

# Implementación de un sistema invernadero inteligente

Rojas Pérez José Gabriel <sup>1[0009-0005-1227-7335]</sup>, Lozano Cruz José Ángel <sup>2[0009-0009-3034-924X]</sup>  
y Jimenez Bonilla Giovani <sup>3[0009-0009-7064-477X]</sup>

<sup>1</sup> Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla,  
Puebla 72570, México.

**Abstract.** Este proyecto presenta el desarrollo de un sistema de control térmico para un invernadero automatizado, utilizando un microcontrolador ESP32, un sensor de temperatura LM35, un ventilador, un foco incandescente, un display de 7 segmentos y una interfaz gráfica en Python. El objetivo principal fue comparar el comportamiento y desempeño de un sistema de lazo abierto frente a uno de lazo cerrado, evaluando su estabilidad y capacidad de mantener condiciones óptimas de temperatura dentro del invernadero. En el modo abierto, el usuario controla manualmente los actuadores desde la interfaz; mientras que, en el modo cerrado, el sistema regula de forma autónoma el encendido del ventilador cuando se supera el umbral térmico definido. Los resultados experimentales demostraron un funcionamiento estable, una comunicación efectiva entre hardware y software, y una respuesta confiable ante cambios de temperatura. Además, el proyecto promueve el aprendizaje práctico de conceptos de automatización, control y electrónica, proporcionando una herramienta didáctica replicable para entornos educativos y de laboratorio.

**Palabras clave:** ESP32, LM35, invernadero inteligente, control de lazo abierto, control de lazo cerrado, IoT, automatización, Python, sistema embebido, control térmico.

## 1 Introducción

El presente proyecto surge a partir de la necesidad de diseñar y construir, con recursos accesibles y asequibles dentro del entorno académico, un sistema capaz de regular la temperatura de un invernadero y evaluar la estabilidad de su funcionamiento bajo condiciones reales de operación. La automatización y el control de variables ambientales en invernaderos representan una de las aplicaciones más relevantes dentro del ámbito del Internet de las Cosas (IoT), debido a su impacto directo en la eficiencia energética, la optimización del crecimiento vegetal y la reducción del esfuerzo humano.

Con este propósito, se desarrolló un prototipo experimental basado en un microcontrolador ESP32, el cual funge como unidad central de procesamiento y comunicación. El sistema incorpora un sensor de temperatura LM35 para la medición en tiempo real, un display de 7 segmentos para la visualización local de los valores registrados, una

bombilla de corriente alterna como fuente de calor y un ventilador para la disipación térmica. Todos estos elementos se integraron mediante una interfaz gráfica de usuario (GUI) desarrollada en Python, la cual permite monitorear y controlar el sistema tanto en modo manual como automático. El trabajo propone y compara dos estrategias de control: el sistema de lazo abierto y el sistema de lazo cerrado. En el primer caso, el usuario establece manualmente la consigna de temperatura y acciona los actuadores (foco y ventilador) desde la interfaz, con el objetivo de observar el comportamiento del sistema ante variaciones térmicas, analizar los tiempos de respuesta y determinar los márgenes de estabilidad.

En cambio, el sistema de lazo cerrado incorpora un mecanismo de retroalimentación, donde el sensor LM35 envía información al ESP32, el cual toma decisiones automáticas: al superar la temperatura umbral definida, el microcontrolador activa el ventilador para mantener el equilibrio térmico dentro del invernadero. Durante la fase de pruebas, se realizaron ensayos de calibración y verificación de la estabilidad del sensor, así como pruebas de respuesta térmica mediante el encendido y apagado de la bombilla hasta alcanzar el punto de actuación del ventilador. Posteriormente, se evaluó la usabilidad del sistema a través de la interfaz Python, contrastando la operación manual frente al control automático. Estos ensayos permitieron detectar áreas de mejora en el código, ajustar los límites de actuación y optimizar la lógica de redondeo en la presentación de datos, garantizando una lectura precisa y coherente entre el sensor y el display.

Además de los aspectos técnicos, se documentaron criterios de seguridad eléctrica, recomendaciones de diseño estructural y procedimientos de montaje, con el fin de asegurar una reproducción segura y confiable del prototipo en un entorno educativo o de laboratorio. El desarrollo de este sistema no solo constituye una herramienta didáctica para comprender los fundamentos del control automático y la electrónica aplicada, sino que también representa un punto de partida para futuras ampliaciones, tales como la integración de sensores de humedad, luz o CO<sub>2</sub>, la conexión inalámbrica mediante MQTT y la implementación de algoritmos de inteligencia artificial para la gestión predictiva del entorno del invernadero. Finalmente, el resultado de este proyecto demuestra la viabilidad de construir un sistema de control térmico funcional, de bajo costo y de fácil replicación, utilizando componentes accesibles y software de código abierto. La experiencia adquirida en la implementación de las estrategias de lazo abierto y lazo cerrado brinda una comprensión integral de los principios de control, retroalimentación y automatización, aportando una base sólida para el desarrollo de futuros sistemas de invernaderos inteligentes.

## 2. Estado del arte

Durante la implementación de un sistema invernadero, recapitulamos trabajos referentes a nuestro proyecto por el cual con base a la investigación de Rocamora-Osorio et al. (2025) se implementó un sistema para monitorear el crecimiento de plantas de *Cannabis sativa* L. en invernaderos utilizando visión artificial basada en cámaras de vigilancia integradas a un sistema IoT embebido que automatiza la captura y almacenamiento de

imágenes. En cual se obtuvieron los siguientes resultados: se demostró que es posible implementar un sistema integrado de visión artificial e IoT para monitorizar el crecimiento y detectar el estrés hídrico en el cáñamo industrial, sin necesidad de cámaras sofisticadas ni conocimientos avanzados de programación. El sistema se basa en plataformas de código abierto, sensores de bajo coste y cámaras de vídeo convencionales, lo que permite su replicación en diversos entornos. Este enfoque facilita un análisis no invasivo, automático y eficiente del desarrollo del cultivo, sentando así las bases para sistemas de apoyo a la toma de decisiones agrícolas sencillos y económicos. Además, los trabajos realizados por Galon et al. (2025) menciona que la aplicación IoT permite la monitorización remota de parámetros. El sistema monitoriza con precisión el entorno del invernadero y mantiene los umbrales establecidos por el usuario. Basándose en los datos recopilados sobre el crecimiento de las plantas, el sistema mantiene la vida vegetal dentro del invernadero. Es necesario seguir mejorando para conocer el estado del suministro de agua y predecir condiciones climáticas extremas.

Hablando un poco más de trabajos mexicanos, nos es posible rescatar los resultados de (Pacheco Zavala et al., n.d.) donde se implementa un sistema embebido que a través de un microcontrolador que mediante redes Wifi intercambia información con un gestor denominado Broker mediante el protocolo MQTT (Message Queuing Telemetry Transport). Con sus resultados nos menciona que, el envío de información se realiza vía WiFi y el único cableado existente es para alimentar los sistemas embebidos mediante eliminadores de 5V. Una mejora futura es la alimentación de los sistemas embebidos mediante baterías lo que permitirá un comportamiento modular y eliminar el cableado. Por lo que concluye con la implementación exitosa de sensores y actuadores estratégicamente ubicados ha permitido obtener datos en tiempo real sobre las condiciones ambientales dentro del invernadero y realizar acciones de control automatizado. La aplicación de esta tecnología podría impactar en un crecimiento más saludable de las plantas, un ahorro significativo de agua y energía, y una reducción de riesgos en la cosecha. Sin embargo, para trabajos futuros, se sugiere continuar optimizando los sensores, integrar inteligencia artificial y avanzar en la automatización para una gestión aún más precisa y autónoma Pacheco Zavala et al. (n.d.).

**Tabla 1.** Comparación de trabajos relacionados con el sistema invernadero.

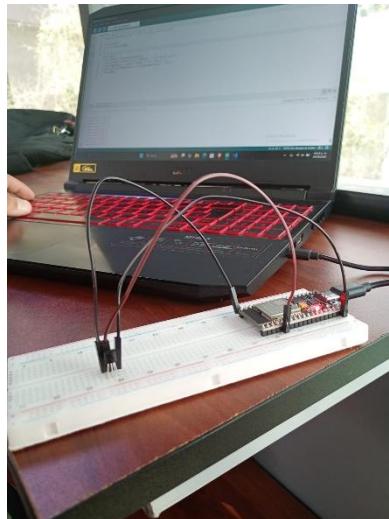
Artículo	Objetivo de la investigación	Relación con nuestro proyecto
Automated Monitoring and Control System of Solar Greenhouse Using ESP32 and Blynk Application	Desarrollar un sistema IoT con ESP32 y Blynk para monitorear y controlar remotamente variables de un invernadero solar.	Se relaciona por el uso del ESP32 como microcontrolador principal y la automatización de variables ambientales en un entorno de invernadero, coincidiendo con el enfoque de control y monitoreo remoto del proyecto.

Monitoreo inteligente de invernadero a escala basado en el Internet de las Cosas (IoT)	Diseñar una arquitectura modular con nodos ESP32 y MQTT/Node-RED para monitoreo y control en tiempo real (T°, HR, humedad de suelo).	Está vinculado al proyecto por la implementación de comunicación IoT basada en ESP32 y MQTT, así como por la obtención de datos ambientales en tiempo real para la toma de decisiones automatizada dentro del invernadero.
Automated IoT-Based Monitoring of Industrial Hemp in Greenhouses Using Open-Source Systems and Computer Vision	Integrar visión por computadora y sensoria ambiental en una plataforma open-source para seguimiento automatizado del cultivo.	Se relaciona directamente por la combinación de IoT y visión artificial en el monitoreo de cultivos, aportando un antecedente sobre el uso de sistemas abiertos y sensores económicos en el control inteligente de invernaderos.

### 3. Metodología

#### 3.1. Implementación de un invernadero con microcontrolador

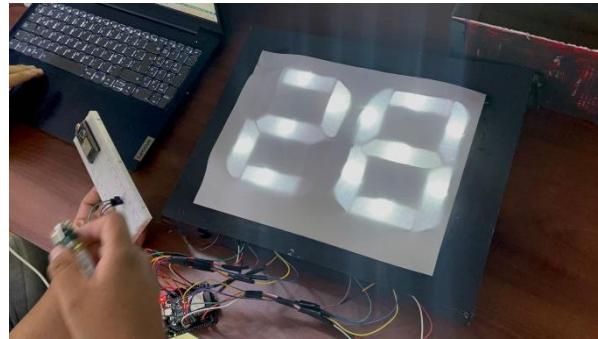
Para esta parte comenzamos mejorando el display de 7 segmentos que va enlazado con nuestro proyecto ya que es pilar fundamental para el contador de temperatura. Además, se integró un sensor de temperatura el cual conectamos a través de un protoboard a la placa de ESP32. Al igual observamos que la temperatura se mostraba tanto en la salida de IDE como en el display.



*Fig. 1 Sensor conectado a la placa ESP32.*

Posteriormente, se realizó el ajuste de temperatura donde en la consola del IDE nos lo mostraba de manera decimal por lo que ajustamos el código para que en el display se muestre si una temperatura se encuentra en el rango de 27.1 a 27.9, en este caso al ser

27.5 automáticamente se redondea a su siguiente número próximo que en este caso sería 28 como podemos observar en la fig.2.



*Fig. 2 Display conectado a la placa ESP32.*

### **3.1.1. Implementación de los componentes del invernadero**

Para lograr que nuestro invernadero vaya tomando la forma deseada usaremos una bombilla de corriente alterna y un ventilador, el cual son parte fundamental para comenzar a controlar la temperatura en este caso se aplicaran los dos tipos de enlaces como el enlace abierto y cerrado. Con la ayuda de una estructura de madera empezamos a diseñar las bases que tendrá nuestro proyecto. Se realizaron pruebas que avalan el correcto funcionamiento de la comunicación entre el foco y el ventilador con la interfaz, donde establecemos una temperatura requerida, si llega a esa temperatura el ventilador se activa de manera automática, cumpliendo la funcionalidad de un sistema de control de lazo cerrado que se explicará más adelante.

### **3.1.2. Sistema de lazo abierto**

El proyecto se requiere aplicar un sistema de control de lazo abierto en nuestro invernadero. Un sistema de control de lazo abierto se caracteriza por que no recibe ninguna información o retroalimentación sobre el estado de la variable, por lo regular estos se utilizan cuando la variable es predecible y tiene un amplio margen de error, ya que se puede calcular el tiempo o las veces que se debe de repetir el ciclo para completar el proceso (Mecafenix, 2023).

En nuestro caso, para lograr este sistema primero ajustamos el código de nuestro microcontrolador para que reciba las instrucciones que le daremos a través de una interfaz que desarrollamos en Python, donde de manera manual ingresaremos la temperatura deseada, donde el foco comenzara a dar calor a nuestro invernadero, posteriormente, tenemos la opción de controlar el estado de hibernación en caso de que la temperatura se haya excedido podremos activar un ventilador de manera manual.

### 3.1.3. Sistema de lazo cerrado

También, en nuestro proyecto se necesita implementar un sistema de control de lazo cerrado, que se empleara con el uso de nuestro sensor. Este sistema es más completo ya que recibe información sobre los estados que va tomando la variable. Esta retroalimentación se logra colocando sensores que envían información de puntos clave del proceso para que así pueda actuar de manera autónoma (Mecafenix, 2023).

A diferencia del enlace abierto, aquí establecemos una temperatura requerida donde a través de un sensor de temperatura LM35, activa de manera automática el ventilador, permitiendo tener un control más automático ya que no es necesaria nuestra presencia, así asegurándonos de que cuando la temperatura exceda los 25 C°, el sensor mandara la señal al microcontrolador de que es momento de refrescar o evitar que el calor exceda su punto máximo manteniendo protegido a nuestras plantitas.

## 3.2. Desarrollo de la interfaz del sistema de control

En el desarrollo de la interfaz para el control del invernadero nos hemos centrado principalmente en poder entender el funcionamiento de lo que se está haciendo en el invernadero y como sacarle provecho a sus hardware y software, pues como se observará en las imágenes hemos dividido la interfaz en varios bloques de funciones.

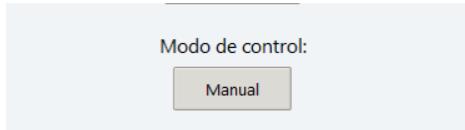


*Fig. 3 Interfaz gráfica con la implementación del sistema de control de ambos lazos (abierto y cerrado).*

Como se puede observar aquí en el centro tenemos la temperatura que está captando nuestro sensor, ya que la temperatura al ser una variable muy volátil ajustamos un límite nuevo para el sistema cerrado, permitiéndonos así adaptarnos a las necesidades sin tener que optar por cambiar el sistema completo.

### 3.2.1. Interfaz de sistema de control de lazo abierto

En este caso al momento de pensar en la interfaz se optó por el uso de bloques para una optimización de interfaces lo que nos permitió hacer un control de lazo abierto que en este caso es al momento de presionar el botón de automático, lo que esto cambiaría a manual.



*Fig. 3 Interfaz de control de lazo abierto*

En este caso al tener el modo de control en manual nos permite hacer:

- Prender foco
- Apagar foco
- Prender ventilador
- Apagar ventilador

Todo sin necesidad de que se siga una temperatura específica lo cual es útil en caso de que alguno de los componentes ya no funcione adecuadamente poder probarlo de manera más eficiente. En código por defecto la interfaz estará en modo automático o lazo cerrado en caso de que no sea así se ejecutará un bucle que nos permite mandar las señales para la activación de nuestros relés en el esp 32.

```
def cambiar_modo():
    if modo.get() == "Automático":
        modo.set("Manual")
        if ser: ser.write(b'M')
    else:
        modo.set("Automático")
        if ser: ser.write(b'A')
```

*Fig. 4 Bucle de modo de interfaz*

```

def encender_foco():
    if ser:
        ser.write(b'F')

def apagar_foco():
    if ser:
        ser.write(b'f')

def encender_ventilador():
    if ser:
        ser.write(b'V')

def apagar_ventilador():
    if ser:
        ser.write(b'v')

```

*Fig. 5 Control manual de foco y ventilador*

Para el uso de los botones se importó las librerías thinker y ttk lo que nos facilita diseños de botones etc., esto para darle un poco de diseño más agradable sin tanta simpleza y dificultad, al igual que nos permite dividir la interfaz en bloques de funciones.

```

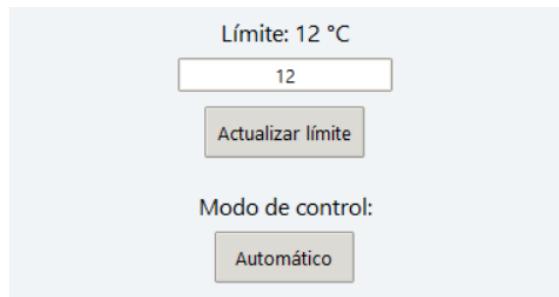
ttk.Button(frame_botones, text="Encender Foco", command=encender_foco).grid(row=0, column=0, padx=10, pady=5)
ttk.Button(frame_botones, text="Apagar Foco", command=apagar_foco).grid(row=0, column=1, padx=10, pady=5)
ttk.Button(frame_botones, text="Encender Ventilador", command=encender_ventilador).grid(row=1, column=0, padx=10, pady=5)
ttk.Button(frame_botones, text="Apagar Ventilador", command=apagar_ventilador).grid(row=1, column=1, padx=10, pady=5)

```

*Fig. 6 Diseño de botones del modo manual de la interfaz.*

### 3.2.2. Interfaz de sistema de control de lazo cerrado

Para la interfaz de lazo cerrado se pensó en llamarse modo automático, es decir, mientras no definamos una temperatura el foco permanecerá encendido, pero en este caso, lo que se implementó fue el poder definir esa temperatura que nos ayudaría a una mejor facilidad de pruebas e implementaciones en diferentes entornos, permitiendo adaptación fácil.



*Fig. 7 Diseño de interfaz de control lazo cerrado.*

Para la programación de la interfaz del lazo cerrado se pensó en primero mandar los datos de la actualización de la temperatura.

```
# Funciones de control
def actualizar_temperatura():
    if simulado:
        temperatura_var.set(f"{random.randint(20, 35)} °C")
    elif ser and ser.in_waiting:
        linea = ser.readline().decode('utf-8').strip()
        if "Temp:" in linea:
            try:
                valor = linea.split("Temp:")[1].strip()
                temperatura_var.set(f"{valor} °C")
            except:
                temperatura_var.set("Error")
    root.after(500, actualizar_temperatura)
```

*Fig. 8 Código de actualización de temperatura.*

Esto nos permite poder tener variaciones sin afectar el sistema o que se confunda al momento de activar o desactivar los componentes. Una vez realizado esta función nos percatamos que, si queremos actualizar el límite que ya por default está, antes de enviar los datos hay que recabarlos esto con la finalidad de que haya una completa seguridad de que se definió un límite nuevo y funcione acorde como se establece.

```
def actualizar_limite():
    nuevo = entrada_limite.get()
    if nuevo.isdigit():
        if ser:
            ser.write(b'L' + nuevo.encode() + b'\n')
    limite_var.set(f"Límite: {nuevo} °C")
```

*Fig. 9 Actualización del límite.*

#### 4. Pruebas

En esta sección se describen las pruebas realizadas para lograr el objetivo del proyecto. Se realizó una prueba de temperatura simple hasta la conexión al circuito, basada en el sistema de control abierto y cerrado de lazo. A continuación, se detallan las pruebas realizadas. Estas pruebas permitieron verificar el funcionamiento del sistema en diferentes condiciones.

#### 4.1. Prueba de sensor con el display de 7 segmentos

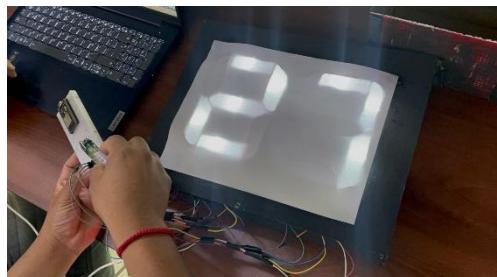


Fig. 10 Prueba de temperatura con display de 7 segmentos con un encendedor.

#### 4.2. Prueba de conexión con los componentes del invernador

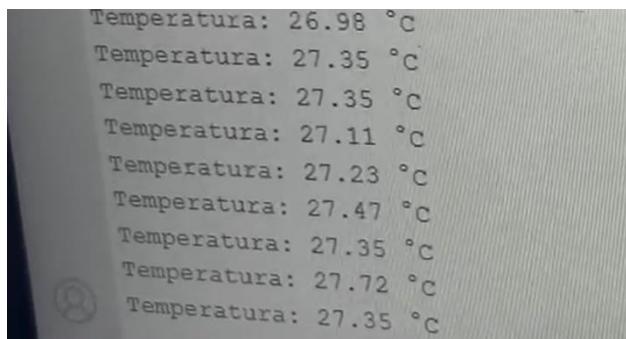


Fig. 11 Actualizacion de temperatura a través de la consola de IDE Arduino.



Fig. 12 Primera prueba de conexión con el reley y un foco led.



*Fig. 13 Primera prueba del circuito completo antes de ser montado a la estructura.*

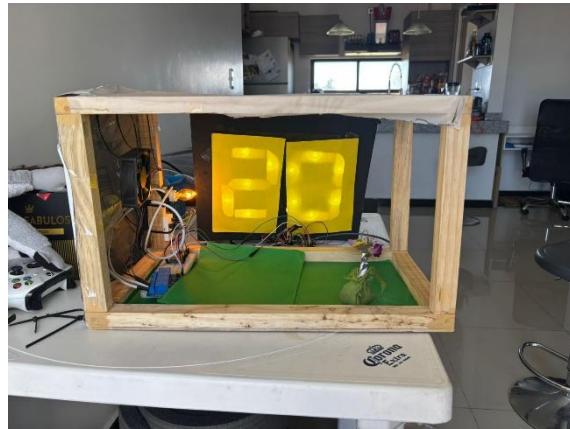
## 5. Resultados

Volvimos a rediseñar el display para lograr que se pueda observar de manera eficaz la temperatura, donde optamos por el color amarillo para una mejor visualización.



*Fig. 14 Rediseño del display de 7 segmentos. . .*

Esta es la estructura ya implementada con el circuito y el sensor al igual que la implementación del display para la temperatura.



*Fig. 15 Sistema de control para el invernadero.*

## 6. Conclusiones

En conclusión, el proyecto demostró la viabilidad de un sistema de control térmico funcional, económico y replicable para un invernadero mediante ESP32, sensor LM35 y un display de siete segmentos heredado de un proyecto previo. Se montó sobre una estructura de madera e incluyó una interfaz gráfica en Python para fijar consignas, visualizar lecturas y operar en lazo abierto o en lazo cerrado. En lazo abierto el accionamiento manual resultó útil para pruebas y calibraciones, pero dependió del operador y fue sensible a perturbaciones. En lazo cerrado el ESP32 comparó la medida del LM35 con la consigna y activó el ventilador de forma automática, manteniendo la temperatura más estable y reduciendo la intervención humana. Las pruebas confirmaron comunicación confiable y tiempos de respuesta adecuados para un prototipo a pequeña escala, con mayor estabilidad y menor variabilidad. Se documentaron decisiones de diseño y oportunidades de mejora como mejor filtrado y calibración del LM35, uso de umbrales adaptativos según condiciones y medición sistemática del desempeño. Como pasos siguientes se propone añadir sensores de humedad del aire, humedad del suelo y luminosidad, habilitar conectividad inalámbrica con telemetría y tableros en tiempo real, incorporar control PID básico o lógicas difusas y analítica predictiva para anticipar eventos, y avanzar hacia autonomía energética con baterías. Con estas mejoras el sistema puede evolucionar hacia una plataforma integral de apoyo a la decisión manteniendo bajo costo, reproducibilidad y facilidad de mantenimiento.

## 7. Referencias

1. Galon, M. L. Q., Tumaliwan, M. V. R., & Sejera, M. M. (2025). Automated Monitoring and Control System of Solar Greenhouse Using ESP32 and Blynk Application. 2024 IEEE 6th Eurasia Conference on IoT, Communication and Engineering, 57. <https://doi.org/10.3390/enproc2025092057>
2. Pacheco Zavala, C., Sofia Hernández Varela, R., Emmanuel Gutiérrez Avelar, A., Mora Martínez, M. A., Robles, N. S., & Tinoco, G. T. (n.d.). VOLUMEN 21 XXVIII Verano De la Ciencia Monitoreo inteligente de invernadero a escala basado en el Internet de las Cosas (IOT). [www.jóvenesenlacienca.ugto.mx](http://www.jóvenesenlacienca.ugto.mx)
3. Rocamora-Osorio, C., Aragon-Rodriguez, F., Codes-Alcaraz, A. M., & Ferrández-Pastor, F.-J. (2025). Automated IoT-Based Monitoring of Industrial Hemp in Greenhouses Using Open-Source Systems and Computer Vision. AgriEngineering, 7(9), 272. <https://doi.org/10.3390/agriengineering7090272>
4. Mecafenix, I. (2023). *Que es un sistema de control de lazo abierto y lazo cerrado.* Ingeniería Mecafenix. <https://www.ingmecafenix.com/automatizacion/control/sistema-de-control/>