



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 ...

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 ...

Índice general

Índice general	iii
Índice de figuras	iv
Índice de tablas	v
5. Aspectos relevantes del desarrollo del proyecto	1
5.1. Motivación del proyecto	1
5.2. Formación necesaria	2
5.3. Metodología	3
5.4. Desarrollo del proyecto	4
Bibliografía	9

Índice de figuras

5.1.	Logo de Scrum.	3
5.2.	Conexiones.	4
5.3.	Aplicación de escritorio InverIoT.	5
5.4.	Historial de los datos.	6
5.5.	Gráfica mostrando los datos en un intervalo de fechas.	6
5.6.	Intensidad de luz superando los umbrales.	7
5.7.	Vista del historial: al cargar, muestra los datos del día actual.	7

Índice de tablas

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 [?] 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 [?] 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 [?], 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 [?] 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 [?] mide la temperatura y humedad ambiente, el BH1750 [?] evalúa la intensidad lumínica, y el sensor de humedad de suelo [?] 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) [?] 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) [?] y SPI (Serial Peripheral Interface) [?].
- **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) [?] para la transmisión de datos en la web y MQTT (Message Queuing Telemetry Transport) [?] 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) [?] establecen estándares para varios protocolos de comunicación, garantizando la consistencia y la interoperabilidad en la industria. Por ejemplo, el IEEE 802.3 [?] define estándares para Ethernet, mientras que el IEEE 802.15 [?] 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](#) [?] 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 [?] como sistema operativo, Apache [?] como servidor web, MySQL [?] como sistema de gestión de bases de datos y PHP [?] 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.

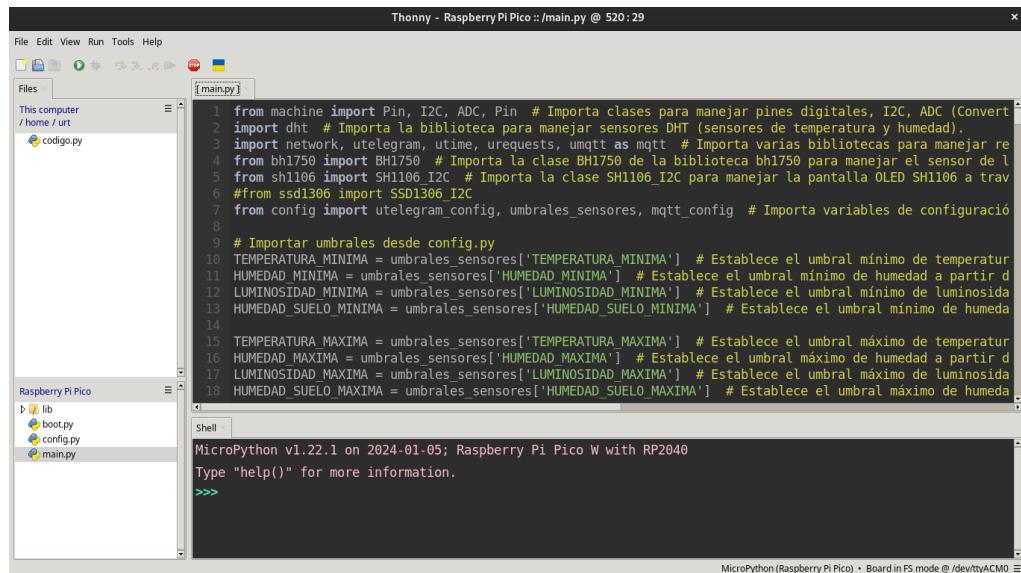


Figura 4.1: IDE Thonny.

Entorno de desarrollo Python

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

Jupyter Notebook [?] 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 [?] para manipulación y procesamiento de datos.

4.2. Control de datos

APIS

- **API de Telegram:** La API de Telegram [?], accesible a través de la URL <https://api.telegram.org/bot<token>>, juega un papel fundamental en la integración de bots de Telegram [?] 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 [?], 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 [?] 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.

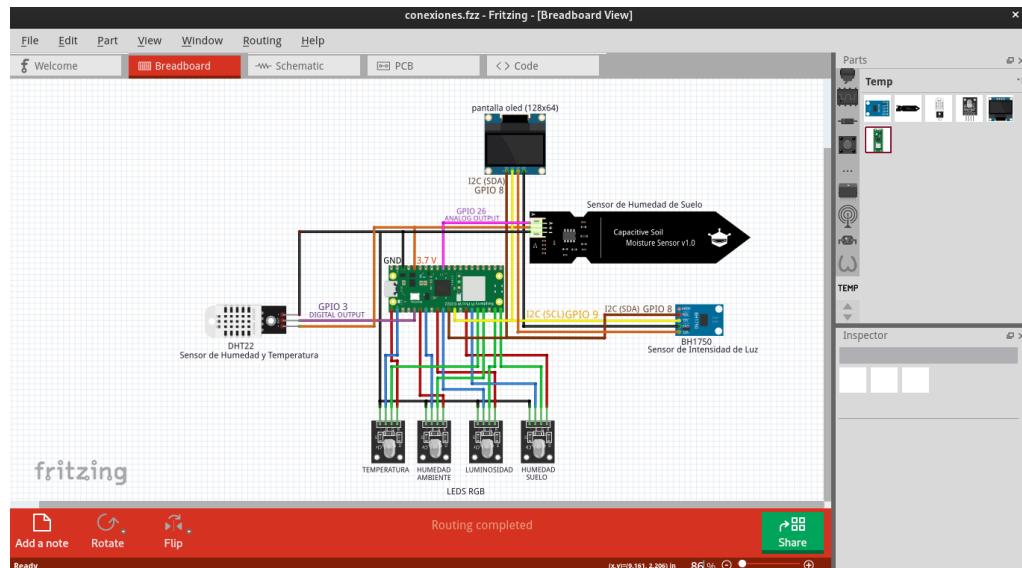


Figura 4.2: Esquema de conexiones diseñado con Fritzing.

4.4. Metodologías

Modularidad

La modularidad es un principio clave en la metodología de desarrollo que implica la división de un sistema en módulos independientes y autónomos, cada uno de los cuales realiza una función específica y bien definida. En el contexto del diseño de un sistema económico IoT para la monitorización de invernaderos de cannabis medicinal, la aplicación de la modularidad implica organizar el software y el hardware en componentes distintos y fácilmente intercambiables.

Características de la Modularidad:

- **Independencia Funcional:** Cada módulo opera de forma independiente, realizando una tarea específica sin depender excesivamente de otros. Esto facilita la comprensión individual y la posibilidad de actualizar o cambiar un módulo sin afectar el funcionamiento general.
- **Interconexión Estandarizada:** Aunque los módulos operan de manera independiente, la interconexión entre ellos sigue estándares predefinidos. Esto facilita la comunicación y la interoperabilidad entre los componentes del sistema.
- **Facilita el Mantenimiento:** La modularidad simplifica el mantenimiento del sistema, ya que las actualizaciones o correcciones pueden realizarse en módulos específicos sin afectar otras partes del sistema. Esto mejora la capacidad de respuesta y reduce el riesgo de errores inadvertidos.
- **Escalabilidad:** La estructura modular permite una fácil escalabilidad del sistema. Se pueden agregar nuevos módulos o funciones sin afectar negativamente la estructura existente, lo que facilita la adaptación del sistema a cambios futuros.
- **Reutilización de Componentes:** Los módulos pueden diseñarse para ser reutilizables en diferentes partes del proyecto o incluso en proyectos futuros. Esto reduce la redundancia y fomenta la eficiencia en el desarrollo.
- **Facilita la Colaboración:** La división en módulos facilita la colaboración entre diferentes equipos o desarrolladores, ya que cada grupo puede trabajar de manera independiente en su propio módulo, siempre y cuando se respeten las interfaces definidas.

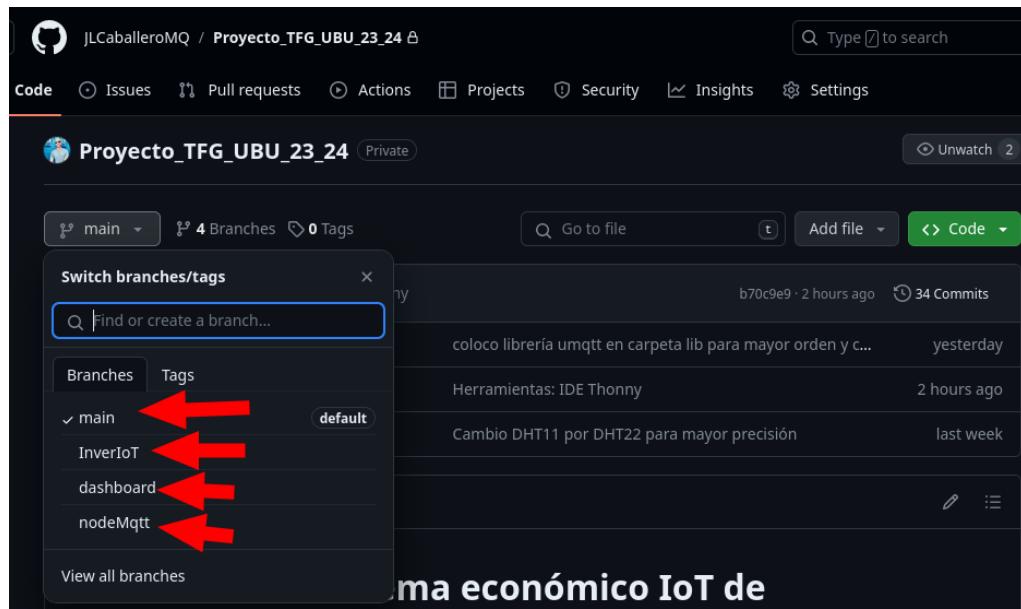


Figura 4.3: Estructura de ramas en GitHub aplicando modularidad.

Cada rama representa un módulo o característica específica del proyecto, permitiendo un desarrollo paralelo y aislado. Las ramas se fusionan con la rama principal una vez que las funcionalidades están completas y probadas. Esta estrategia facilita la gestión modular del proyecto y la colaboración entre desarrolladores.

4.5. Entorno de desarrollo del Proyecto

Control de versiones o CVS, Concurrent Versioning System

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

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

Bitbucket [?] 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 [?] 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 [?] 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 [?], 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 [?] 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

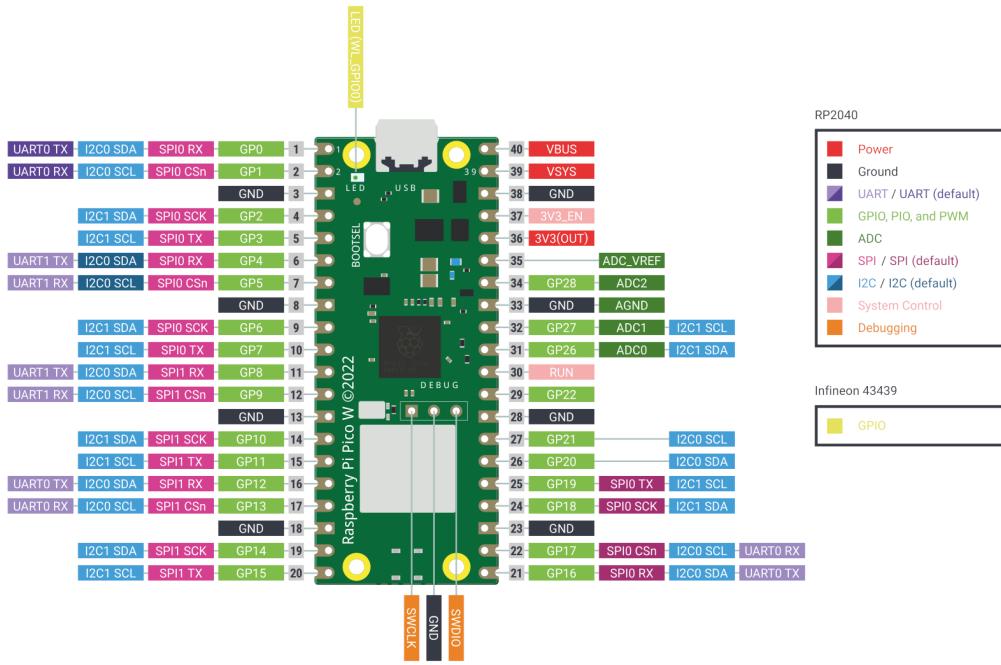


Figura 4.4: Raspberry Pi Pico W Pinout.

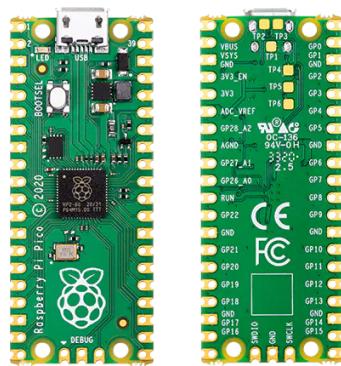


Figura 4.5: Vistas Frontal y posterior de Raspberry Pi Pico W.

Pantalla oled I2C de 128x64 píxeles y 1.3 pulgadas

La pantalla OLED [?] 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 4.6: Pantalla oled.

Sensor de luz BH1750

El sensor BH1750 [?] 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 4.7: Vista frontal del sensor BH1750.



Figura 4.8: Lado posterior del sensor BH1750.

Sensor de temperatura y humedad DHT22

El sensor DHT22 [?] (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 4.9: Sensor DHT22.

Sensor de humedad de suelo

El sensor de humedad de suelo Higrometro V1.2 [?] 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 4.10: Sensor de humedad de suelo Higrometro V1.2.

KY-016 FZ0455 Módulo led RGB de 3 colores

El módulo KY-016 [?] 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

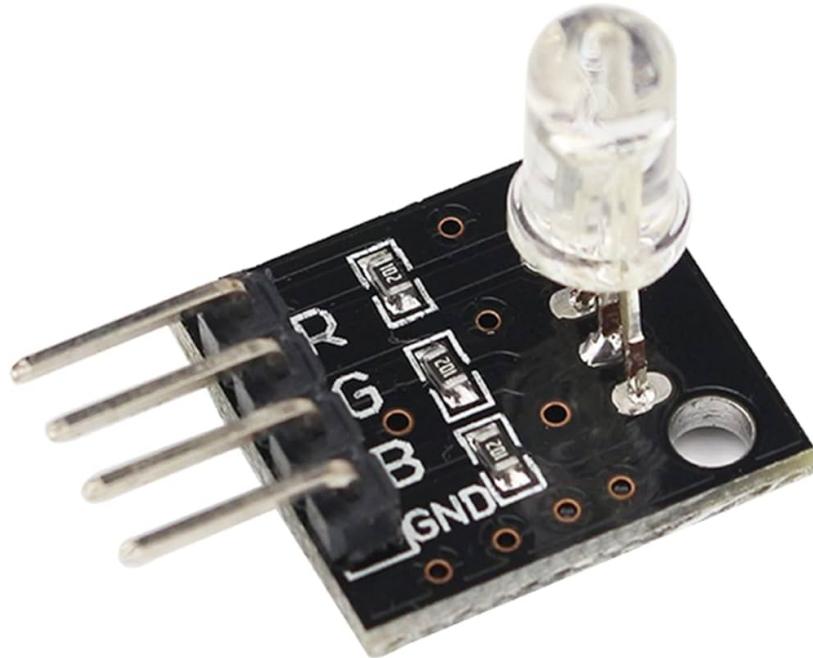


Figura 4.11: Módulo led RGB KY-016.

Enrutador WiFi USB 4G LTE

El Enrutador WiFi [?] 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 4.12: Enrutador WiFi USB 4G LTE.

Batería externa 20000 mAh

Es un Power Bank [?] 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 4.13: Batería externa 20000 mAh.

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

Este Panel Solar [?] 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 4.14: 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 [?], el BH1750 [?] y el sensor de humedad del suelo [?], 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 [?] también ha requerido conocimientos específicos en redes y protocolos de comunicación.

La integración de MQTT [?] 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 implementar el enfoque Scrum [?] como metodología principal para llevar a cabo el proyecto de manera iterativa y ágil. La intención fue recrear un entorno de trabajo lo más fiel posible a las dinámicas laborales, a pesar de las limitaciones de interacción con un equipo real. Se establecieron reglas generales mínimas:

- Se estructuraron las tareas en sprints semanales.
- Después de cada sprint semanal, se entregó trabajo de manera incremental.
- En cada revisión de sprint, se planificaron las tareas para la siguiente semana.
- A través de revisiones semanales, el proyecto se mantuvo flexible, permitiendo la integración de cambios en trabajos pequeños y mejorando continuamente.
- La estimación de tiempos por tareas se llevó a cabo utilizando el método Kanban basado en la dificultad.
- El estado de los issues se gestionó a través de un Kanban, reflejando la evolución del trabajo.



Figura 5.1: Logo de Scrum.

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.

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

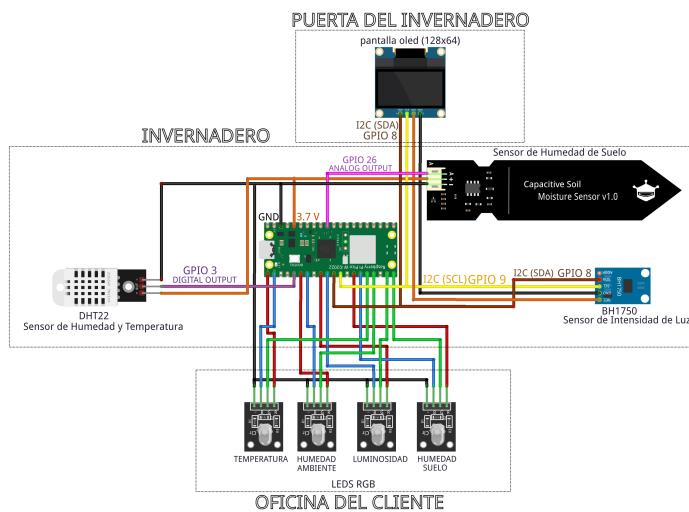


Figura 5.2: Conexiones.

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

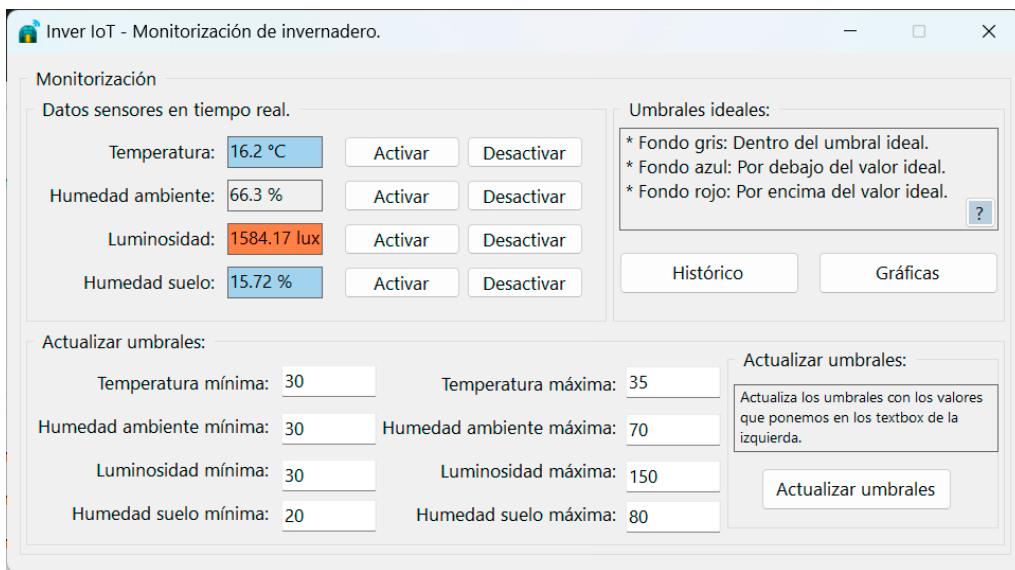


Figura 5.3: Aplicación de escritorio **InverIoT**.

Inver IoT - Histórico de datos.							
	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 5.4: Historial de los datos.



Figura 5.5: 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. La plataforma incluye una gráfica en tiempo real y la capacidad de acceder a un historial con filtro de fecha. Los umbrales utilizados se extraen de la tabla **TFG_UBU** en la base de datos MySQL [?].

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



Figura 5.6: Intensidad de luz superando los umbrales.

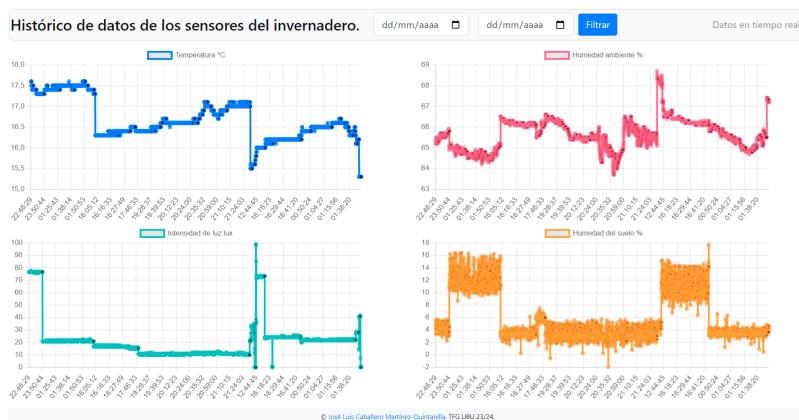


Figura 5.7: Vista del historial: al cargar, muestra los datos del día actual.

NodeMqtt

nodeMqtt es un servicio en Node.js [?] que escucha los topics **invernadero/sensores** e **invernadero/umbrales**. Los datos son enviados por la placa Raspberry Pi Pico W RP2040 [?], que recopila valores de sensores. El servicio **nodeMqtt** captura, formatea y luego inserta estos datos en la base de datos MySQL [?] 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.