



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**Diseño de un sistema
económico IoT de
monitorización de
invernaderos de cannabis
medicinal**



Presentado por José Luis Caballero
Martínez-Quintanilla
en Universidad de Burgos — 16 de febrero de
2024

Tutor: Alejandro Merino Gómez
Tutor: Carlos Cambra Baseca



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. Alejandro Merino Gómez, profesor del departamento de Digitalización, área de Systems Engineering and Automation y D. Carlos Cambra Baseca, profesor del departamento de Digitalización del área de Computer Science and Artificial Intelligence,

Exponen:

Que el alumno D. José Luis Caballero Martínez-Quintanilla, con DNI 48471169-A, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Diseño de un sistema económico IoT de monitorización de invernaderos de cannabis medicinal.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 16 de febrero de 2024

Vº. Bº. del Tutor:

D. Merino Gómez, Alejandro

Vº. Bº. del co-tutor:

D. Cambra Baseca, Carlos

Resumen

El presente trabajo de fin de grado aborda el diseño de un sistema económico basado en el Internet de las cosas (IoT) para la monitorización de invernaderos destinados al cultivo de cannabis medicinal. El objetivo principal es mejorar la eficiencia y la calidad del cultivo mediante la implementación de sensores y dispositivos conectados que permitan la recolección y análisis de datos en tiempo real.

El hardware seleccionado para este proyecto incluye la Raspberry Pi Pico W como unidad central, una pantalla OLED 128x64 para la visualización de información, el sensor DHT22 para la medición de la temperatura y humedad ambiente, el sensor BH1750 para evaluar la intensidad lumínica, y un sensor de humedad de suelo para monitorear las condiciones de la tierra.

A lo largo del trabajo, se detalla el proceso de integración de estos componentes, se describen las tecnologías utilizadas para la comunicación y el manejo de datos, y se presenta la interfaz de usuario diseñada para la visualización de información relevante. Se destacan también las consideraciones económicas que han llevado a la elección de cada componente, buscando una solución asequible sin comprometer la calidad de los resultados.

Los resultados obtenidos demuestran la viabilidad y eficacia del sistema propuesto, ofreciendo a los agricultores de cannabis medicinal un instrumento práctico y accesible para mejorar la gestión de sus invernaderos. Este trabajo contribuye al campo emergente de la agricultura inteligente y sostenible, abriendo posibilidades para futuras investigaciones y aplicaciones en el ámbito de la monitorización agrícola basada en IoT.

Descriptores

Raspberry Pi Pico W, Micropython, Autónomo, Sistema Domótico, Bot, Telegram, Python, MQTT, SQL ...

Abstract

This project deals with the design of an economic system based on the Internet of Things (IoT) for the monitoring of greenhouses for the cultivation of medical cannabis. The main objective is to improve the efficiency and quality of the crop through the implementation of sensors and connected devices that allow the collection and analysis of data in real time.

The hardware selected for this project includes the Raspberry Pi Pico W as the central unit, a 128x64 OLED screen for displaying information, the DHT22 sensor for measuring ambient temperature and humidity, the BH1750 sensor for evaluating light intensity, and a soil moisture sensor for monitoring soil conditions.

Throughout the paper, the integration process of these components is detailed, the technologies used for communication and data management are described, and the user interface designed for the visualization of relevant information is presented. It also highlights the economic considerations that led to the choice of each component, seeking an affordable solution without compromising the quality of the results.

The results obtained demonstrate the feasibility and effectiveness of the proposed system, offering medical cannabis farmers a practical and accessible tool to improve the management of their greenhouses. This work contributes to the emerging field of smart and sustainable agriculture, opening possibilities for future research and applications in the field of IoT-based agricultural monitoring.

Keywords

Raspberry Pi Pico W, Micropython, Autonomous, Domotic System, Bot, Telegram, Python, MQTT, SQL...

Índice general

Índice general	iii
Índice de figuras	v
Índice de tablas	vii
1. Introducción	1
2. Objetivos del proyecto	3
2.1. Objetivos del Software	3
2.2. Objetivos Técnicos	4
3. Conceptos teóricos	5
3.1. Internet de las cosas (IoT)	5
3.2. Raspberry Pi Pico W	6
3.3. MicroPython	6
3.4. GPIO	6
3.5. Sensores (DHT22, BH1750, Sensor de Humedad de Suelo)	7
3.6. Interfaz de Usuario	7
3.7. Protocolos de Comunicación	7
3.8. WiFi	9
3.9. Servidor LAMP	9
4. Técnicas y herramientas	11
4.1. Entorno Software	11
4.2. Control de datos	13
4.3. Entorno Hardware	14

4.4. Metodologías	15
4.5. Entorno de desarrollo del Proyecto	16
4.6. Entorno físico	20
5. Aspectos relevantes del desarrollo del proyecto	31
5.1. Motivación del proyecto	31
5.2. Formación necesaria	32
5.3. Metodología	33
5.4. Desarrollo del proyecto	38
6. Trabajos relacionados	53
6.1. Fortalezas y debilidades este proyecto	55
7. Conclusiones y Líneas de trabajo futuras	57
7.1. Conclusiones	57
7.2. Líneas de trabajo futuras	58
Bibliografía	61

Índice de figuras

3.1.	LAMP.	9
4.1.	IDE Thonny.	12
4.2.	Esquema de conexiones diseñado con Fritzing.	14
4.3.	Raspberry Pi Pico W Pinout.	21
4.4.	Vistas Frontal y posterior de Raspberry Pi Pico W.	21
4.5.	Pantalla oled.	22
4.6.	Vista frontal del sensor BH1750.	23
4.7.	Lado posterior del sensor BH1750.	24
4.8.	Sensor DHT22.	25
4.9.	Sensor de humedad de suelo Higrometro V1.2.	26
4.10.	Módulo led RGB KY-016.	27
4.11.	Enrutador WiFi USB 4G LTE.	28
4.12.	Batería externa 20000 mAh.	29
4.13.	Panel solar y soporte ajustable.	30
5.1.	Raspberry Pi Pico W conectada a sensores y una pantalla oled.	38
5.2.	Raspberry Pi Pico W y sensores conectados mediante un proto-board.	39
5.3.	Pantalla oled conectada a la Raspberry Pi Pico W.	39
5.4.	Alertas del bot de Telegram.	40
5.5.	Servicios instalados y operativos en el servidor LAMP.	41
5.6.	Se ha implementado la librería umqtt en la Raspberry Pi Pico W.	41
5.7.	Data almacenada en la base de datos.	42
5.8.	Consulta a la base de datos mediante comandos usando Telegram.	42
5.9.	El led rojo y el color naranja en la aplicación InverIoT indican que se ha sobrepasado el umbral de luz.	43
5.10.	Al hacer clic en activar se enciende el led de color verde representando que estamos activando un mecanismo.	43

5.11. Página Web InverIoT desarrollada con Node.js.	45
5.12. Conexiones.	45
5.13. Aplicación de escritorio InverIoT	46
5.14. Enviando órdenes mediante MQTT la aplicación recibe datos en tiempo real.	46
5.15. Para mostrar el historial de datos la aplicación extrae información de una base de datos almacenada en un servidor LAMP.	47
5.16. Botones para activar o desactivar mecanismos.	47
5.17. Historial de los datos mostrara en la aplicación.	48
5.18. Gráfica mostrando los datos en un intervalo de fechas.	48
5.19. Intensidad de luz superando los umbrales.	49
5.20. Vista del historial, al cargar, muestra los datos del día actual.	49
5.21. Extrae los datos de la base de datos para mostrar el historial.	50
5.22. Mediante MQTT extrae los datos para mostrarlos en tiempo real.	50
5.23. MQTT con Node.js para almacenar datos de los sensores.	51

Índice de tablas

4.1.	Características de la Raspberry Pi Pico W.	20
4.2.	Características de la pantalla oled	22
4.3.	Características del sensor de luz BH1750.	23
4.4.	Características del sensor de temperatura y humedad DHT22. .	25
4.5.	Características del sensor de humedad de suelo.	26
4.6.	Características del módulo led RGB KY-016.	27
4.7.	Características del enrutador WiFi USB 4G LTE.	28
4.8.	Características de la batería externa 20000 mAh.	29
4.9.	Características del panel solar.	30
5.1.	Características del servidor HP ProLiant.	40
5.2.	Umbrales ideales para un invernadero de cannabis medicinal. .	44
6.1.	Comparativa de las características de los proyectos.	56

1. Introducción

El panorama en constante cambio de la industria del cannabis medicinal ha generado una creciente necesidad de optimizar los métodos de cultivo para garantizar la consistencia y calidad de los productos. En este contexto, el Internet de las cosas (IoT) emerge como una herramienta clave para transformar la gestión de invernaderos, permitiendo una monitorización en tiempo real y decisiones informadas por parte de los agricultores.

Este trabajo se centra en el diseño de un sistema asequible de monitorización basado en IoT para invernaderos de cannabis medicinal. La elección de hardware, liderada por la Raspberry Pi Pico W como unidad central, se fundamenta en la capacidad de esta plataforma para equilibrar eficiencia y costos. Sensores especializados, como el DHT22 para temperatura y humedad, el BH1750 para intensidad lumínica, y un sensor de humedad de suelo, complementan la infraestructura del sistema.

Más allá de la mejora en la eficiencia del cultivo de cannabis medicinal, este proyecto busca situarse en la vanguardia de la agricultura inteligente y sostenible. La combinación de IoT con prácticas agrícolas avanzadas tiene el potencial de transformar radicalmente la forma en que se gestionan los cultivos, promoviendo la sostenibilidad y ofreciendo una solución práctica para agricultores de diversos niveles de recursos.

En resumen, este trabajo se adentra en el diseño de un sistema completo y accesible para la monitorización de invernaderos, utilizando la innovación tecnológica como catalizador para mejorar la calidad de los cultivos y contribuir al avance de la agricultura moderna.

2. Objetivos del proyecto

2.1. Objetivos del Software

- **Desarrollo del Sistema de Adquisición de Datos:** Implementar un sistema eficiente de adquisición de datos que pueda recopilar información precisa proveniente de los sensores (DHT22, BH1750, sensor de humedad de suelo) instalados en el invernadero.
- **Diseño de la Interfaz de Usuario:** Desarrollar una interfaz de usuario intuitiva y fácil de usar, que permita a los usuarios visualizar en tiempo real los datos recopilados y tomar decisiones informadas sobre el control del entorno del invernadero.
- **Implementación de Protocolos de Comunicación:** Establecer protocolos de comunicación eficientes para la transmisión de datos entre los sensores y la Raspberry Pi Pico W, así como para la conexión con otros dispositivos o sistemas externos si es necesario.

2.2. Objetivos Técnicos

- **Integración de Hardware:** Seleccionar, configurar e integrar de manera óptima el hardware necesario (Raspberry Pi Pico W, pantalla OLED 128x64, sensores) para garantizar la estabilidad y la eficacia del sistema.
- **Eficiencia Energética:** Implementar estrategias de programación y configuración de hardware que optimicen el consumo de energía, asegurando una operación sostenible del sistema en entornos con recursos limitados. Precisamente la Raspberry Pi Pico W es conocida por su bajo consumo energético.
- **Validación en Entorno Real:** Realizar pruebas exhaustivas del sistema en un entorno de cultivo de cannabis medicinal real, evaluando su rendimiento, confiabilidad y adaptabilidad a condiciones variables.
- **Consideraciones Económicas:** Evaluar y optimizar los costos asociados al desarrollo e implementación del sistema, garantizando la viabilidad económica para agricultores de diferentes escalas y recursos.
- **Documentación Completa:** Generar una documentación detallada que abarque el diseño, la implementación y la configuración del sistema, facilitando la comprensión y la replicación por parte de otros interesados. Estos objetivos, tanto del software como técnicos, se plantean con el propósito de cumplir con los requisitos del proyecto y garantizar la efectividad y utilidad del sistema de monitorización propuesto para invernaderos de cannabis medicinal.

3. Conceptos teóricos

3.1. Internet de las cosas (IoT)

El Internet de las cosas (IoT) se refiere a la red de dispositivos físicos que están integrados con tecnología, software, sensores y conectividad de red para recopilar y compartir datos. Estos dispositivos, también llamados **objetos inteligentes** o **dispositivos conectados**, pueden interactuar entre sí y con sistemas informáticos en red para realizar tareas específicas, recopilar información y ofrecer funcionalidades avanzadas.

Características clave del IoT:

- **Conectividad:** Los dispositivos IoT están equipados con tecnologías de comunicación, como Wi-Fi, Bluetooth, Zigbee o incluso 5G, que les permiten conectarse a redes y compartir datos.
- **Sensores y Actuadores:** Los dispositivos IoT suelen estar equipados con sensores para recopilar datos del entorno circundante. Además, algunos dispositivos también incorporan actuadores que les permiten realizar acciones físicas en respuesta a comandos.
- **Inteligencia y Procesamiento de Datos:** Muchos dispositivos IoT cuentan con capacidad de procesamiento local o dependen de la nube para analizar y procesar datos recopilados. La inteligencia artificial y el aprendizaje automático también se aplican para extraer información significativa de grandes conjuntos de datos.
- **Interconexión:** La capacidad de interconectar dispositivos IoT permite que trabajen juntos de manera coordinada, compartiendo información y actuando en función de datos colectivos.

- **Aplicaciones Diversas:** El IoT se aplica en una amplia variedad de campos, como la agricultura inteligente, la salud, la domótica, la industria manufacturera, la logística y más. Esto permite la automatización y mejora de procesos en diversas industrias.

- **Seguridad y Privacidad:** Dado que los dispositivos IoT manejan datos sensibles, la seguridad y la privacidad son aspectos críticos. Se implementan medidas de seguridad para proteger la integridad y la confidencialidad de los datos transmitidos y almacenados.

El Internet de las cosas tiene un impacto significativo en la forma en que interactuamos con el entorno, mejorando la eficiencia, la toma de decisiones y la calidad de vida. A medida que la tecnología avanza, el alcance y la influencia del IoT continúan expandiéndose, creando un ecosistema cada vez más interconectado.

3.2. Raspberry Pi Pico W

La Raspberry Pi Pico W [11] es una placa de desarrollo de bajo costo que combina versatilidad y conectividad inalámbrica. Se elige como unidad central para el sistema, actuando como el cerebro que controla y coordina la recopilación y transmisión de datos.

3.3. MicroPython

MicroPython [7] es una implementación del lenguaje de programación Python diseñada para sistemas embebidos y microcontroladores como la Raspberry Pi Pico W. Permite una programación sencilla y eficiente de la unidad central, facilitando el desarrollo de aplicaciones en entornos limitados.

3.4. GPIO

GPIO [10], por sus siglas en inglés, General Purpose Input/Output, se refiere a los pines de propósito general presentes en microcontroladores y sistemas embebidos. Estos pines son versátiles y pueden configurarse tanto como entradas como salidas, permitiendo la interacción del microcontrolador con el entorno externo.

- **Entradas (Input):** En modo de entrada, un pin GPIO puede recibir señales eléctricas del exterior, como las provenientes de sensores o

3.5. SENSORES (DHT22, BH1750, SENSOR DE HUMEDAD DE SUELO)

interruptores. Estos pines detectan y leen el estado de la señal, que puede ser alto (1) o bajo (0), dependiendo de la presencia de voltaje.

- **Salidas (Output):** En modo de salida, un pin GPIO puede enviar señales eléctricas al exterior, como las necesarias para controlar actuadores, LEDs u otros dispositivos. El microcontrolador puede controlar el estado del pin, estableciendo un voltaje alto o bajo según sea necesario.

La flexibilidad de los pines GPIO permite a los desarrolladores adaptar el comportamiento del microcontrolador a una amplia variedad de aplicaciones. La manipulación directa de estos pines a través de código facilita la interconexión con componentes externos y la implementación de funciones específicas en proyectos de hardware. En el contexto de la Raspberry Pi Pico W [11] y otros dispositivos similares, la gestión de los pines GPIO es esencial para controlar y coordinar la interacción con sensores, actuadores y otros periféricos.

3.5. Sensores (DHT22, BH1750, Sensor de Humedad de Suelo)

Los sensores seleccionados desempeñan roles cruciales en la monitorización del entorno del invernadero. El DHT22 [41] mide la temperatura y humedad ambiente, el BH1750 [17] evalúa la intensidad lumínica, y el sensor de humedad de suelo [8] monitorea las condiciones de la tierra.

3.6. Interfaz de Usuario

La interfaz de usuario proporciona una plataforma visual para la interpretación de datos recopilados. Su diseño debe ser intuitivo, permitiendo acceder y comprender fácilmente la información relevante sobre el estado del invernadero.

3.7. Protocolos de Comunicación

Los protocolos de comunicación son conjuntos de reglas y estándares que especifican cómo los dispositivos deben intercambiar información entre sí. Estos protocolos definen el formato, la secuencia y el significado de los mensajes que se envían y reciben, facilitando la comunicación efectiva

entre sistemas o dispositivos electrónicos. Los protocolos son esenciales para garantizar la interoperabilidad y la correcta transmisión de datos en entornos tecnológicos diversos.

Existen diferentes tipos de protocolos de comunicación, cada uno diseñado para propósitos específicos y adaptado a diferentes situaciones. Algunos ejemplos comunes incluyen:

- **Protocolos de Red:** Estos protocolos se utilizan para la transmisión de datos a través de redes de computadoras. Ejemplos incluyen el Protocolo de Internet (IP) para direccionamiento y enrutamiento, y el Protocolo de Control de Transmisión (TCP) para la gestión de conexiones fiables.
- **Protocolos de Comunicación Inalámbrica:** Para la comunicación sin cables, existen protocolos específicos como el estándar IEEE 802.11 (Wi-Fi) [23] para redes inalámbricas y el Bluetooth para la conexión de dispositivos cercanos.
- **Protocolos de Comunicación Serie:** En entornos donde se requiere una comunicación punto a punto, se utilizan protocolos serie como UART (Universal Asynchronous Receiver-Transmitter) [24] y SPI (Serial Peripheral Interface) [25].
- **Protocolos de Comunicación de Periféricos:** Para la interconexión de periféricos y componentes electrónicos, se utilizan protocolos como I2C (Inter-Integrated Circuit) y SPI para la transmisión de datos entre microcontroladores, sensores y otros dispositivos.
- **Protocolos de Aplicación:** Estos protocolos se centran en la comunicación a nivel de aplicación y pueden incluir estándares como HTTP (Hypertext Transfer Protocol) [9] para la transmisión de datos en la web y MQTT (Message Queuing Telemetry Transport) [26] para la comunicación en el Internet de las cosas (IoT).

En cuanto a normativas, algunas organizaciones como el IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) [20] establecen estándares para varios protocolos de comunicación, garantizando la consistencia y la interoperabilidad en la industria. Por ejemplo, el IEEE 802.3 [22] define estándares para Ethernet, mientras que el IEEE 802.15 [21] se centra en protocolos para redes inalámbricas de área personal (WPAN), como Bluetooth. Estas normativas son esenciales para la creación de sistemas y dispositivos que puedan comunicarse de manera efectiva y sin problemas.

3.8. WiFi

Wi-Fi, derivado de **Wireless Fidelity**, es una tecnología de comunicación inalámbrica que permite la conexión de dispositivos electrónicos a una red local o a Internet sin necesidad de cables físicos. Esta tecnología utiliza ondas de radio para la transmisión de datos entre dispositivos compatibles, siguiendo los estándares establecidos por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), específicamente dentro de la familia de normas 802.11.

La norma [IEEE 802.11 \[23\]](#) abarca diversas versiones, cada una con mejoras y características específicas. Por ejemplo, las variantes más comunes incluyen 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac, y 802.11ax. Cada una de estas normas define aspectos técnicos como la velocidad de transmisión, el rango de frecuencias utilizadas, y las características de seguridad.

La conexión a una red Wi-Fi implica que los dispositivos inalámbricos cumplen con estas normas para garantizar una comunicación eficiente y segura. Los routers Wi-Fi, que actúan como puntos de acceso, son fundamentales para establecer y gestionar estas conexiones inalámbricas, permitiendo que múltiples dispositivos se conecten y comparten recursos en un área determinada.

3.9. Servidor LAMP

Un servidor LAMP es un entorno de desarrollo web que utiliza Linux [\[29\]](#) como sistema operativo, Apache [\[3\]](#) como servidor web, MySQL [\[33\]](#) como sistema de gestión de bases de datos y PHP [\[38\]](#) como lenguaje de programación para el desarrollo de aplicaciones dinámicas. Este conjunto de tecnologías trabaja de manera integrada para permitir la creación y ejecución de aplicaciones web de manera eficiente y robusta. Este stack LAMP es ampliamente utilizado para el desarrollo de aplicaciones web y sitios dinámicos.



Figura 3.1: LAMP.

4. Técnicas y herramientas

A lo largo del desarrollo del proyecto, se han empleado diversas tecnologías, herramientas y elementos esenciales que requieren familiaridad antes de avanzar en la ejecución del proyecto. La elección de estas opciones en lugar de otras se basa en una evaluación detallada, la cual queda documentada en esta sección con el propósito de brindar una justificación fundamentada.

4.1. Entorno Software

Entorno de desarrollo Micropython

- **Herramientas valoradas:** [Thonny IDE](#), [Mu Editor](#), [Upycraft](#), [PyCharm](#).

- **Herramienta elegida:** [Thonny IDE](#).

Thonny IDE se presenta como la opción más idónea, proporcionando una combinación de simplicidad, integración nativa con MicroPython, herramientas de depuración eficaces y un sólido soporte comunitario, todo lo cual contribuye a un entorno de desarrollo eficiente y centrado en el usuario para la programación en la Raspberry Pi Pico W con MicroPython.

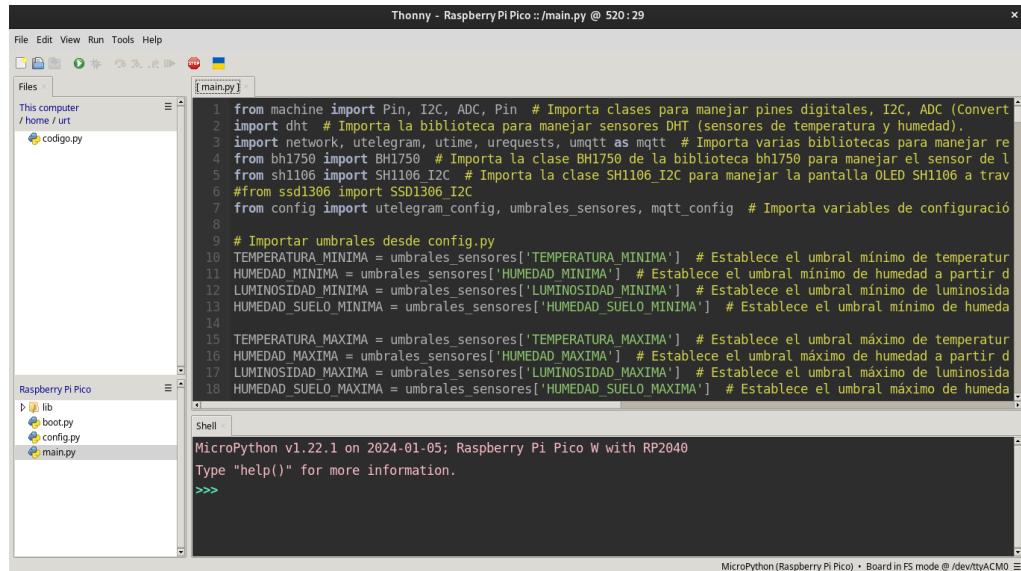


Figura 4.1: IDE Thonny.

Entorno de desarrollo Python

- Herramientas valoradas: Google colab y Jupyter Notebook.
 - Herramienta elegida: Jupyter Notebook.

Jupyter Notebook [35] es una herramienta de código abierto poderosa que facilita la creación y compartición de documentos interactivos. Estos documentos pueden contener código ejecutable, visualizaciones, texto narrativo y otros elementos. Opto por utilizar Jupyter Notebook en mi Trabajo de Fin de Grado debido a su versatilidad y capacidad para aprovechar los entornos virtuales que he configurado en mi computadora. Este entorno se ha vuelto fundamental para llevar a cabo análisis de datos, especialmente haciendo uso intensivo de la biblioteca Pandas [37] para manipulación y procesamiento de datos.

4.2. Control de datos

APIS

- **API de Telegram:** La API de Telegram [45], accesible a través de la URL <https://api.telegram.org/bot<token>>, juega un papel fundamental en la integración de bots de Telegram [46] con el sistema. Permite la interacción bidireccional entre el sistema y los usuarios a través de la plataforma de mensajería Telegram. Al utilizar esta API, se establece una conexión segura para enviar y recibir mensajes, comandos y datos entre el sistema y los usuarios mediante la implementación de funciones específicas proporcionadas por Telegram. Esto posibilita la notificación remota, la recopilación de datos y la activación de acciones automatizadas, contribuyendo así a la funcionalidad integral del sistema de control de datos. Esta API fue usada en el código MicroPython.

- **API WorldTime** La API de WorldTime [49], accesible mediante la URL <http://worldtimeapi.org/api/ip>, ofrece información precisa sobre la hora mundial basada en la ubicación asociada a la dirección IP del usuario. Al realizar una solicitud a esta API, se obtiene información detallada sobre la hora actual, la fecha, el huso horario y otros datos relacionados con la hora en la ubicación correspondiente a la dirección IP consultada. Esta API es valiosa para aplicaciones que requieren sincronización horaria precisa y la visualización de la hora actual en diferentes regiones del mundo. Su facilidad de acceso y respuesta estructurada la convierten en una herramienta útil para integrar la información horaria global en aplicaciones y sistemas. Esta API fue usada en el código MicroPython.

4.3. Entorno Hardware

Fritzing

Fritzing [13] es una herramienta de diseño electrónico de código abierto que facilita la creación de esquemas, prototipos y layouts de placas de circuito impreso (PCB). Diseñado para usuarios, desde principiantes hasta expertos, Fritzing ofrece una interfaz intuitiva y gráfica que permite la conexión visual de componentes electrónicos, como sensores, actuadores y placas de desarrollo. Su funcionalidad principal abarca desde la creación de esquemas eléctricos hasta la generación de archivos Gerber para la fabricación de PCB. Fritzing se destaca por su accesibilidad y capacidad para convertir conceptos de diseño en proyectos electrónicos tangibles, lo que lo convierte en una herramienta valiosa para ingenieros, diseñadores y entusiastas que desean visualizar y materializar sus ideas electrónicas.

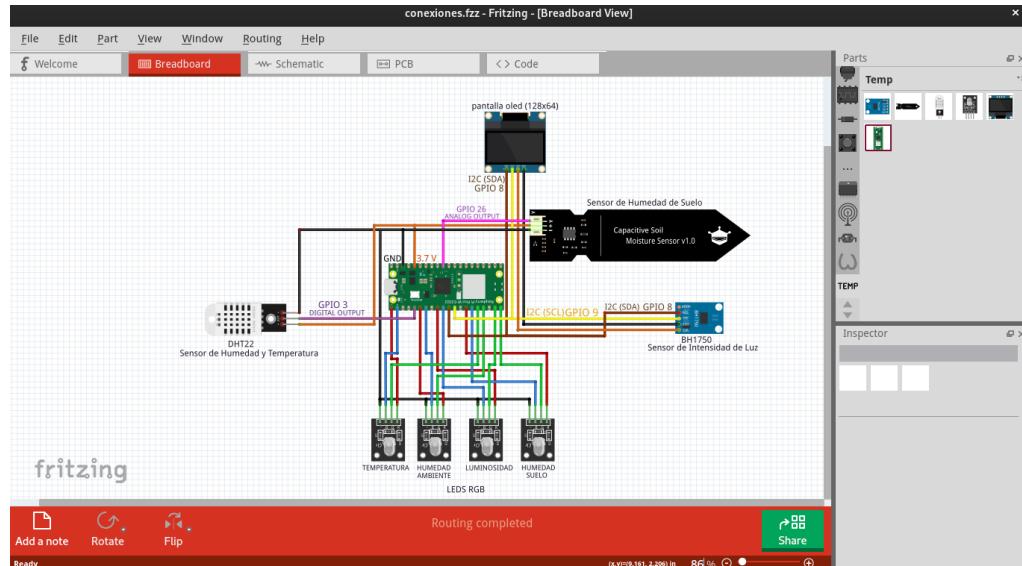


Figura 4.2: Esquema de conexiones diseñado con Fritzing.

4.4. Metodologías

Modelo de prototipos

El modelo de prototipos [18] es una metodología de desarrollo de software que se centra en la creación temprana y rápida de versiones simplificadas de un sistema para obtener retroalimentación y comprensión de los requisitos del usuario. En lugar de esperar hasta la fase final del desarrollo para presentar un producto completo, el enfoque de prototipos permite construir versiones iterativas y evolutivas del sistema.

El proceso típico del modelo de prototipos implica las siguientes etapas:

- **Definición de Requisitos:** Se identifican y documentan los requisitos iniciales del sistema.
- **Desarrollo del Prototipo:** Se construye un prototipo inicial que representa visualmente las funcionalidades clave del sistema.
- **Evaluación y Retroalimentación:** Los usuarios y las partes interesadas evalúan el prototipo y proporcionan retroalimentación sobre su diseño y funcionalidad.
- **Refinamiento Iterativo:** Con base en la retroalimentación recibida, se realizan ajustes y mejoras en el prototipo, creando iteraciones sucesivas.
- **Validación de Requisitos:** La validación continua de los prototipos ayuda a refinar y validar los requisitos del sistema a lo largo del proceso.

Este enfoque iterativo y colaborativo permite una mayor flexibilidad en la adaptación a cambios en los requisitos del usuario y facilita la comunicación entre los desarrolladores y los stakeholders. El modelo de prototipos es particularmente útil en situaciones donde los requisitos no están completamente claros desde el principio o cuando se busca una comprensión más profunda de las necesidades del usuario antes de la implementación final del sistema.

Metodología cascada

La metodología cascada [16], también conocida como el modelo de desarrollo en cascada, es una metodología de desarrollo de software que sigue un enfoque lineal y secuencial. Aunque ha sido utilizada en proyectos de software

en el pasado, en la actualidad se considera menos flexible y adaptable en comparación con metodologías más modernas y ágiles. En el modelo cascada, el proceso de desarrollo sigue una secuencia fija de fases, que generalmente incluyen:

1. Requisitos
2. Diseño
3. Implementación
4. Pruebas
5. Despliegue
6. Mantenimiento

Cada fase depende de la finalización exitosa de la fase anterior, y los cambios en las etapas posteriores pueden ser difíciles y costosos de implementar una vez que el proyecto ha avanzado. Este enfoque puede no ser el más adecuado para proyectos donde los requisitos pueden cambiar con frecuencia o donde la adaptabilidad es esencial.

En muchos proyectos actuales, especialmente en el ámbito académico, se prefiere el uso de metodologías más ágiles y flexibles, como Scrum, Kanban o el enfoque de desarrollo iterativo e incremental.

En conclusión, aunque la metodología cascada ha sido utilizada en el pasado, hoy en día muchas organizaciones y proyectos prefieren metodologías más ágiles y adaptables. La elección de la metodología dependerá de los requisitos y características específicas de cada proyecto de TFG.

4.5. Entorno de desarrollo del Proyecto

Control de versiones o CVS, Concurrent Versioning System

- Herramientas valoradas: [Git](#), [SVN](#).
- Herramienta elegida: [Git](#).

Git [14] es un sistema de control de versiones distribuido, ampliamente utilizado en el desarrollo de software y es conocido por su velocidad, flexibilidad y eficiencia en la gestión de versiones de código fuente.

Comparación con SVN:

Subversion (SVN) [43] es otro sistema de control de versiones, pero sigue un modelo centralizado, a diferencia del modelo distribuido de Git. Aquí hay algunas diferencias clave y ventajas de Git sobre SVN:

- **Descentralización:** Git es completamente descentralizado, lo que significa que cada usuario tiene una copia completa del repositorio con todo su historial. En SVN, los usuarios dependen del servidor central para muchas operaciones.
- **Ramificación y Fusiones Rápidas:** Git permite ramificaciones ligeras y fusiones rápidas, lo que facilita el desarrollo paralelo y la gestión de características en diferentes ramas. SVN maneja ramificaciones y fusiones, pero históricamente ha sido menos eficiente en comparación con Git.
- **Eficiencia en Red:** Git es más eficiente en términos de ancho de banda y operaciones de red, ya que las operaciones se realizan localmente en la mayoría de los casos. SVN, al depender más del servidor central, puede ser más lento en operaciones que implican la red.
- **Historial Completo y Offline:** Git almacena el historial completo localmente, lo que permite el trabajo fuera de línea. SVN requiere acceso al servidor para recuperar el historial completo.
- **Flexibilidad y Escalabilidad:** Git es conocido por su flexibilidad y escalabilidad, especialmente en proyectos grandes y complejos. SVN puede encontrar limitaciones en proyectos extensos y complejos.

Hosting del Repositorio

- **Herramientas valoradas:** [Github](#) y [Bitbucket](#).
- **Herramienta elegida:** [Github](#).

GitHub [15] es una plataforma de desarrollo colaborativo basada en Git que facilita el alojamiento y la colaboración en proyectos de software.

Bitbucket [5] es otra plataforma de desarrollo colaborativo basada en Git, pero hay algunas diferencias clave. Aquí se exploran estas diferencias y se destaca una ventaja de GitHub sobre Bitbucket:

- **Visibilidad del Código Fuente:** GitHub ha sido históricamente más popular para proyectos de código abierto, y los repositorios públicos en GitHub suelen tener una mayor visibilidad y participación de la comunidad en comparación con Bitbucket.
- **Comunidad y Desarrollo Abierto:** GitHub se ha convertido en el estándar de facto para proyectos de código abierto, y muchos desarrolladores buscan activamente proyectos en GitHub. Esto hace que sea más fácil para los proyectos atraer colaboradores y contribuyentes.
- **Integraciones y Ecosistema:** GitHub tiene una amplia gama de integraciones y herramientas de terceros que son ampliamente utilizadas en la comunidad de desarrollo. La rica integración con servicios como Travis CI, CircleCI y otros facilita el desarrollo y la automatización.
- **Herramientas de Colaboración:** Mientras que Bitbucket ofrece características similares, GitHub es generalmente preferido por su interfaz de usuario intuitiva, herramientas de revisión de código más robustas y una experiencia general más pulida para la colaboración.
- **Repositorios Públicos Gratuitos:** GitHub permite la creación de repositorios públicos de forma gratuita, lo que fomenta la participación y el desarrollo colaborativo en proyectos de código abierto. Bitbucket, por otro lado, suele ofrecer repositorios privados gratuitos, pero limita la cantidad de colaboradores.

Editor del proyecto

- **Herramientas valoradas:** [VIM](#), [Visual Studio Code](#), [Sublime](#).
- **Herramienta elegida:** [Visual Studio Code](#).

Dibujos, diagramas y planos

- **Herramientas valoradas:** [Inkscape](#), [GIMP](#), [Paint](#), [Paint3D](#), [Photoshop](#), y [Draw.io](#).
- **Herramienta elegida:** [GIMP](#) y [Photoshop](#).

Procesador de textos L^AT_EX

- **Herramientas valoradas:** L^AT_EX, MS Word, Sublime, Overleaf.
- **Herramienta elegida:** L^AT_EX y Overleaf.

L^AT_EX [27] es un sistema de preparación de documentos que se utiliza ampliamente para la creación de documentos científicos, técnicos y académicos. A diferencia de los procesadores de texto tradicionales, LaTeX se basa en la creación de documentos mediante instrucciones de marcado en lugar de formatos visuales directos. Los usuarios escriben el contenido del documento junto con comandos LaTeX que definen la estructura, el formato y otros elementos.

Overleaf [36] es una plataforma en línea que permite la colaboración en tiempo real y la edición de documentos LaTeX. Algunas razones para considerar el uso de Overleaf junto con LaTeX son:

- **Colaboración en Tiempo Real:** Overleaf facilita la colaboración en documentos LaTeX entre múltiples autores en tiempo real, permitiendo que varios colaboradores trabajen simultáneamente en un proyecto.
- **Entorno en Línea:** No es necesario instalar LaTeX localmente en tu máquina. Overleaf proporciona un entorno en línea que elimina la necesidad de configurar y mantener un sistema LaTeX en tu computadora.
- **Plantillas y Recursos:** Overleaf ofrece una variedad de plantillas predefinidas para diferentes tipos de documentos, lo que facilita el inicio de nuevos proyectos. Además, proporciona acceso a una amplia variedad de recursos y tutoriales.
- **Control de Versiones Integrado:** Overleaf tiene un sistema de control de versiones integrado que permite realizar un seguimiento de los cambios en el documento y revertir a versiones anteriores si es necesario.
- **Exportación y Publicación:** Overleaf permite exportar documentos a diferentes formatos (PDF, HTML, etc.) y facilita la publicación directa en revistas académicas que admiten el formato LaTeX.

Calidad del Código

- **Herramientas valoradas:** [SonarCloud](#), [Codebeat](#), [Codacy](#) y [SonarQube](#).
- **Herramienta elegida:** [SonarCloud](#).

SonarCloud [40], elegido para optimizar la calidad del código en mi proyecto, realiza revisiones exhaustivas en la nube y ofrece valiosas sugerencias para mejorar. Su integración en tiempo real con GitHub asegura una evaluación continua y facilita la corrección de problemas de código de manera inmediata.

4.6. Entorno físico

Raspberry Pi Pico W

La Raspberry Pi Pico W [12] es un microcontrolador de bajo costo y alto rendimiento desarrollado por la Fundación Raspberry Pi. Se basa en el chip RP2040 diseñado por Raspberry Pi, que integra un procesador ARM Cortex-M0+ de doble núcleo y ofrece una serie de características diseñadas para proyectos de electrónica y programación embebida.

Tabla 4.1: Características de la Raspberry Pi Pico W.

Característica	Descripción
Microcontroller	RP2040 @ 133MHz, 264kB SRAM
Procesador	Dual Core ARM Cortex-M0+
Flash memory	2MB
Exposed GPIO	26
ADC	3 channels
I2C	2
SPI	2
UART	2
PWM	16 channels
Wireless interfaces	Wifi 802.11n, Bluetooth 5.2

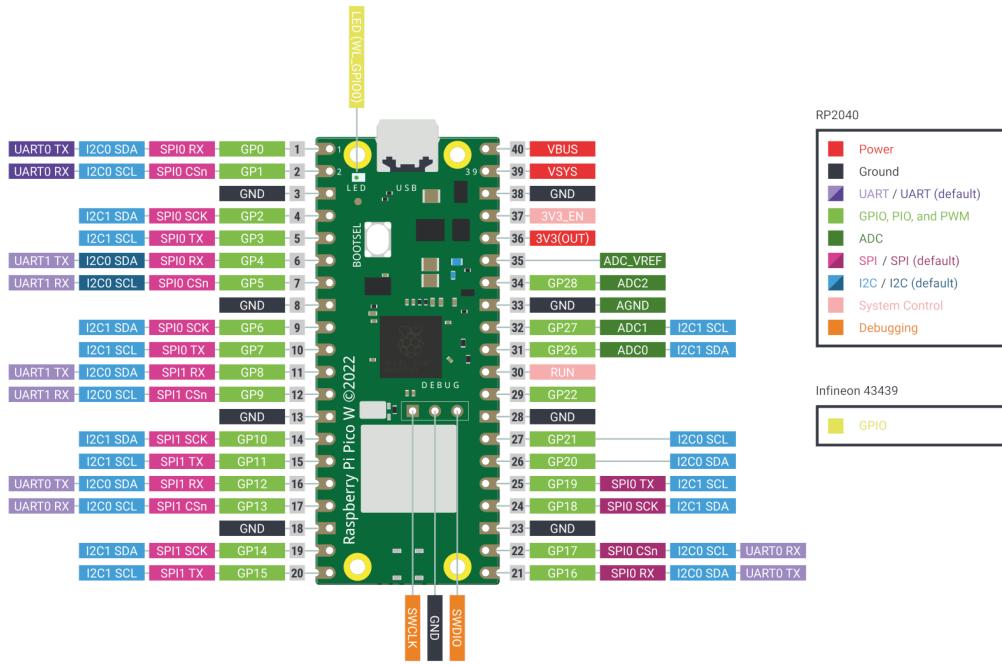


Figura 4.3: Raspberry Pi Pico W Pinout.

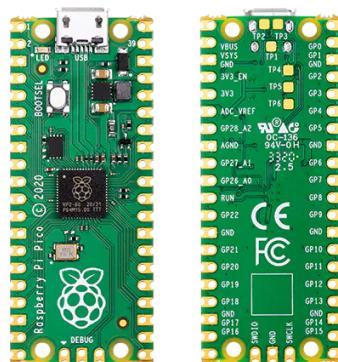


Figura 4.4: Vistas Frontal y posterior de Raspberry Pi Pico W.

Pantalla oled I2C de 128x64 píxeles y 1.3 pulgadas

La pantalla OLED [48] de 128x64 de 1.3 pulgadas con interfaz I2C es un dispositivo de visualización que utiliza la tecnología OLED (Diodo Orgánico de Emisión de Luz) y se comunica a través del protocolo I2C (Inter-Integrated Circuit).

Tabla 4.2: Características de la pantalla oled

Característica	Descripción
Voltaje de operación	3V - 5.5 V DC
Interfaz	I2C
Resolución	128×64
Monocromo	píxeles blancos (fondo negro)
Ángulo de visión	160°
Área visible (display)	23×11.5 mm
Consumo de energía ultra bajo	0.08 W (cuando están encendidos todos los píxeles)
item Temperatura de trabajo	-30°C - 70°C
item Dimensiones	27×27×4.1mm
item Peso	5 gramos

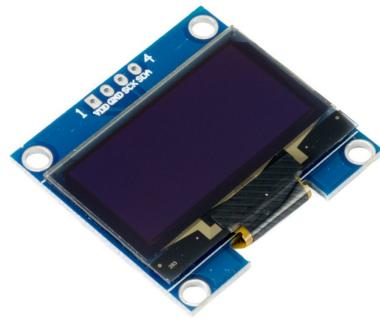


Figura 4.5: Pantalla oled.

Sensor de luz BH1750

El sensor BH1750 [17] es un sensor de intensidad de luz digital que mide la luminosidad ambiental en lux (lx). Este dispositivo es utilizado comúnmente en aplicaciones donde se requiere monitoreo de la luz, como en sistemas de iluminación automática, dispositivos de ahorro de energía y proyectos de IoT. Posee un conversor interno de 16-bit, por lo que entrega una salida digital en formato I2C

Tabla 4.3: Características del sensor de luz BH1750.

Característica	Descripción
Voltaje de Operación	3V – 5V
Interfaz digital	I2C
Respuesta espectral	similar a la del ojo humano
Rango de medición	1 lux - 65535 lux
Consumo de energía	bajo



Figura 4.6: Vista frontal del sensor BH1750.



Figura 4.7: Lado posterior del sensor BH1750.

Sensor de temperatura y humedad DHT22

El sensor DHT22 [41] (también conocido como AM2302) es un sensor de temperatura y humedad digital utilizado en aplicaciones donde es crucial monitorear y medir las condiciones ambientales.

Tabla 4.4: Características del sensor de temperatura y humedad DHT22.

Característica	Descripción
Voltaje de funcionamiento	3 V - 5.5 V
Forma de salida de señal	señal digital
Rango de medición de temperatura	-40 °C - 80 °C
Rango de medición de humedad	0 - 100 % HR
Resolución de temperatura	0.1 °C
Resolución de humedad	0.1 % HR
Precisión de medición de temperatura	±0.5 °C
Precisión de medición de humedad	±2 % HR
Tamaño	28.2 x 13.1 x 10 mm
Tiempo de sensado	2s
Modelo	AM2302

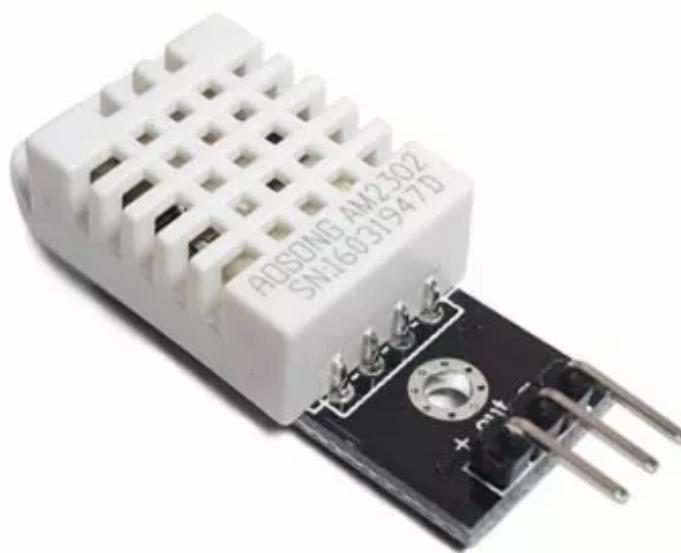


Figura 4.8: Sensor DHT22.

Sensor de humedad de suelo

El sensor de humedad de suelo Higrometro V1.2 [8] es un dispositivo diseñado para medir la humedad del suelo en entornos agrícolas, de jardinería u otros contextos donde el control de la humedad del suelo es esencial.

Tabla 4.5: Características del sensor de humedad de suelo.

Característica	Descripción
Voltaje de alimentación	3.3V - 5V DC
Corriente operación	5 mA
Voltaje de la señal de salida	0 a 5V (Analógico)
Modelo	capacitive soil moisture sensor v1.2
Vida útil	3 años mínimo
Conecotor	PH2.0-3P
Dimensiones	98×23 mm
Peso	15 gramos



Figura 4.9: Sensor de humedad de suelo Higrometro V1.2.

KY-016 FZ0455 Módulo led RGB de 3 colores

El módulo KY-016 [6] FZ0455 es un dispositivo electrónico que contiene un LED RGB (Light Emitting Diode - Diodo Emisor de Luz) capaz de emitir luz en tres colores primarios: rojo, verde y azul. Este módulo se utiliza comúnmente en proyectos electrónicos y de iluminación para agregar efectos de luz y color controlables.

Tabla 4.6: Características del módulo led RGB KY-016.

Característica	Descripción
Voltaje operativo	3.3 V – 5 V
Modo de accionamiento led	Accionamiento de cátodo común
Tamaño	3.5×0.8 cm
Voltaje de avance V_f [Red]	1.8 V
Voltaje de avance V_f [Green, Blue]	2.8 V
Corriente directa hacia adelante I_f	20 mA

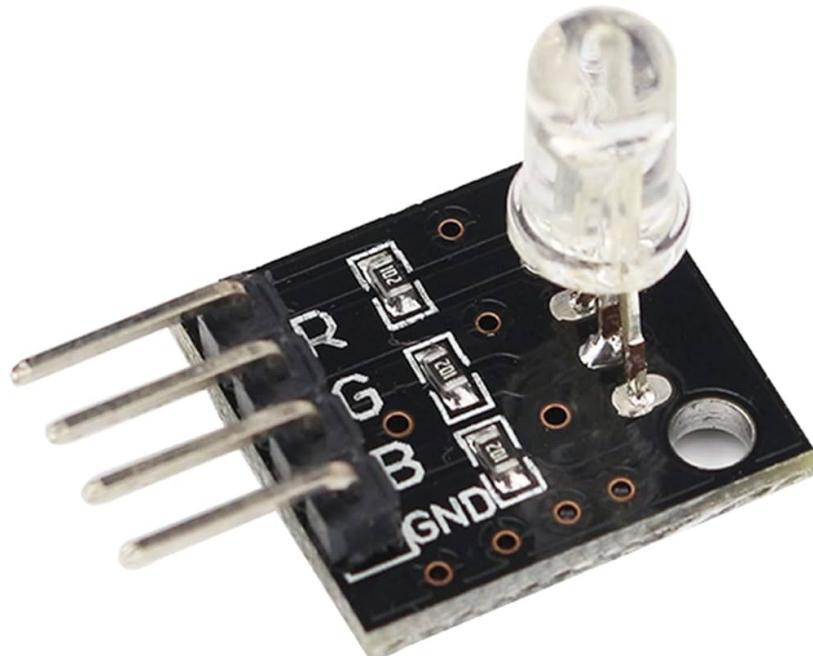


Figura 4.10: Módulo led RGB KY-016.

Enrutador WiFi USB 4G LTE

El Enrutador WiFi [2] USB 4G LTE es un dispositivo portátil que permite establecer una conexión a Internet de alta velocidad utilizando la red móvil 4G LTE. Diseñado para la movilidad y la conveniencia, este router WiFi ofrece una solución eficiente para aquellos que requieren acceso a la red en cualquier lugar donde haya cobertura de red móvil. Con una ranura para tarjeta SIM, el dispositivo facilita la inserción de una tarjeta SIM compatible, lo que lo convierte en una opción versátil para la conectividad a Internet en movimiento. Con una velocidad de transferencia de hasta 150 Mbps, este enrutador proporciona un rendimiento adecuado para actividades en línea como navegación web, transmisión de video y juegos en línea.

Tabla 4.7: Características del enrutador WiFi USB 4G LTE.

Característica	Descripción
Velocidad de Transferencia	Hasta 150 Mbps
Red Móvil Compatible	Tecnología 4G LTE
4G LTE FDD	B1/B3/B5
Wifi	Support IEEE802.11b/g/n Band 2.4G
Número máximo de usuarios	10
Ranura para Tarjeta SIM	Necesario para la conexión a la red móvil
Conectividad WiFi	Proporciona una red WiFi local
Compatibilidad	Computadoras, tabletas y teléfonos
Seguridad	Cifrado WPA/WPA2
Alimentación por batería	Es opcional y permite mayor portabilidad
Indicadores LED	Estado de la conexión e intensidad de la señal
Material	ABS
Dimensiones	4.33×2.76×0.79 pulgadas



Figura 4.11: Enrutador WiFi USB 4G LTE.

Batería externa 20000 mAh

Es un Power Bank [1] portátil diseñado para cargar dispositivos móviles de forma rápida y eficiente. Con dos salidas USB-A y una entrada Type-C, proporciona flexibilidad para cargar múltiples dispositivos simultáneamente y recargarse fácilmente. Fabricada con aleación de aluminio, esta mini Powerbank, en elegante color gris, ofrece durabilidad y un diseño compacto que facilita su transporte. Con una capacidad de 20000mAh, es ideal para mantener tus dispositivos cargados mientras estás en movimiento, ya sea en viajes, actividades al aire libre o situaciones de emergencia.

Tabla 4.8: Características de la batería externa 20000 mAh.

Característica	Descripción
Tipo de conector entrada	usb tipo C (5V/2A)
Tipo de conector salida	usb tipo A2 (5V/2A)
Marca	YWTESCH
Color	Gris
Tensión	5 voltios
Amperaje	2 A
Capacidad máxima	20000 mAh
Número de puertos	2
tamaño	108×69×26 mm



Figura 4.12: Batería externa 20000 mAh.

Panel solar IP65 de 8W y 5V con cargador USB tipo C

Este Panel Solar [28] es una solución eficiente para la carga de dispositivos móviles utilizando energía solar. Es una opción versátil para cargar dispositivos compatibles, como teléfonos, tabletas y otros dispositivos alimentados por USB. Compacto y fácil de transportar, es ideal para actividades al aire libre, viajes y situaciones donde la carga convencional no está disponible.

Tabla 4.9: Características del panel solar.

Característica	Descripción
USB	Tipo C con cable de 3 m
Capacidad	8W y 5V
Material del panel	Silicio monocristalino
Vida útil	20 años
Accesorios	Soporte ajustable 360° con tornillos y tojinos
Dimensiones	23×18.5×1 cm
Tipo de protección	IP65 (resistente al polvo y agua)



Figura 4.13: Panel solar y soporte ajustable.

5. Aspectos relevantes del desarrollo del proyecto

En esta sección, se destacarán los elementos más significativos del desarrollo, proporcionando una perspectiva detallada de las decisiones adoptadas para alcanzar los objetivos del proyecto. Se resumirá la experiencia práctica, detallando cómo se abordaron los desafíos específicos, y se evaluará la importancia de estas soluciones en el contexto general del alcance del proyecto.

5.1. Motivación del proyecto

La motivación subyacente a este proyecto surge de la creciente importancia y demanda en el ámbito de la monitorización de invernaderos, especialmente en el contexto del cultivo de cannabis medicinal. La necesidad de implementar soluciones tecnológicas eficientes para garantizar condiciones óptimas de cultivo y maximizar la producción ha impulsado la elección de este tema.

El objetivo principal radica en diseñar un sistema económico y eficaz que permita a los cultivadores de cannabis medicinal supervisar las condiciones ambientales de sus invernaderos de manera remota. Este proyecto busca proporcionar una herramienta valiosa para mejorar la eficiencia, la calidad y la consistencia en la producción de cannabis medicinal, al tiempo que se abordan los desafíos específicos asociados con la monitorización de variables críticas como temperatura, humedad y luz.

5.2. Formación necesaria

El proyecto ha requerido una formación interdisciplinaria que abarca diversas áreas. En primer lugar, la comprensión profunda de los principios de la programación y el desarrollo de software ha sido esencial. La elección de la Raspberry Pi Pico W como plataforma y la programación en MicroPython para controlar los dispositivos hardware involucra un conocimiento sólido en programación embebida y desarrollo en entornos limitados.

Además, la utilización de sensores especializados, como el DHT22 [41], el BH1750 [17] y el sensor de humedad del suelo [8], ha demandado una comprensión detallada de los principios de operación de estos dispositivos y cómo integrar sus lecturas en el sistema general. Esto implica una formación técnica en electrónica y sensores.

La implementación de conceptos relacionados con el Internet de las cosas (IoT) y la comunicación inalámbrica mediante la Raspberry Pi Pico W [11] también ha requerido conocimientos específicos en redes y protocolos de comunicación.

La integración de MQTT [26] en el código MicroPython para la Raspberry Pi Pico W ha requerido conocimientos específicos sobre este protocolo de mensajería ligero diseñado para la eficiente comunicación entre dispositivos en redes IoT.

En resumen, la formación necesaria abarca habilidades en programación embebida, electrónica, redes, IoT y desarrollo de software. La combinación de estas competencias ha sido esencial para la ejecución exitosa del proyecto, demostrando la importancia de una formación integral y multidisciplinaria en el ámbito de la ingeniería y la tecnología.

5.3. Metodología

Se ha optado por la metodología del **Modelo de Prototipos** [18] debido a su capacidad para proporcionar retroalimentación temprana, clarificar requisitos, adaptarse a cambios, validar prácticamente funcionalidades, reducir riesgos, comprender la interfaz de usuario y facilitar la mejora continua. Este enfoque iterativo permite identificar y corregir problemas desde las primeras etapas, garantizando la eficiencia y el éxito en el desarrollo del sistema IoT de monitorización de invernaderos de cannabis medicinal.

Identificación de Requisitos Iniciales:

Hardware:

- Raspberry Pi Pico W [11] como la plataforma central.
- Sensores DHT22 [41] para medir temperatura y humedad ambiente.
- Sensor BH1750 [17] para evaluar la intensidad de luz.
- Sensor de humedad de suelo [8] para monitorizar las condiciones del suelo.
- Pantalla OLED [48] para visualización de datos.
- LEDs RGB [6] para indicar condiciones críticas.

Software:

- Desarrollo de un sistema en MicroPython para la Raspberry Pi Pico W 5.4.
- Implementación de un servidor LAMP para almacenar y gestionar datos.
- Integración de la API de Telegram [45] para recibir alertas.
- Aplicación de escritorio para Windows 5.4, desarrollada en Visual Studio 2022 en lenguaje C#. Incluye su instalador.
- Dashboard web 5.4, realizado con Node.js, accesible desde internet.

Funcionalidades clave:

- Captura de datos ambientales como temperatura, humedad, luz y humedad del suelo.
- Representación gráfica de los datos en una pantalla OLED.
- Envío de alertas a través de Telegram en caso de valores fuera de los umbrales ideales.
- Interacción bidireccional para activar mecanismos de respuesta a condiciones ambientales adversas.

Desarrollo del Primer Prototipo:

En esta etapa, se enfocó en la integración de los componentes hardware definidos, como la Raspberry Pi Pico W [11] y los sensores de humedad, temperatura [41], luminosidad [17] y humedad del suelo [8]. Se estableció la conexión con el servidor LAMP utilizando MQTT [26] para la transmisión de datos en tiempo real.

Durante la fase de desarrollo, se configuraron y probaron los mecanismos de alerta a través del bot de Telegram y la activación de los LEDs RGB para indicar condiciones críticas. También se diseñó e implementó la lógica de umbrales para verificar si los valores ambientales están dentro de los límites ideales.

Evaluación del Primer Prototipo:

El objetivo principal fue validar la efectividad de las funcionalidades implementadas y su capacidad para cumplir con los requisitos planteados en la fase inicial del proyecto.

- **Pruebas de Mecanismos de Alerta y Retroalimentación Visual:** Se realizaron pruebas del sistema de alertas a través del bot de Telegram. Se verificó la capacidad del sistema para enviar notificaciones en tiempo real sobre condiciones ambientales críticas, como temperaturas extremas o niveles de humedad fuera de los umbrales establecidos.

Además, se evaluó la respuesta de los LEDs RGB conectados, los cuales indican visualmente las condiciones del entorno. Cada color del LED se asoció con un estado específico, permitiendo una retroalimentación instantánea sobre el estado actual del invernadero.

- **Verificación de la Lógica de Umbrales:** La lógica de umbrales, diseñada para verificar si los valores ambientales se mantienen dentro de los límites ideales, está en fase de pruebas en un entorno no controlado. En este contexto, se observa que los valores experimentan salidas fuera de los umbrales establecidos. Sin embargo, se ha verificado con éxito que la lógica implementada para activar las alertas funciona de manera efectiva cuando los valores se desvían de los límites ideales predefinidos.

Retroalimentación y Mejoras:

Tras la evaluación del primer prototipo, se recopiló retroalimentación valiosa para orientar mejoras futuras. Se identificaron áreas de oportunidad, como la necesidad de ajustar los umbrales para adaptarse a condiciones específicas del entorno y mejorar la precisión de las alertas. La retroalimentación del usuario sobre la interfaz y la usabilidad también se tuvo en cuenta para realizar ajustes en el diseño del dashboard web y la aplicación de escritorio. Estas observaciones se utilizarán como guía para la siguiente iteración del prototipo, buscando una mayor eficiencia y adecuación a los requisitos específicos del sistema.

Desarrollo de Subsiguentes Prototipos:

Con base en la retroalimentación obtenida del primer prototipo, se inició el desarrollo de subsiguientes prototipos con el objetivo de implementar mejoras significativas. Se priorizó la optimización de los umbrales de alerta, considerando las condiciones específicas del entorno de monitoreo. También se abordaron ajustes en la interfaz del dashboard web y la aplicación de escritorio, respondiendo a las sugerencias de los usuarios para mejorar la usabilidad y la experiencia general. Cada nuevo prototipo se sometió a pruebas exhaustivas para validar las mejoras implementadas y garantizar un funcionamiento más eficiente del sistema en su conjunto. Este proceso iterativo permitió refinamientos continuos, enfocados en lograr un sistema de monitoreo más preciso y adaptado a las necesidades específicas del usuario.

Validación Continua:

Se llevaron a cabo pruebas exhaustivas para verificar el funcionamiento del sistema, evaluando diversos escenarios y condiciones ambientales para confirmar la precisión de los sensores y la respuesta del sistema. Se realizaron pruebas de estrés para evaluar la capacidad de respuesta en situaciones extremas.

Durante un período prolongado, se recopilaron datos para analizar el rendimiento a lo largo del tiempo, comparándolos con los umbrales ideales establecidos. Algunos valores se salieron del rango ideal debido a las pruebas en un entorno no controlado, pero se verificó que el sistema funcionó correctamente y generó alertas de manera adecuada al encontrarse en un ambiente controlado.

La retroalimentación continua de las pruebas y la validación de los resultados se utilizaron para ajustar el sistema, mejorando su robustez y confiabilidad. Este ciclo de validación y ajuste continuo contribuyó a optimizar el sistema para lograr un monitoreo preciso y eficiente de las condiciones del invernadero.

Integración de Componentes:

Durante esta etapa, se procedió a la integración de los diferentes componentes del sistema. Se conectaron y configuraron la Raspberry Pi Pico W [11] como plataforma central, junto con los sensores DHT22, BH1750 [17] y el sensor de humedad del suelo [8]. Se aseguró la comunicación efectiva entre estos elementos, permitiendo la captura precisa de datos ambientales.

El desarrollo de un sistema en MicroPython [7] para la Raspberry Pi Pico W se integró con éxito con el servidor LAMP, facilitando la gestión y almacenamiento de datos. La implementación de la API de Telegram [45] se conectó con el sistema para recibir alertas en tiempo real.

La aplicación de escritorio para Windows 5.4, desarrollada en Visual Studio 2022 con lenguaje C# [30], se integró sin problemas, permitiendo la visualización de datos en tiempo real y la interacción bidireccional con el sistema.

El dashboard web 5.4, creado con Node.js, se integró para ofrecer una interfaz accesible a través de internet, proporcionando una visualización completa y en tiempo real de los datos capturados. Esta etapa de integración aseguró la cohesión y el funcionamiento conjunto de todos los componentes del sistema IoT para el monitoreo de invernaderos.

Ajustes Finales:

Estos ajustes se centraron en mejorar la precisión de los sensores, la eficacia de las alertas y la respuesta de los mecanismos de control. Se llevaron a cabo modificaciones en el código del sistema para perfeccionar la lógica

de umbrales y asegurar que los valores se mantuvieran dentro de los límites ideales.

Además, se realizaron ajustes en la configuración de la pantalla OLED y los LEDs RGB para una visualización más clara y distintiva de las condiciones del invernadero. La interfaz de usuario se mejoró para facilitar la comprensión de los datos mostrados en tiempo real. También se realizaron pruebas adicionales para verificar la estabilidad del sistema en condiciones diversas.

Este proceso de ajustes finales contribuyó a la optimización global del sistema, asegurando un monitoreo preciso y una respuesta efectiva a las variaciones ambientales en el invernadero. La retroalimentación continua y la iteración fueron fundamentales en esta fase para lograr un prototipo final robusto y funcional.

Documentación:

Se elaboró una documentación completa que abarca aspectos técnicos, funcionales y de implementación del sistema desarrollado. La documentación detallada incluye manuales de usuario para las distintas interfaces, descripciones técnicas de hardware y software, así como instrucciones para la instalación y configuración del sistema.

Además, se generaron documentos específicos para el mantenimiento del sistema, facilitando futuras actualizaciones y mejoras.

Esta documentación sirve como recurso integral para cualquier persona que interactúe con el sistema, desde usuarios finales hasta desarrolladores o técnicos de mantenimiento. Su elaboración garantiza la comprensión y el uso efectivo del sistema, contribuyendo a su sostenibilidad a lo largo del tiempo.

Implementación Final:

La implementación final del sistema se llevó a cabo tras la validación y ajustes continuos a lo largo de las fases de desarrollo y prototipado. En esta etapa, se consolidaron todas las funcionalidades y características planificadas, integrando los componentes hardware y software de manera coherente.

Se realizó una verificación exhaustiva de la interacción entre los sensores, la Raspberry Pi Pico W, la pantalla OLED, y los LEDs RGB. Se mejoró la respuesta del sistema ante condiciones ambientales diversas y se afinaron los algoritmos de control para garantizar un monitoreo preciso y eficiente del invernadero.

Además, se implementaron las últimas medidas de seguridad y se realizaron pruebas exhaustivas para garantizar la estabilidad del sistema en distintos escenarios. Se documentaron detalladamente los pasos necesarios para replicar la implementación final, asegurando una fácil reproducción y mantenimiento.

La implementación final refleja la evolución del sistema desde su concepción hasta la etapa de producción, consolidando todas las lecciones aprendidas durante el desarrollo y prototipado.

5.4. Desarrollo del proyecto

El propósito inicial radicaba en la recopilación de datos ambientales, tales como humedad, temperatura, intensidad de luz y humedad del suelo, con el fin de almacenarlos y presentarlos de manera gráfica.

Con este propósito, se optó por la Raspberry Pi Pico W debido a su eficiencia energética y coste accesible. Se conectaron al sistema un sensor de humedad y temperatura DHT22 [41], un sensor de intensidad de luz BH1750 [17], y un sensor de humedad de suelo [8]. Además, se incorporaron LEDs RGB [6] para señalizar alertas. A esta parte del proyecto se ha llamado **Hardware 5.4**.

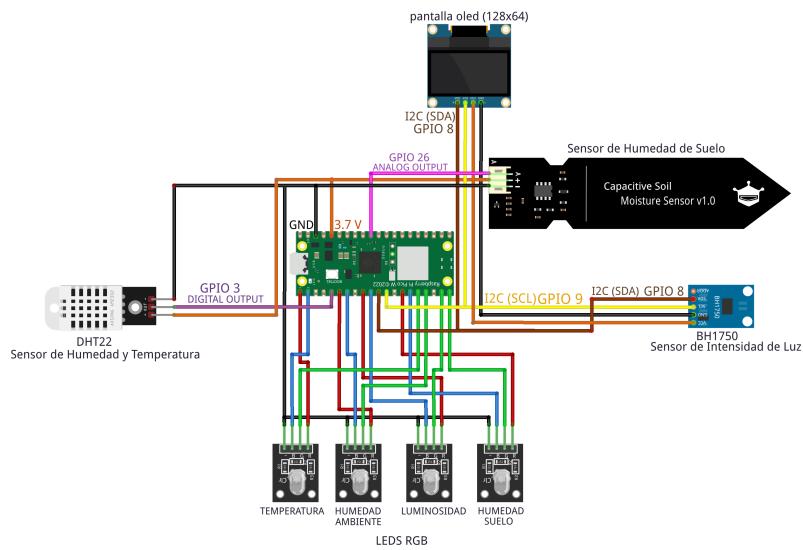


Figura 5.1: Raspberry Pi Pico W conectada a sensores y una pantalla oled.

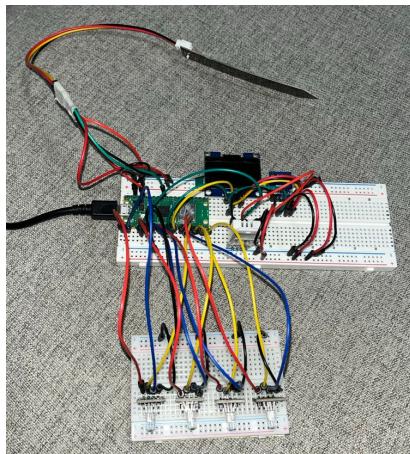


Figura 5.2: Raspberry Pi Pico W y sensores conectados mediante un protoboard.

Una vez definida la idea del TFG y los componentes hardware, surgía la interrogante: "¿Dónde enviar los datos? ¿Cómo gestionarlos?". La primera medida fue mostrar los datos en una pantalla OLED [48] conectada a la protoboard.

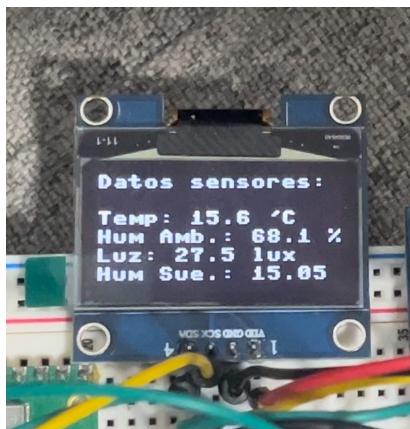


Figura 5.3: Pantalla oled conectada a la Raspberry Pi Pico W.

Sin embargo, al estar fuera de la red, surgieron nuevas incógnitas, ya que inicialmente se trató de un sistema unidireccional en el cual el sensor recolecta los datos y los muestra en la pantalla OLED.

Más tarde, se implementó la funcionalidad de enviar alertas a través de Telegram en caso de que los valores recopilados excedieran los umbrales ideales.

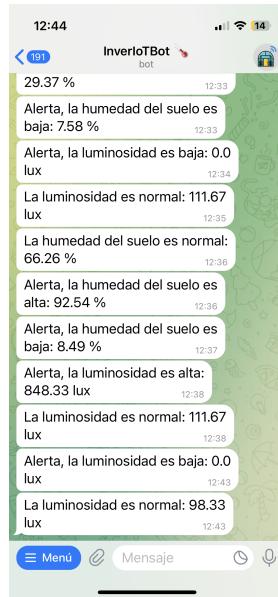
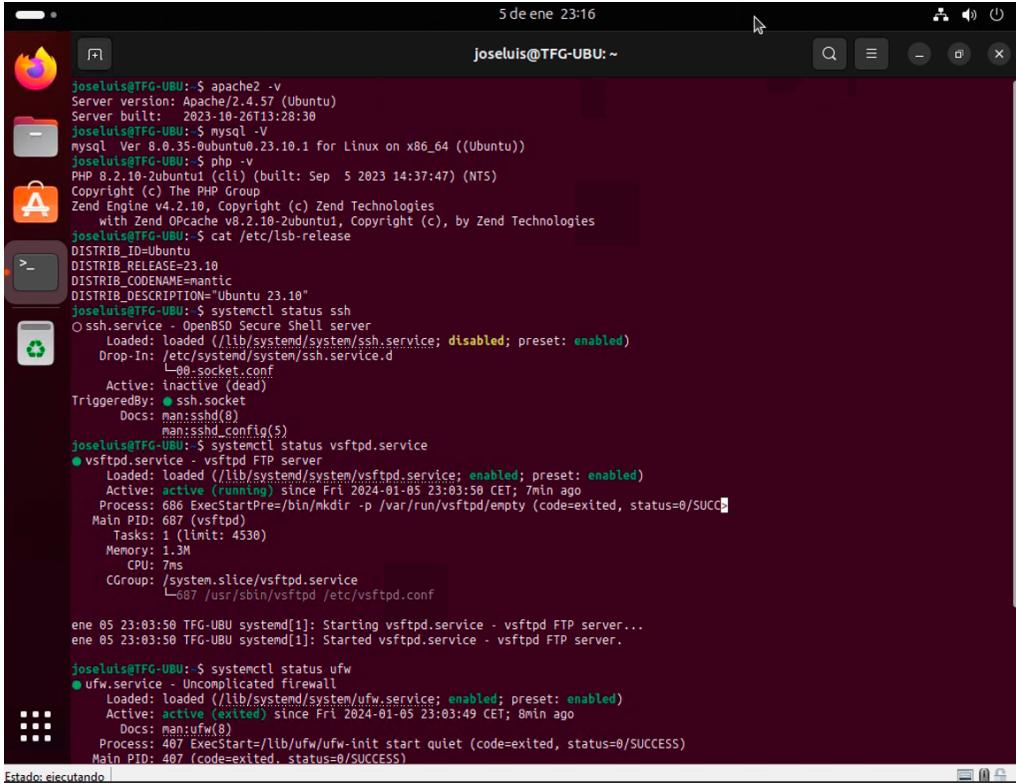


Figura 5.4: Alertas del bot de Telegram.

Posteriormente, opté por establecer un servidor LAMP para facilitar el envío y manejo de datos. La implementación de este servidor se llevó a cabo en el sistema operativo Ubuntu [47], el cual fue virtualizado utilizando Hyper-V [31]. En esta instancia, se emplea un servidor físico HP ProLiant [19] para la virtualización de los servicios Apache [3], MySQL [33], PHP [38] y SSH [42].

Tabla 5.1: Características del servidor HP ProLiant.

Característica	Descripción
Sistema operativo	Windows Server 2019 Standard
Fabricante	Hewlett Packard Enterprise
Modelo	Hewlett Packard Enterprise x64 Class PC
Procesador	Intel(R) Xeon(R) Silver 4210R CPU @ 2.4GHz 2.39GHz
Memoria instalada (RAM)	128 GB (128 GB utilizable)
Tipo de sistema	Sistema operativo de 64 bits, procesador x64



```

5 de ene 23:16
joseluis@TFG-UBU: ~

joseluis@TFG-UBU: $ apache2 -v
Server version: Apache/2.4.57 (Ubuntu)
Server built: 2023-10-26T13:28:30
joseluis@TFG-UBU: $ mysql -V
mysql Ver 8.0.35-0ubuntu0.23.10.1 for Linux on x86_64 ((Ubuntu))
joseluis@TFG-UBU: $ php -v
PHP 8.2.10-2ubuntu1 (cli) (built: Sep 5 2023 14:37:47) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.2.10, Copyright (c) Zend Technologies
with Zend OPCache v8.2.10-2ubuntu1, Copyright (c), by Zend Technologies
joseluis@TFG-UBU: $ cat /etc/lsb-release
DISTRIB_ID=ubuntu
DISTRIB_RELEASE=23.10
DISTRIB_CODENAME=mantic
DISTRIB_DESCRIPTION="Ubuntu 23.10"
joseluis@TFG-UBU: $ systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; disabled; preset: enabled)
  Drop-In: /etc/systemd/system/ssh.service.d
            └─00-socket.conf
    Active: inactive (dead)
      TriggeredBy: ● ssh.socket
        Docs: man:sshd(8)
               man:sshd_config(5)
joseluis@TFG-UBU: $ systemctl status vsftpd.service
● vsftpd.service - vsftpd FTP server
  Loaded: loaded (/lib/systemd/system/vsftpd.service; enabled; preset: enabled)
  Active: active (running) since Fri 2024-01-05 23:03:50 CET; 7min ago
    Process: 686 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exited, status=0/SUCCESS)
   Main PID: 687 (vsftpd)
      Tasks: 1 (limit: 4530)
     Memory: 7ms
        CPU: 7ms
       CGroup: /system.slice/vsftpd.service
                 └─687 /usr/sbin/vsftpd /etc/vsftpd.conf

ene 05 23:03:50 TFG-UBU systemd[1]: Starting vsftpd.service - vsftpd FTP server...
ene 05 23:03:50 TFG-UBU systemd[1]: Started vsftpd.service - vsftpd FTP server.

joseluis@TFG-UBU: $ systemctl status ufw
● ufw.service - Uncomplicated firewall
  Loaded: loaded (/lib/systemd/system/ufw.service; enabled; preset: enabled)
  Active: active (exited) since Fri 2024-01-05 23:03:49 CET; 8min ago
    Docs: man:ufw(8)
  Process: 407 ExecStart=/lib/ufw/ufw-init start quiet (code=exited, status=0/SUCCESS)
   Main PID: 407 (code=exited, status=0/SUCCESS)

Estado: ejecutando

```

Figura 5.5: Servicios instalados y operativos en el servidor LAMP.

Basándome en esta información y en mi experiencia previa trabajando con **Smart Cities**, tomo la decisión de enviar la información recopilada por los sensores al servidor mediante MQTT [26].

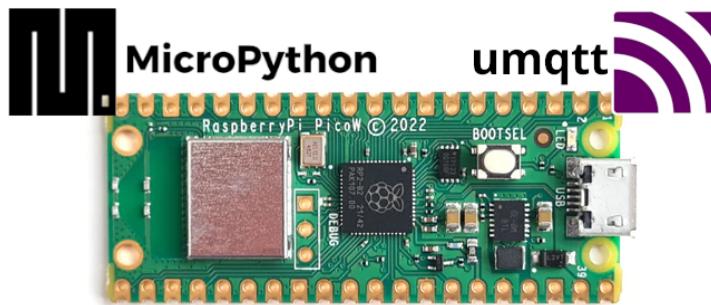


Figura 5.6: Se ha implementado la librería umqtt en la Raspberry Pi Pico W.

MQTT permite la visualización en tiempo real de los datos en una página web alojada en el servidor LAMP cada vez que la placa junto con el sensor envíe datos. Además, posibilita el almacenamiento de los datos en la base de datos MySQL [33], permitiéndome mantener un historial y realizar consultas. A esta parte del proyecto se ha llamado **NodeMqtt 5.4**.

	<input type="checkbox"/>										
	←	→		id	1	fecha	hora	temperatura	humedad	intensidad_luz	humedad_suelo
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	4118	2024-02-03 22:45:13	20.00
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	4117	2024-02-03 22:45:10	20.00
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	4116	2024-02-03 22:45:06	20.00
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	4115	2024-02-03 22:45:03	19.90
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	4114	2024-02-03 22:45:00	19.90
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	4113	2024-02-03 22:44:57	19.90
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete			-0.97
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete			-0.97
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete			-0.97

Figura 5.7: Data almacenada en la base de datos.

Listo, he leído e incluso almacenado los datos que se envían. Además, con la implementación de un bot de Telegram [44], ahora puedo consultar en tiempo real la información proveniente de los sensores.

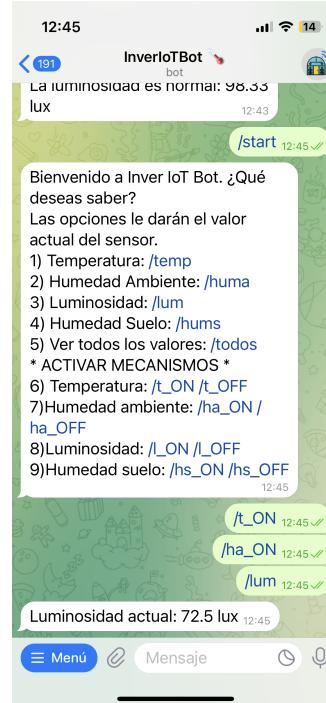


Figura 5.8: Consulta a la base de datos mediante comandos usando Telegram.

También optamos por desarrollar una aplicación de escritorio para Windows en C# [30] al que se ha llamado **InverIoT 5.4**, que permite la visualización en tiempo real de los datos. ¡Problema resuelto! Ahora, puedo recibir y monitorear la temperatura, luminosidad y otros valores del invernadero.

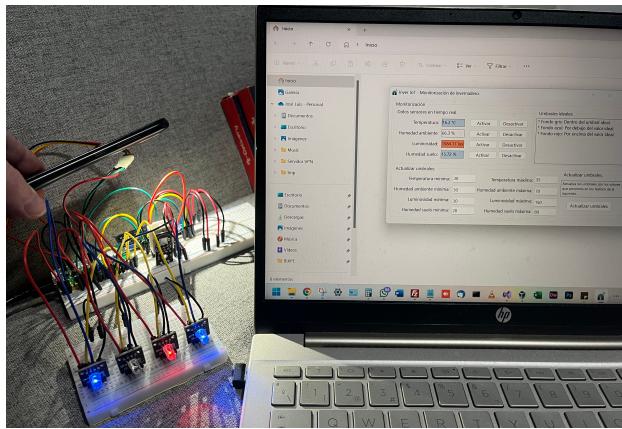


Figura 5.9: El led rojo y el color naranja en la aplicación InverIoT indican que se ha sobrepasado el umbral de luz.

Sin embargo, al caer la noche, la luminosidad y la temperatura disminuyen, planteando un nuevo desafío. En este caso, necesitaría establecer una comunicación bidireccional con la placa Raspberry Pi Pico W para, por ejemplo, encender los focos y activar los calentadores.

Este cambio a una comunicación bidireccional se logró suscribiendo la Raspberry Pi Pico W a un topic MQTT. Ahora, puedo enviar órdenes para procesarlas según sea necesario.

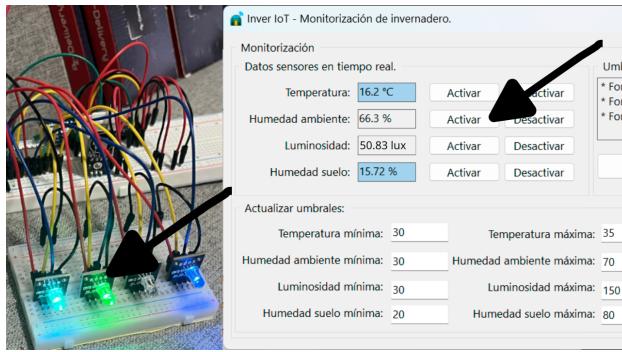


Figura 5.10: Al hacer clic en activar se enciende el led de color verde representando que estamos activando un mecanismo.

Sin embargo, la activación de mecanismos físicos como encender focos, activar calefacción, regar el suelo o humidificar el ambiente aún no se ha implementado, ya que corresponde a otras especialidades como telecomunicación industrial o mecánica.

El cannabis medicinal requiere condiciones ambientales constantes y dentro de un umbral específico para un crecimiento óptimo y delicado.

Definimos umbrales (valor mínimo/máximo) para mantener los valores de los sensores dentro de límites aceptables. En caso de detectar un valor anómalo fuera de estos umbrales, se genera una alerta. Para esta función, empleamos un bot de Telegram que envía mensajes de alerta, y se configura un LED RGB para cada medición (temperatura, humedad, luminosidad, humedad del suelo). Si la temperatura es baja, por ejemplo, el LED se ilumina en azul y envía una alerta por Telegram; si la temperatura es alta, el LED se enciende en rojo. Un LED apagado indica valores normales dentro del umbral.

Tabla 5.2: Umbrales ideales para un invernadero de cannabis medicinal.

Característica	Mínimo	Máximo
TEMPERATURA	30	35
HUMEDAD	30	70
LUMINOSIDAD	30	150
HUMEDAD DEL SUELO	20	80

La intervención manual también es posible, como activar ventiladores al observar un LED azul. Además, estos procesos pueden automatizarse. Por ejemplo, si la luminosidad cae por debajo del umbral, los focos se encienden hasta alcanzar el valor óptimo, por ahora eso no está implementado en este proyecto.

La información se visualiza en tiempo real en una página web desarrollada con Node.js, incluyendo un histórico de datos almacenados en la base de datos.

Hardware

- La Raspberry Pi Pico W estará situada en el invernadero para la recopilación de datos.
- Los LEDs RGB, instalados en la oficina del cliente, indicarán visualmente si los valores superan los umbrales establecidos.

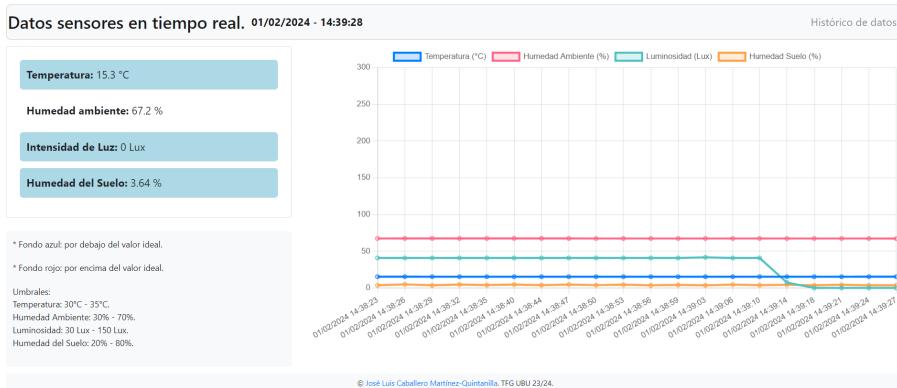


Figura 5.11: Página Web InverIoT desarrollada con Node.js.

- La pantalla OLED estará colocada en la puerta del invernadero para mostrar los valores actualizados.

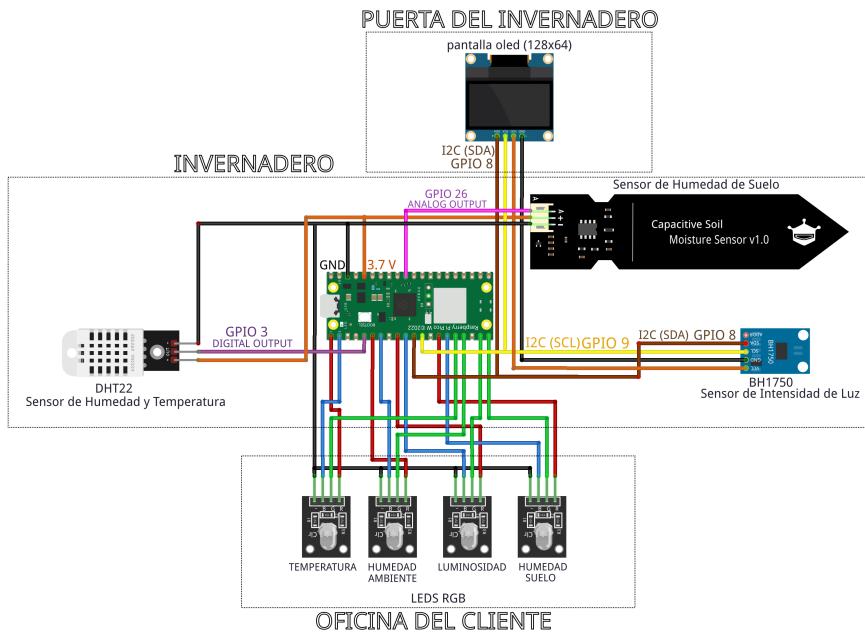


Figura 5.12: Conexiones.

InverIoT

Se desarrolla una aplicación de escritorio para Windows llamada InverIoT, que permite al usuario monitorizar en tiempo real los datos provenientes de los sensores conectados a la placa.

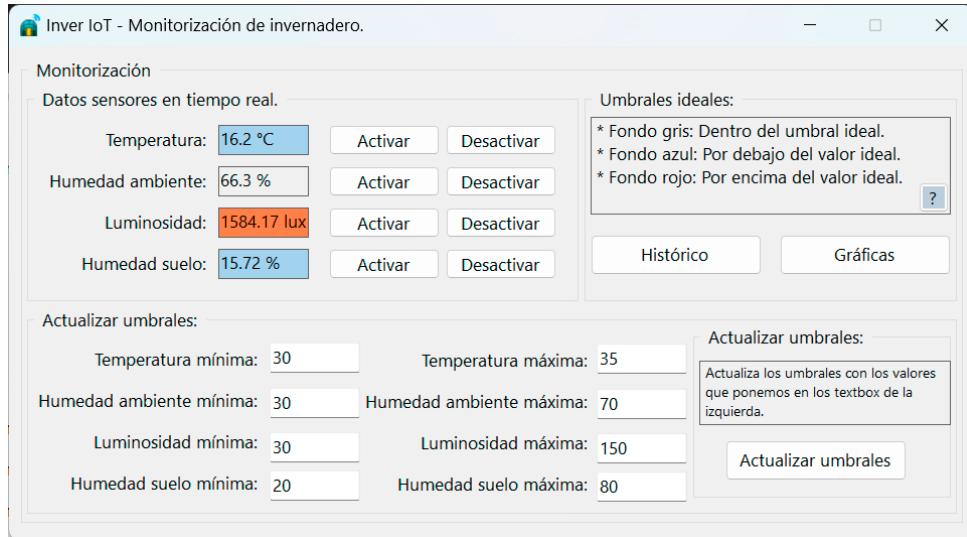


Figura 5.13: Aplicación de escritorio **InverIoT**.

Utilizando el protocolo MQTT [26], la aplicación suscribe y recibe los datos publicados en el mismo tema.

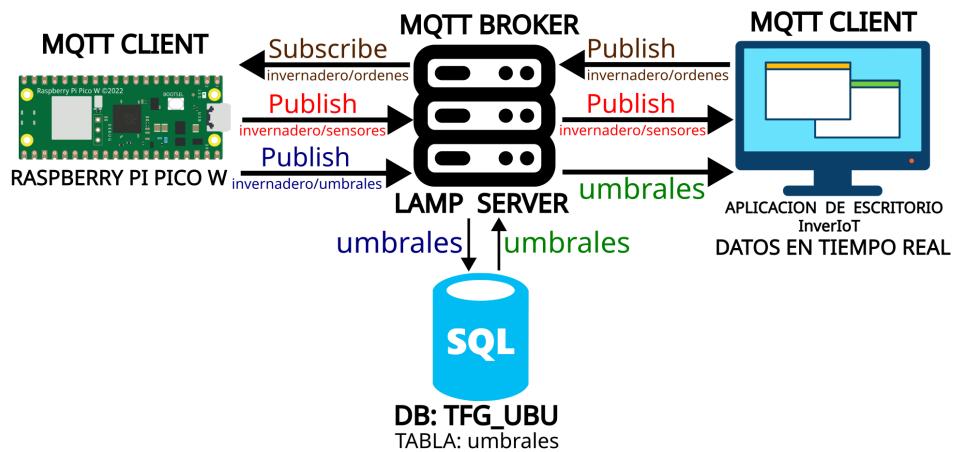


Figura 5.14: Enviando órdenes mediante MQTT la aplicación recibe datos en tiempo real.

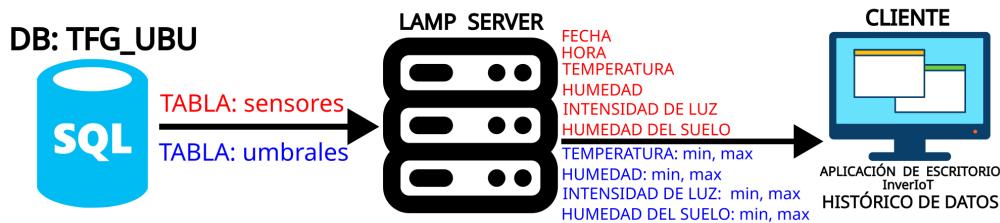


Figura 5.15: Para mostrar el historial de datos la aplicación extrae información de una base de datos almacenada en un servidor LAMP.

Estos datos son formateados y presentados en textboxes correspondientes, con unidades de medida agregadas.

Se incorporan botones para activar o desactivar mecanismos, como un LED verde, que corrigen valores fuera de los umbrales ideales indicados en la parte derecha de la interfaz.

Datos sensores en tiempo real.			
Temperatura:	16.2 °C	Activar	Desactivar
Humedad ambiente:	66.3 %	Activar	Desactivar
Luminosidad:	1584.17 lux	Activar	Desactivar
Humedad suelo:	15.72 %	Activar	Desactivar

Figura 5.16: Botones para activar o desactivar mecanismos.

Además, se añaden 8 textboxes en la parte inferior para ajustar manualmente los valores mínimos y máximos de cada parámetro.

Los umbrales se actualizan desde la base de datos del servidor LAMP, proporcionando indicadores visuales de color azul para valores bajos, gris para valores normales y rojo para valores altos.

Se desarrolló la aplicación de escritorio utilizando Windows Forms y programada en lenguaje C# [30].

La base de datos es alimentada por lo que en ese proyecto llamamos **No-deMqtt 5.4**, y la aplicación de escritorio **InverIoT** extrae esa información para mostrarla como un historial de datos.

La base de datos contiene dos tablas llamadas **sensores** y **umbrales**.

	Id	Fecha	Hora	Temperatura	Humedad Ambiente	Luminosidad	Humedad Suelo
▶	2592	31/01/2024	12:26:28	15,60	68,70	20,83	3,64
	2593	31/01/2024	12:26:32	15,50	68,50	20,83	3,82
	2594	31/01/2024	12:26:36	15,50	68,40	20,83	5,22
	2595	31/01/2024	12:26:40	15,50	68,40	20,83	4,79
	2596	31/01/2024	12:26:44	15,50	68,30	21,67	4,85
	2597	31/01/2024	12:26:48	15,50	68,30	21,67	4,13
	2598	31/01/2024	12:26:52	15,50	68,30	20,83	5,22
	2599	31/01/2024	12:26:57	15,50	68,30	21,67	5,04
	2600	31/01/2024	12:27:01	15,50	68,30	21,67	4,49
	2601	31/01/2024	12:27:05	15,50	68,30	21,67	5,16
	2602	31/01/2024	12:27:09	15,50	68,30	21,67	5,46
	2603	31/01/2024	12:27:14	15,50	68,30	21,67	4,43
	2604	31/01/2024	12:27:18	15,50	68,30	21,67	5,16
	2605	31/01/2024	12:27:22	15,50	68,30	21,67	4,37
	2606	31/01/2024	12:27:26	15,50	68,30	21,67	4,13
	2607	31/01/2024	12:27:31	15,50	68,30	23,33	4,06
	2608	31/01/2024	12:27:35	15,50	68,30	27,50	4,73
	2609	31/01/2024	12:27:40	15,50	68,20	33,33	4,43
	2610	31/01/2024	12:27:45	15,50	68,20	33,33	4,55
	2611	31/01/2024	12:27:49	15,50	68,20	32,50	5,22
	2612	31/01/2024	12:27:54	15,50	68,20	33,33	4,55
	2613	31/01/2024	12:27:58	15,50	68,20	33,33	4,43

Figura 5.17: Historial de los datos mostrara en la aplicación.



Figura 5.18: Gráfica mostrando los datos en un intervalo de fechas.

Dashboard

Se desarrolla un dashboard que permite al usuario visualizar en tiempo real los datos capturados por los sensores, con una interfaz similar a la aplicación de escritorio.

Los valores que exceden los umbrales establecidos se destacan mediante cambios de color.



Figura 5.19: Intensidad de luz superando los umbrales.

La plataforma incluye una gráfica en tiempo real y la capacidad de acceder a un historial con filtro de fecha.



Figura 5.20: Vista del historial, al cargar, muestra los datos del día actual.

Los umbrales utilizados se extraen de la tabla **TFG_UBU** en la base de datos MySQL [33].

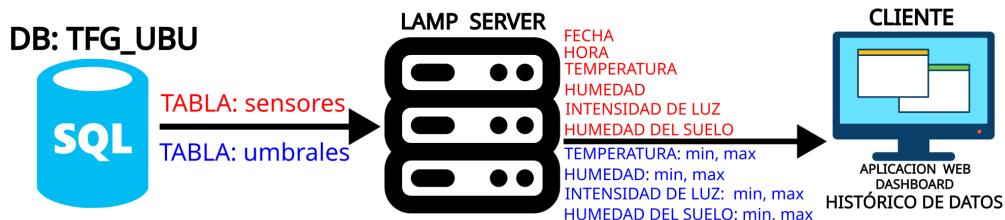


Figura 5.21: Extrae los datos de la base de datos para mostrar el historial.

La Raspberry Pi Pico W utiliza una librería llamada **umqtt** para poder trabajar con MQTT, así es como envía data de los sensores y de los umbrales establecidos.

Los umbrales establecidos son almacenados en una base de datos que se encuentra en el servidor LAMP.

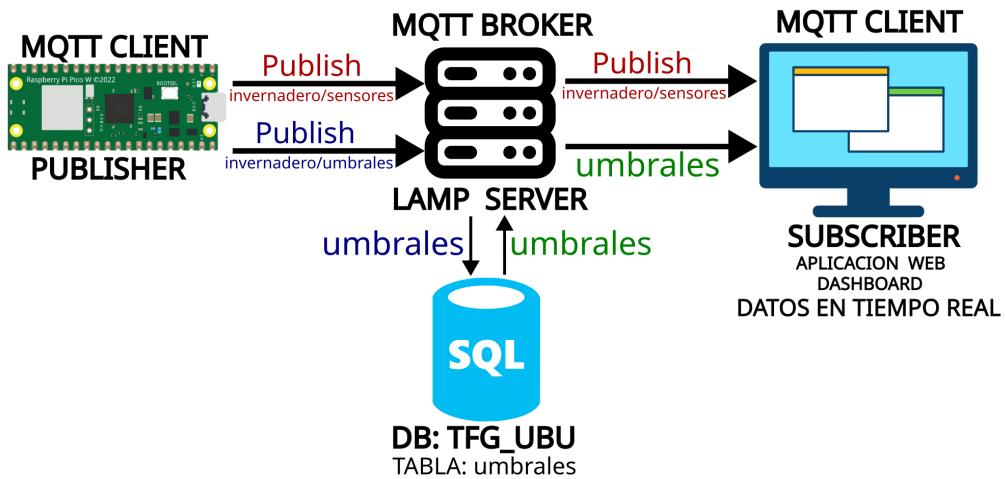


Figura 5.22: Mediante MQTT extrae los datos para mostrarlos en tiempo real.

El panel de control está disponible para su acceso a través del siguiente enlace: [InverIoT Dashboard](#)

NodeMqtt

nodeMqtt es un servicio en Node.js [34] que escucha los topics **invernadero/sensores** e **invernadero/umbrales**. Utiliza la librería MQTT.js [32]. Los datos son enviados por la placa Raspberry Pi Pico W RP2040 [11], que recopila valores de sensores. El servicio **nodeMqtt** captura, formatea y luego inserta estos datos en la base de datos MySQL [33] para los valores de sensores, además de actualizar los umbrales correspondientes.

Está conformado por los siguientes archivos:

- **index.js:** Es el punto de entrada del código de la aplicación.
- **package.json:** Es un archivo de configuración que describe la aplicación y sus dependencias.

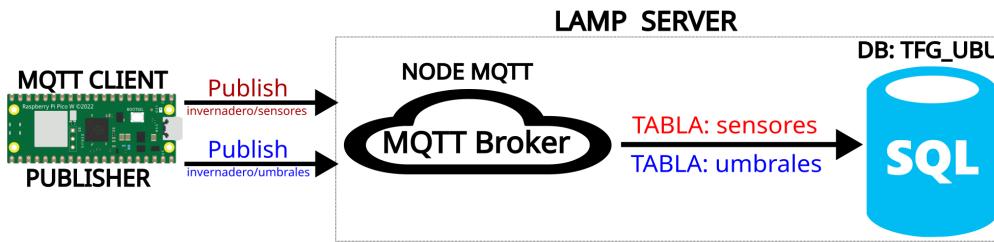


Figura 5.23: MQTT con Node.js para almacenar datos de los sensores.

6. Trabajos relacionados

Para comprender plenamente la contribución y novedad de este proyecto, es esencial explorar investigaciones y desarrollos previos relacionados.

Monitorización de la energía consumida mediante Raspberry Pi para sistema domótico

Se trata de un TFG desarrollado en la Universidad Carlos III de Madrid que trata sobre un sistema de monitorización de energía consumida para una instalación domótica mediante el uso de una Raspberry Pi, en el que se debe realizar el diseño del circuito y la programación correspondientes para poder extraer datos de diez sensores de corriente de forma simultánea.

Cada sensor recoge información correspondiente a una línea del cuadro de electricidad donde irá instalado.

En el desarrollo de este proyecto se pueden diferenciar dos partes:

- Montaje del circuito que lleva los datos obtenidos de los sensores de corriente hacia la Raspberry Pi para que puedan ser gestionados adecuadamente.
- Configuración de la Raspberry Pi para la lectura y procesado de los datos obtenidos a través de cada uno de los sensores. Tales datos son guardados en diferentes ficheros dependiendo del canal de los que provengan, denominados datos_canalX.txt, donde X es el número del sensor.

Introduciendo un comando determinado el sistema es capaz de:

- Modificar el tiempo existente entre el recojo de cada muestra para un canal específico, que es guardado en un fichero de tiempos para su continua lectura.
- Recoger datos a partir de un canal, fecha y hora, de forma que se lee el fichero datos_canalX correspondiente al canal introducido, y se guardan en otro fichero todos aquellos datos posteriores a la fecha y hora introducidos.

Podemos ver su referencia en la bibliografía en la referencia [4].

En la columna **Proy1** de la tabla 6.1 podemos ver algunas características de este proyecto.

Diseño e implementación de un sistema domótico basado en Raspberry Pi

Se trata de un TFG desarrollado en la Universidad Carlos III de Madrid que trata sobre un sistema domótico de bajo coste, basado en código libre (UNIX/LINUX). Dicho sistema permite al usuario manejar de forma fácil y sencilla elementos finales como motores y leds.

El elemento de software principal es el sistema operativo Raspbian (basado en Debian), aunque ahora es llamado Raspberry Pi OS. Tal sistema operativo es instalado en la RaspberryPi.

El segundo elemento software importante es el servidor web Apache que gestiona la aplicación web desarrollada bajo el patrón de desarrollo Modelo-Vista-Controlador (MVC).

El sistema domótico está compuesto por varios sus-sistemas:

- La aplicación web.
- La base de datos.
- El conjunto de sensores y actuadores.

La aplicación web se ha desarrollado sobre un servidor LAMP que es accesible desde el exterior y que actua de interfaz sobre el usuario del sistema domótico y le propio sistema.

Algunas acciones que puede realizar el usuario mediante este sistema:

- Consultar temperatura y humedad.
- Controlar apertura puertas/persianas.
- Consultar histórico de temperaturas y humedades.
- Consultar movimiento en una habitación.
- Controlar iluminación interior.

Podemos ver su referencia en la bibliografía en la referencia [39].

En la columna **Proy2** de la tabla 6.1 podemos ver algunas características de este proyecto.

6.1. Fortalezas y debilidades este proyecto

Las fortalezas clave del proyecto son:

- Todos los elementos utilizados en este proyecto son fácilmente accesibles, tanto en términos de software como de hardware.
- La elección de la Raspberry Pi Pico W fue crucial gracias a su bajo consumo de energía, su compacto tamaño que facilita la portabilidad y su capacidad de conectividad Wifi.
- La versatilidad de la Raspberry Pi Pico W, con su capacidad de conectividad Wifi, ha posibilitado tanto el envío de datos como la recepción de órdenes. Además, su flexibilidad se evidencia en la implementación exitosa de MQTT en el proyecto.
- La modularidad del sistema permite una fácil escalabilidad, ya que es posible agregar nuevos sensores o funcionalidades sin mayores complicaciones.
- La presentación gráfica de datos en tiempo real en una pantalla OLED y la posibilidad de acceder a través de una aplicación de escritorio y un panel web ofrecen múltiples formas de acceder y visualizar la información recopilada.

Las principales debilidades de este proyecto son:

- La prueba del sistema se realizó en un ambiente no controlado, lo que puede afectar la precisión de los datos y la generación de alertas. Esto podría limitar la fiabilidad de las respuestas del sistema en condiciones reales.
- La funcionalidad del proyecto está intrínsecamente ligada a la conectividad Wifi de la Raspberry Pi Pico W. Problemas en la red Wifi podrían afectar la transmisión de datos y la recepción de órdenes.
- Aunque se han incluido sensores relevantes, la variedad podría ser limitada. La adición de más tipos de sensores podría mejorar la capacidad de monitoreo y la precisión de los datos recopilados.
- A medida que se agregan más sensores y funcionalidades, la capacidad de la Raspberry Pi Pico W podría alcanzar sus límites. En proyectos a mayor escala, se podría requerir hardware más potente.

Características	MICM	Proy1	Proy2
Proyecto libre	✓	✓	✓
Menor precio en modelo de Raspberry Pi	✓		
Menor consumo de energía de Raspberry Pi	✓		
Escalabilidad	✓	✓	✓
Modularidad	✓	✓	✓
Más opciones de visualización de datos	✓		
Mayor potencia en modelo de Raspberry Pi		✓	✓
Más opciones de acceso de datos	✓		
Interacción multiplataforma	✓	✓	✓
WiFi entre elementos	✓	✓	✓
Almacenamiento de datos	✓	✓	✓

Tabla 6.1: Comparativa de las características de los proyectos.

7. Conclusiones y Líneas de trabajo futuras

Con esta sección, finaliza la documentación del proyecto, detallando los objetivos alcanzados. A continuación, se presentarán algunas posibles direcciones para investigaciones futuras que podrían ampliar y mejorar significativamente el proyecto.

7.1. Conclusiones

Al concluir el proyecto, podemos afirmar lo siguiente:

- Se lograron alcanzar los objetivos establecidos al inicio del proyecto, los cuales se centraban en la recopilación y visualización de datos ambientales en invernaderos de cannabis medicinal.
- La elección de la Raspberry Pi Pico W demostró ser acertada, ya que su bajo consumo de energía, tamaño compacto y conectividad WiFi facilitaron la implementación a un costo razonable.
- Tanto los componentes hardware como el software seleccionados son de fácil adquisición, permitiendo replicar el sistema con relativa facilidad. Esto contribuye a la accesibilidad de la tecnología en el ámbito de los invernaderos.
- La capacidad de conectividad WiFi de la Raspberry Pi Pico W permitió la transmisión de datos y la recepción de órdenes de manera eficiente. La implementación de MQTT agregó una capa adicional de versatilidad al sistema.

- Se logró una amplia variedad de formas de visualizar los datos ambientales, proporcionando a los usuarios una comprensión más completa y detallada de las condiciones dentro del invernadero.
- Se identifican oportunidades para futuras mejoras, como la expansión de la capacidad de monitoreo, la integración de más sensores especializados y la implementación de alertas automáticas ante condiciones críticas.
- Este proyecto brinda una solución económica y accesible para invernaderos de cannabis medicinal, con potencial para adaptarse a otros entornos de cultivo.

7.2. Líneas de trabajo futuras

El proyecto se ha desarrollado de manera integral, abordando de manera efectiva la funcionalidad deseada y proporcionando una solución adecuada a las necesidades planteadas. Aunque se ha logrado un producto sólido, existe el potencial para mejorar en diversos aspectos.

- Integrar sensores adicionales para monitorizar parámetros específicos del entorno del invernadero, como la concentración de CO₂, la calidad del suelo, o la presencia de plagas. Esto proporcionaría un monitoreo más completo y detallado.
- Desarrollar algoritmos de análisis de datos más avanzados para identificar patrones, tendencias o anomalías en la información recopilada. Esto proporcionaría insights más profundos sobre el comportamiento del invernadero.
- Mejorar la interfaz de usuario para proporcionar funcionalidades más avanzadas, gráficos interactivos, información estadística y herramientas de análisis. Esto facilitaría la interpretación de los datos y la toma de decisiones.
- Integrar un sistema de riego automatizado que pueda ajustarse en tiempo real según las condiciones ambientales registradas. Esto optimizaría el consumo de agua y mejoraría la eficiencia del proceso de cultivo.
- Extender las pruebas y adaptar el sistema para su uso en otros tipos de cultivos, permitiendo su aplicación en entornos agrícolas más amplios.

- Reforzar la seguridad del sistema, incluyendo medidas para proteger la integridad de los datos y garantizar la privacidad de la información recopilada.

Bibliografía

- [1] Amazon. Batería externa 20000 mah. <https://www.amazon.es/YWTESCH-20000mAh-Cargador-Powerbank-Compatible/dp/B08B4HHJVH>.
- [2] Amazon. Enrutador wifi usb 4g lte, router wifi portátil de 150 mbps. <https://www.amazon.com/-/es/Enrutador-enrutador-exteriores-computadoras-port%C3%A1tiles/dp/B0C3BKF7HP/>.
- [3] Apache. Http server project. <https://httpd.apache.org/>.
- [4] Alejandro Barón Cuevas. Monitorización de la energía consumida mediante raspberry pi para sistema domótico, 09 2017. Tesis de maestría.
- [5] Bitbucket. Bitbucket. <https://bitbucket.org/>.
- [6] datasheetspdf. *KY 016 RGB 5mm LED module.* <https://datasheetspdf.com/mobile/1402027/Joy-IT/KY-016/1>.
- [7] Equipo de Desarrollo de MicroPython. Micropython documentation, 1 2024. <https://github.com/micropython/micropython/wiki>.
- [8] DFROBOT. Capacitive soil moisture sensor, 1 2024. https://wiki.dfrobot.com/Capacitive_Soil_Moisture_Sensor_SKU_SEN0193.
- [9] Gettys J. Mogul J. Frystyk H. Masinter L. Leach P. & Berners-Lee T. (1999) Fielding, R. *Hypertext Transfer Protocol – HTTP/1.1*, 1999. <https://www.ietf.org/rfc/rfc2616.txt>.
- [10] Raspberry Pi Foundation. Gpio. <https://projects.raspberrypi.org/en/projects/physical-computing/1>.

- [11] Raspberry Pi Foundation. Raspberry pi pico w. <https://www.raspberrypi.com/documentation/microcontrollers/raspberry-pi-pico.html#raspberry-pi-pico-w>.
- [12] Raspberry Pi Foundation. Raspberry pi pico w - datasheet. <https://datasheets.raspberrypi.com/picow/pico-w-datasheet.pdf>.
- [13] Fritzing. Electronics made easy. <https://fritzing.org/>.
- [14] Git. free and open source distributed version control system. <https://git-scm.com/>.
- [15] Github. Github. <https://github.com/>.
- [16] GitMind. Metodología cascada: Beneficios, uso y plantillas gratuitas. <https://gitmind.com/es/metodologia-cascada.html>.
- [17] Handsontec. *BH1750 Ambient Light Sensor Module*. <https://www.handsontec.com/dataspecs/sensor/BH1750%20Light%20Sensor.pdf>.
- [18] Hostingplus. Modelo de prototipos: ¿qué es y cuáles son sus etapas? <https://www.hostingplus.com.es/blog/modelo-de-prototipos-que-es-y-cuales-son-sus-etapas/>.
- [19] HPE. Hp proliant servers. https://www.hpe.com/emea_europe/en/hpe-proliant-servers.html.
- [20] IEEE. History of ieee. <https://www.ieee.org/about/ieee-history.html>.
- [21] IEEE. Ieee 802.15 working group for wireless specialty networks (wsn). <https://www.ieee802.org/15/>.
- [22] IEEE. Ieee 802.3 ethernet working group. <https://www.ieee802.org/3/>.
- [23] IEEE. *Normativa IEEE802.11*. <https://standards.ieee.org/ieee/802.11/5536/>.
- [24] IEEEEXPLORE. *Analysis of UART Communication Protocol*. <https://ieeexplore.ieee.org/document/9936199>.
- [25] IEEEEXPLORE. *An introduction to I2C and SPI protocols*. <https://ieeexplore.ieee.org/document/4762946>.

- [26] IEEEEXPLORE. *MQTT Protocol: Fundamentals, Tools and Future Directions.* <https://ieeexplore.ieee.org/document/8931137>.
- [27] LaTeX. The latex project. <https://www.latex-project.org/>.
- [28] Led-atomant. Panel solar ip65 de 8w y 5v con cargador usb tipo c. <https://led-atomant.com/es/jardin-y-solar/506-panel-solar-ip65-de-8w-y-5v-con-cargador-usb-tipo-c.html>.
- [29] Linux.org. Linux. <https://www.linux.org/>.
- [30] Microsoft. *C Sharp documentation.* <https://learn.microsoft.com/en-us/dotnet/csharp/>.
- [31] Microsoft. *Introduction to Hyper-V.* <https://learn.microsoft.com/en-us/virtualization/hyper-v-on-windows/about/>.
- [32] MQTT.js. client library for the mqtt protocol. <https://www.npmjs.com/package/mqtt>.
- [33] MySQL. The world's most popular open source database. <https://www.mysql.com/>.
- [34] Node.js. Open source cross platform javascript runtime environment. <https://nodejs.org/en>.
- [35] Jupyter Notebook. Free software, open standards, and web services for interactive computing across all programming languages. <https://jupyter.org/>.
- [36] Overleaf. Overleaf. <https://www.overleaf.com/>.
- [37] Pandas. Python for data analysis. <https://pandas.pydata.org/>.
- [38] PHP. A popular general-purpose scripting language. <https://www.php.net/>.
- [39] Héctor Santos Senra. Diseño e implementación de un sistema domótico basado en raspberry pi, 02 2017. Tesis de maestría.
- [40] SonarSource. Sonarcloud. <https://www.sonarsource.com/products/sonarcloud/>.
- [41] Sparkfun. *DHT22 Digital-output relative humidity & temperature sensor/module.* <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>.

- [42] SSH. Communications security for humans, systems, and networks. <https://www.ssh.com/>.
- [43] SVN. Apache subversion. <https://subversion.apache.org/>.
- [44] Telegram. Telegram. <https://telegram.org/>.
- [45] Telegram. Telegram api. <https://core.telegram.org/api>.
- [46] Telegram. Telegram bots. <https://core.telegram.org/bots>.
- [47] Ubuntu. Ubuntu. <https://ubuntu.com/>.
- [48] waveshare. *Pantalla oled sh1106 128x64 1.3 pulgadas datasheet*. <https://www.waveshare.com/w/upload/e/e3/1.3inch-SH1106-OLED.pdf>.
- [49] WorlTimeAPI. Simple web service which returns the current local time for a given timezone. <http://worldtimeapi.org/>.