

//Mensaje enviado por puerto serie
Serial.print("A");    Serial.print('\t');
Serial.print(time_buffer);    Serial.print('\t');
Serial.print(ax_m_s2);    Serial.print('\t');
Serial.print(ay_m_s2);    Serial.print('\t');
Serial.print(az_m_s2);    Serial.print('\t');
Serial.print(gx_deg_s);    Serial.print('\t');
Serial.print(gy_deg_s);    Serial.print('\t');
Serial.print(gz_deg_s);    Serial.print('\t');
Serial.print(celsius);    Serial.print('\t');
Serial.print(ang_x);    Serial.print('\t');
Serial.print(ang_y);    Serial.print('\t');

if (counter != freq){

    if (horas < 10) {
        Serial.print("0");
    }
    Serial.print(horas);
    Serial.print(":");
    if (minutos < 10) {
        Serial.print("0");
    }
    Serial.print(minutos);
    Serial.print(":");
    if (segundos < 10) {
        Serial.print("0");
    }
    Serial.print(segundos);
    Serial.print('\t');

    //Imprimir Latitud
    Serial.print(gradosLat);
    Serial.print("°");
    Serial.print(minutosLat);
    Serial.print("'");
```



```
Serial.print((segundosLat*60)/100000 , 4);
Serial.print("");
Serial.print(nmea[2]);
Serial.print('\t');

//Imprimir Longitud
Serial.print(gradosLon);
Serial.print("°");
Serial.print(minutosLon);
Serial.print("'");
Serial.print((segundosLon*60)/100000, 4);
Serial.print("");
Serial.print(nmea[4]);
Serial.print('\t');

//Altitud
Serial.print(nmea[8]);   Serial.print(" m");   Serial.print('\t');

//Nº de satelites
Serial.print(nmea[6]);   Serial.print('\t');

//Calidad de la señal
Serial.print(nmea[5]);   Serial.print('\t');

//Velocidad en Km/h
Serial.print(nmea_1[6]);Serial.println(" km/h");
}

stringplace = 0;
pos = 0;
stringplace_1 = 0;
pos_1 = 0;
counter++;
isrTask = 0;
}
// Ajustar diferencias de frecuencias entre GPS y resto de sensores
  if (counter == freq+1){

//GPS
//-----
char GPS_msg;

unsigned long tiempo_actual = millis();

if (Serial2.available()){
  GPS_msg = Serial2.read();
}

if (Serial2.find("$GNGGA,")){
  GNGGA_msg = Serial2.readStringUntil('\n');
  for (int i = 0; i < GNGGA_msg.length(); i++) {
    if (GNGGA_msg.substring(i, i + 1) == ",") {
      nmea[pos] = GNGGA_msg.substring(stringplace, i);
      stringplace = i + 1;
      pos++;
    }
  }
  if (i == GNGGA_msg.length() - 1) {
```

```
        nmea[pos] = GNGGA_msg.substring(stringplace, i);
    }
}

if (Serial2.find("$GNVTG,")){
GNVTG_msg = Serial2.readStringUntil('\n');
for (int j = 0; j < GNVTG_msg.length(); j++) {
    if (GNVTG_msg.substring(j, j + 1) == ",") {
        nmea_1[pos_1] = GNVTG_msg.substring(stringplace_1, j);
        stringplace_1 = j + 1;
        pos_1++;
    }
    if (j == GNVTG_msg.length() - 1) {
        nmea_1[pos_1] = GNVTG_msg.substring(stringplace_1, j);
    }
}
}

//Conversión de los datos obtenidos
tiempo.UTC = nmea[0];
horas = tiempo.UTC.substring(0, 2).toInt();
minutos = tiempo.UTC.substring(2, 4).toInt();
segundos = tiempo.UTC.substring(4, 6).toInt();
latitud = nmea[1];
gradosLat = latitud.substring(0, 2).toInt();
minutosLat = latitud.substring(2, 4).toInt();
segundosLat = latitud.substring(5).toInt();
longitud = nmea[3];
gradosLon = longitud.substring(1, 3).toInt();
minutosLon = longitud.substring(3, 5).toInt();
segundosLon = longitud.substring(6).toFloat();

mostrado_gps = true;
counter=0;
}
}
```