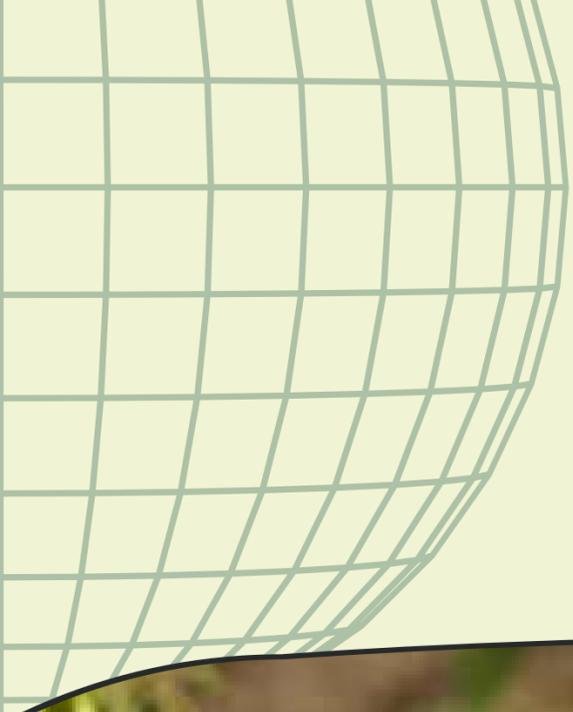


# SMART FARMING

## TIC PRESENTATION

*Growing Tomorrow's  
Harvest Today*

Presented by  
**Ricardo Xavier Paredes**



# INTRODUCCIÓN

## **Título General del TIC :**

CREACIÓN DE UN PROTOTIPO DE MONITOREO DE AGRICULTURA INTELIGENTE UTILIZANDO EQUIPOS DE DESARROLLO DE BAJO COSTO Y COMUNICACIÓN LORAWAN Y WIFI.

## **Título del módulo desarrollado :**

CREACIÓN DE DISPOSITIVOS IOT DE UN PROTOTIPO DE MONITOREO DE AGRICULTURA INTELIGENTE UTILIZANDO EQUIPOS DE BAJO COSTO Y TECNOLOGÍA WIFI.



# GRANJA URKUWAYKU

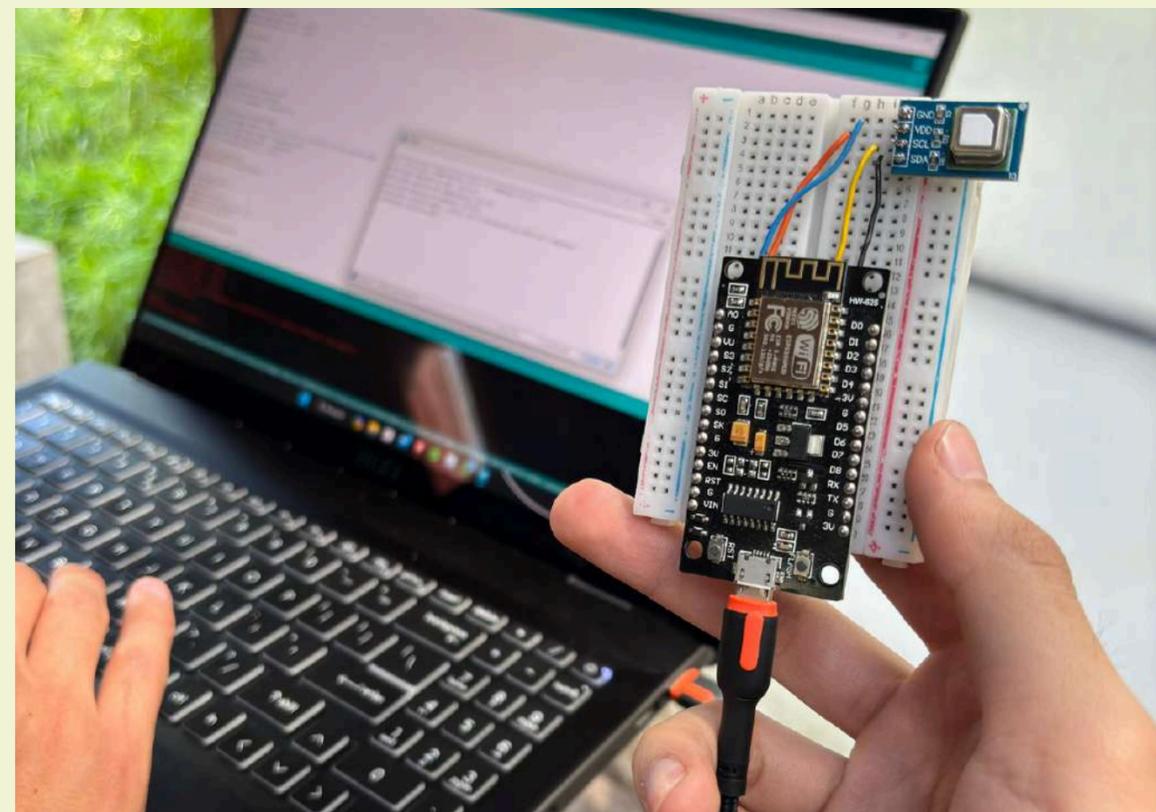


# OBJETIVOS

## OBJETIVOS ESPECÍFICOS

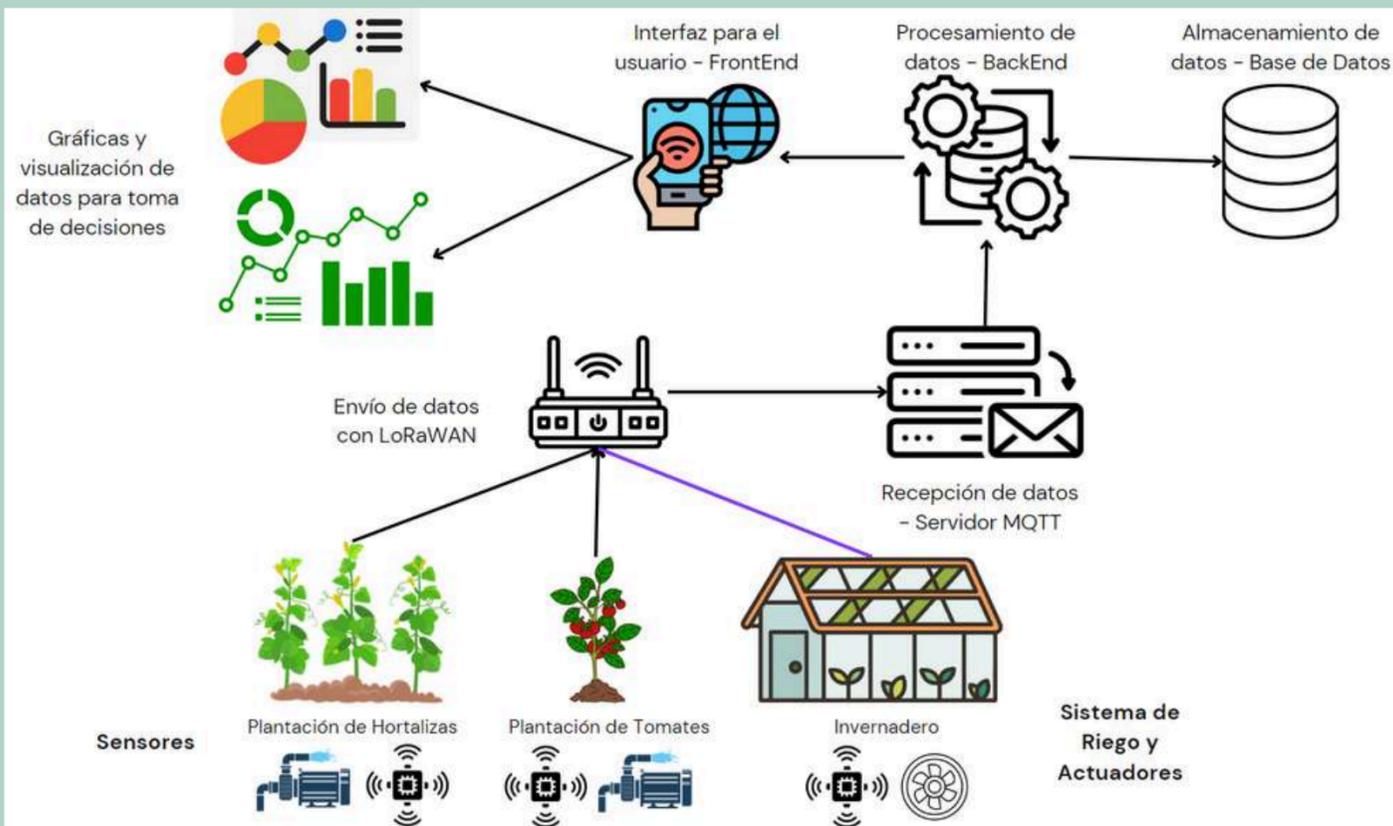
1. Diseñar **la arquitectura del sistema** con sensores y actuadores.
2. Desarrollar el **prototipo físico** integrando sensores y actuadores.
3. Implementar la comunicación **WiFi con servidor MQTT.**
4. Evaluar el sistema en su **hardware y red de comunicación.**

## PRUEBAS DE COBERTURA DEL COMPONENTE

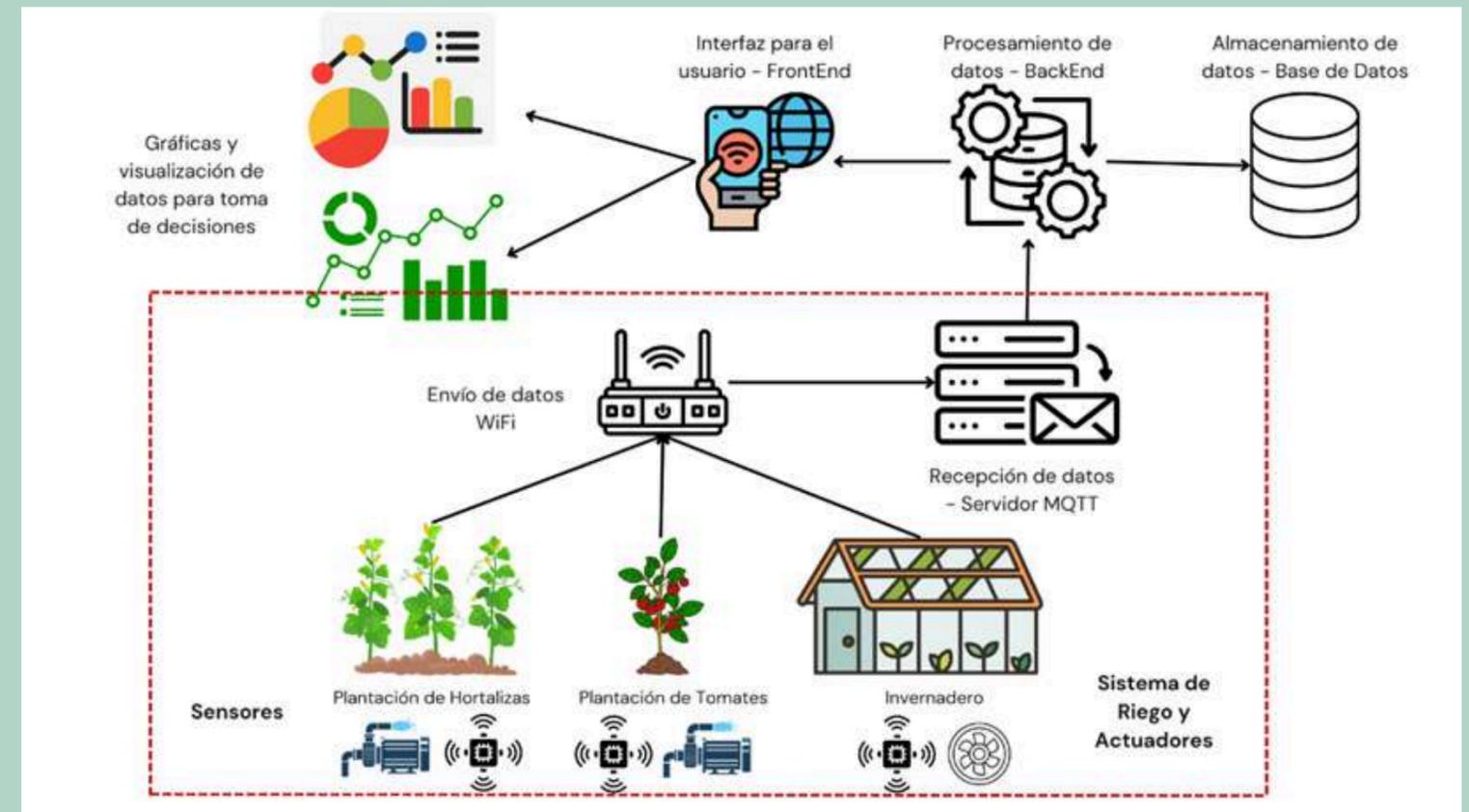


# \* Desarrollo del Componente

## ARQUITECTURA GENERAL



## ARQUITECTURA DEL COMPONENTE

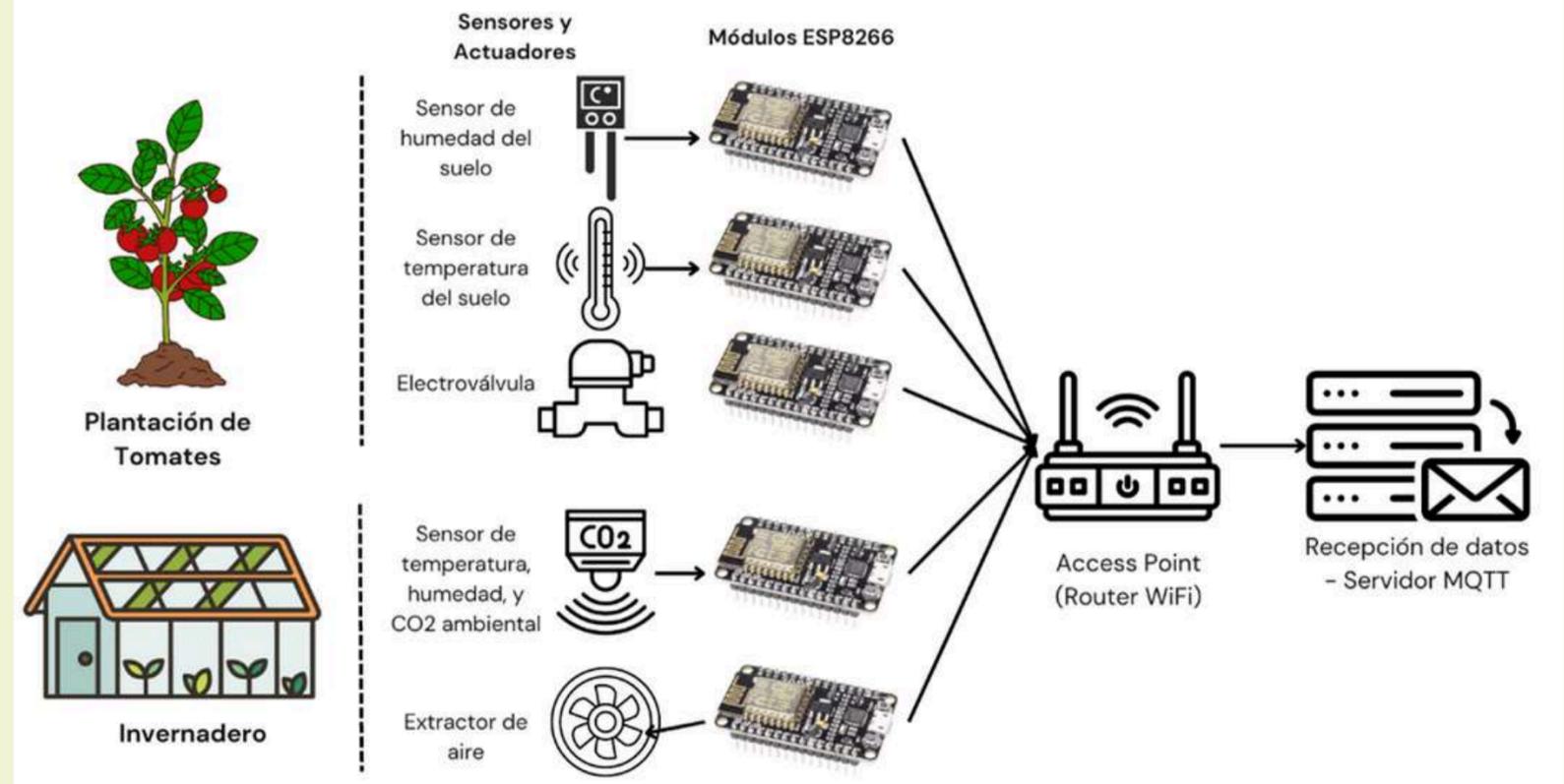


## RESUMEN

- Sistema de Smart Farming de bajo costo utilizando tecnología IoT basada en WiFi.
- Aplicación de la metodología PPDIOO.
- Prototipo probado y optimizado en la Escuela Politécnica Nacional (EPN) antes de su instalación en la finca Urkuwayku.
- Uso del módulo ESP8266 conectado a un servidor MQTT con sensores y actuadores.
- Cobertura de 33,8 metros.
- Consumo energético de 10,21 Wh en 5 horas.
- RSSI promedio de -65,44 dBm.
- Costo por dispositivo menor a 35 USD.

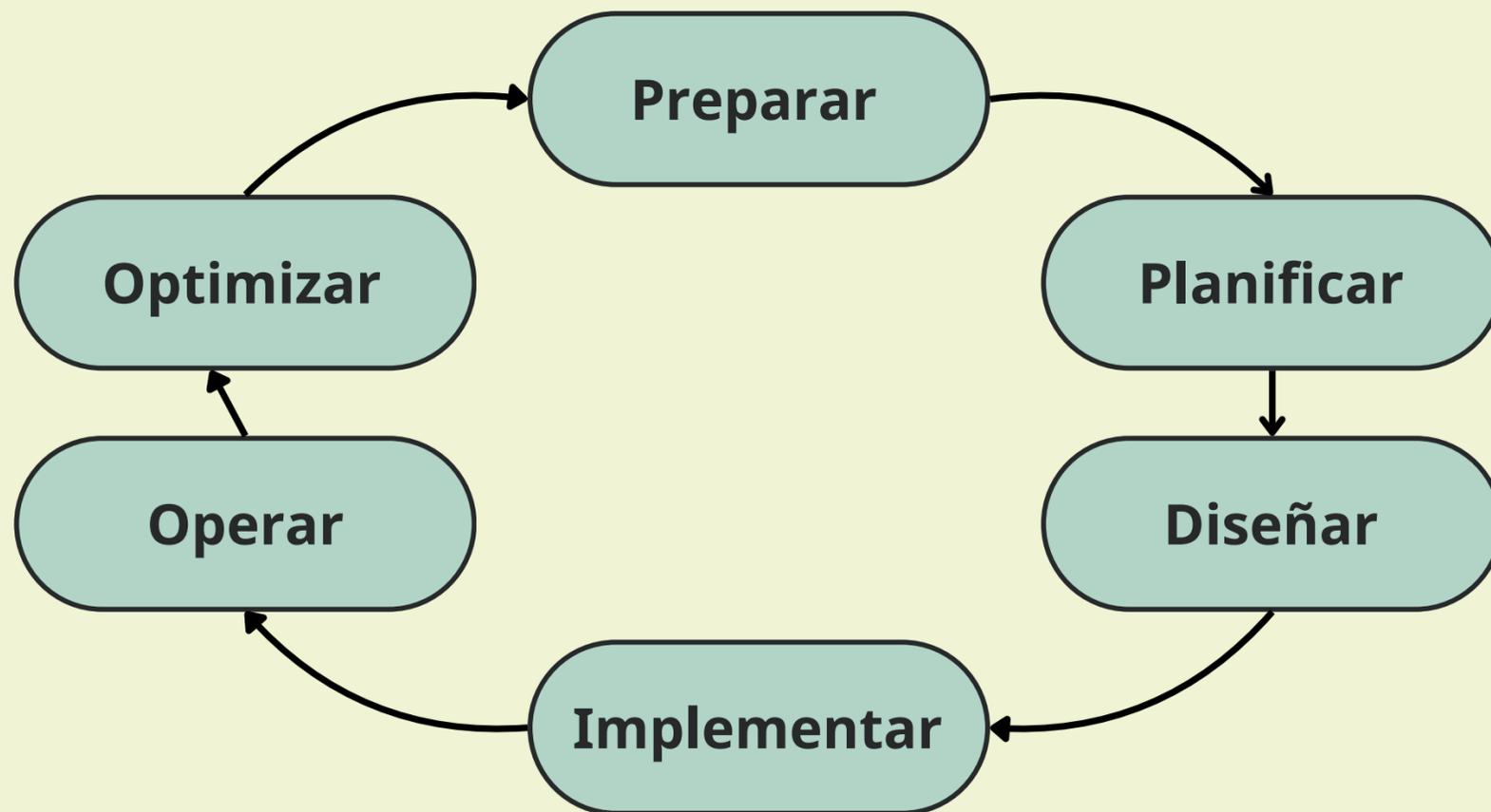
# ¿En qué consiste el componente?

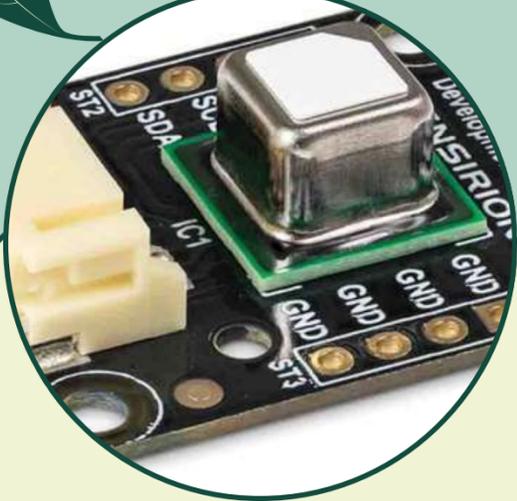
## ARQUITECTURA DEL COMPONENTE



# Metología PPDIOO

La metodología PPDIOO ofrece un **ciclo de vida continuo** que permite la **entrega rápida de soluciones** y la **mejora del prototipo**





**Sensor SCD41  
Sensirion**



**Sensor de HS  
capacitivo V1.2**



**Sensor DS18B20**



**Módulo ESP8266**



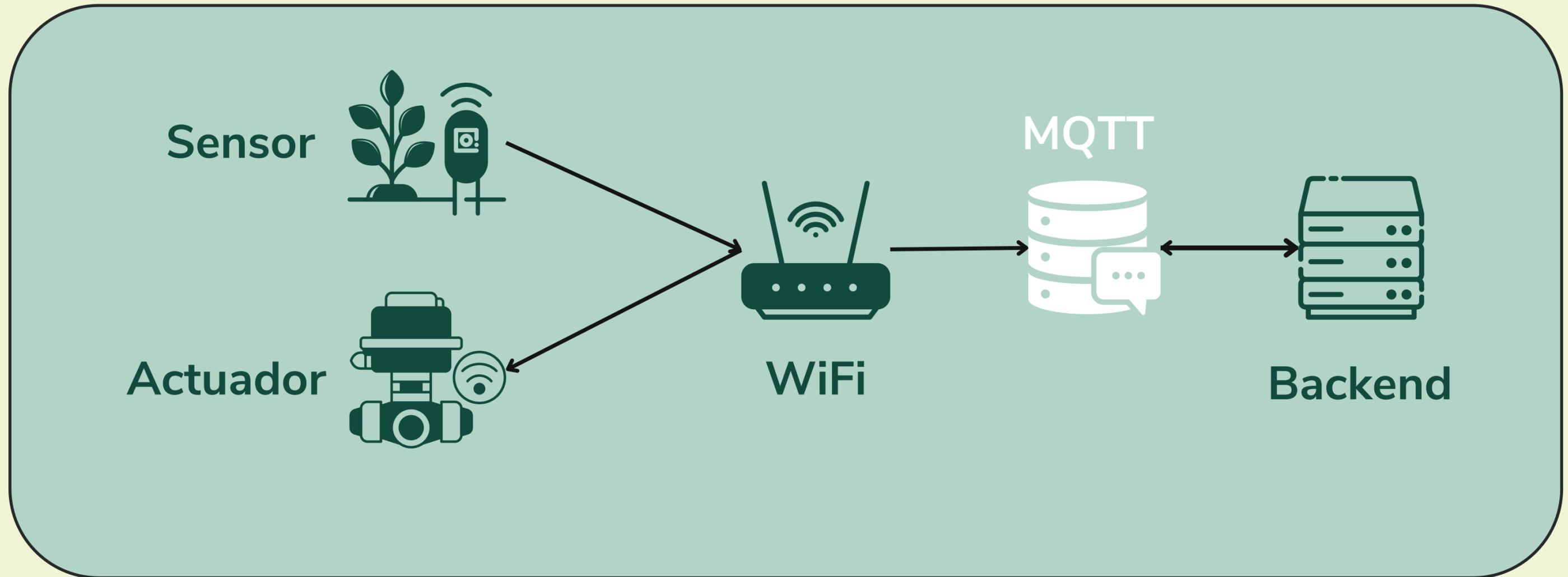
**Relé Electrónico**



**Arduino IDE**

# Equipos y Software Utilizados

# MQTT



# Modelos



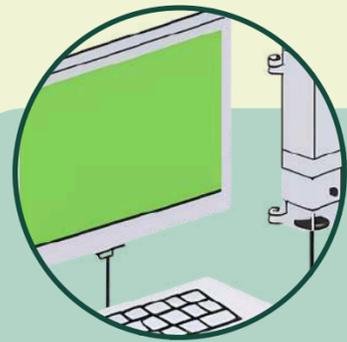
”

**Actuador**



”

**Mensaje**



”

**Regla**

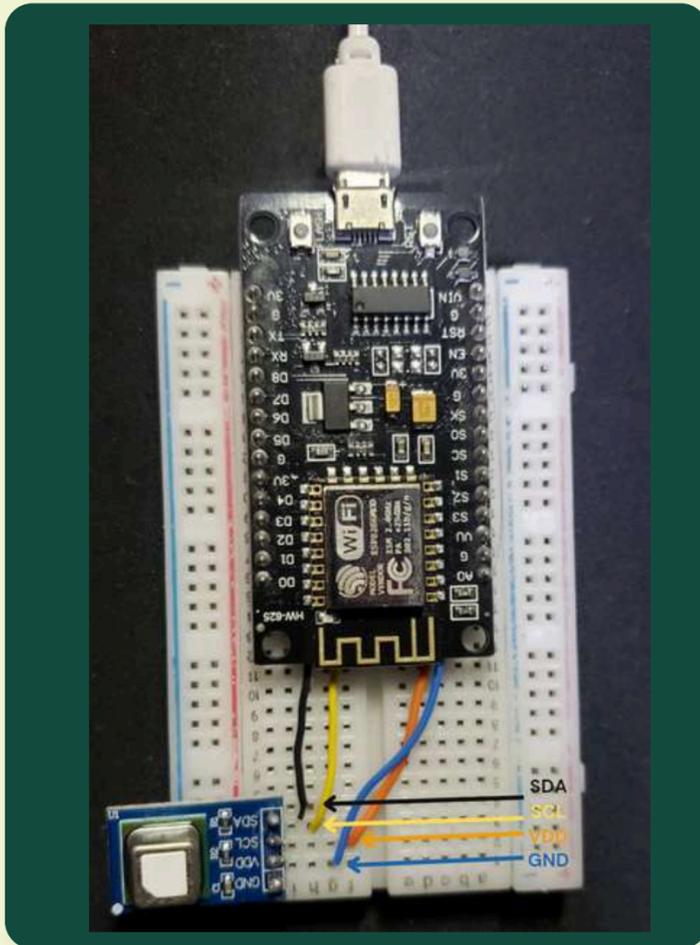


”

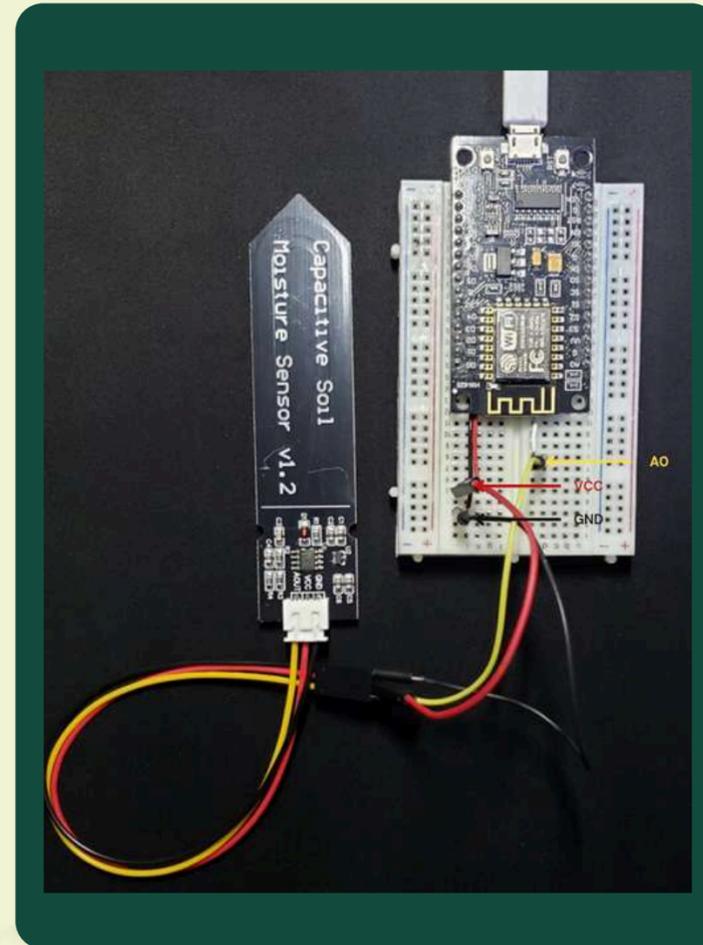
**Sensor**



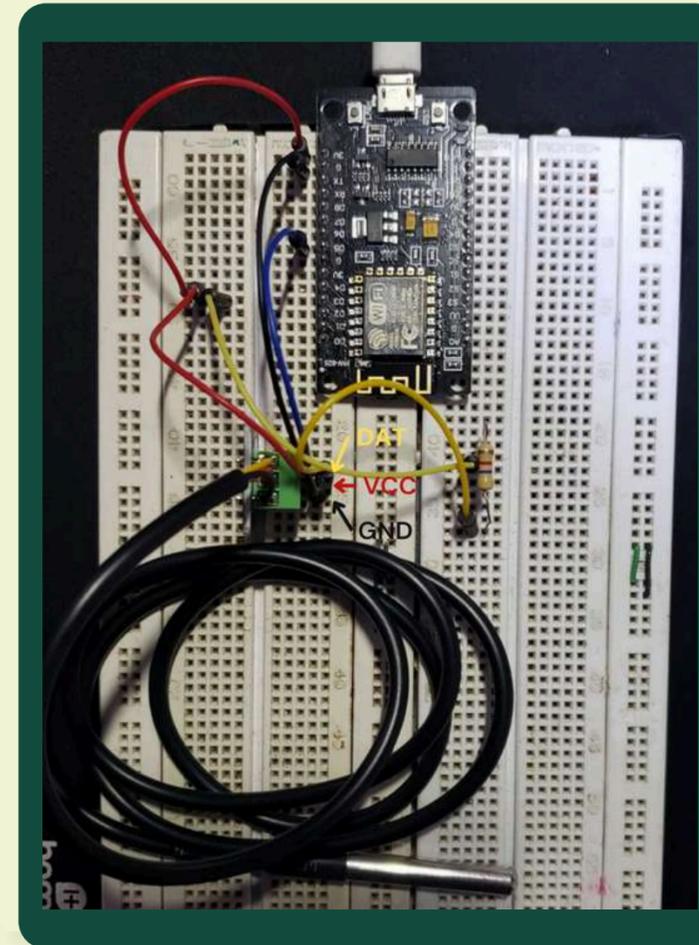
# Implementación



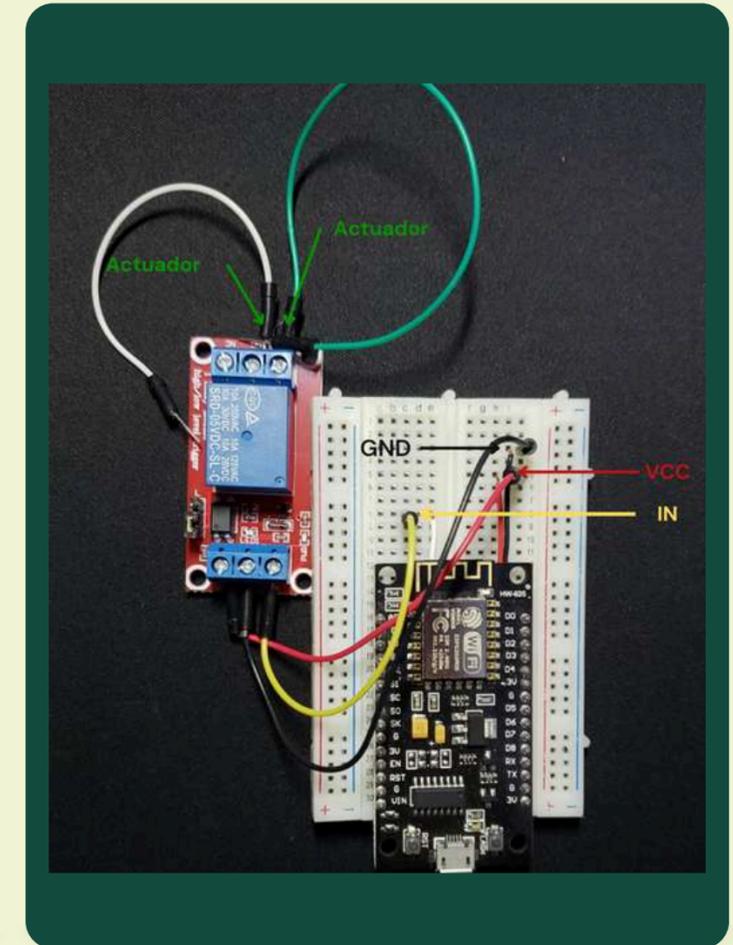
Prototipo del dispositivo IoT del sensor SCD41



Prototipo del dispositivo IoT del sensor analógico capacitivo de HS



Prototipo del Dispositivo IoT del sensor DS18B20



Prototipo del Dispositivo IoT De los Actuadores





# EXTRACTOR DE AIRE

## DATOS RECIBIDOS

```
75 | | desactivarExtractor();
```

Output Serial Monitor x

Message (Enter to send message to 'Generic ESP8266 Module' on 'COM3')

```
Mensaje recibido en tópico [VENTILADORH]: 1
```

```
Extractor de aire activado
```

```
Mensaje recibido en tópico [VENTILADORH]: 1
```

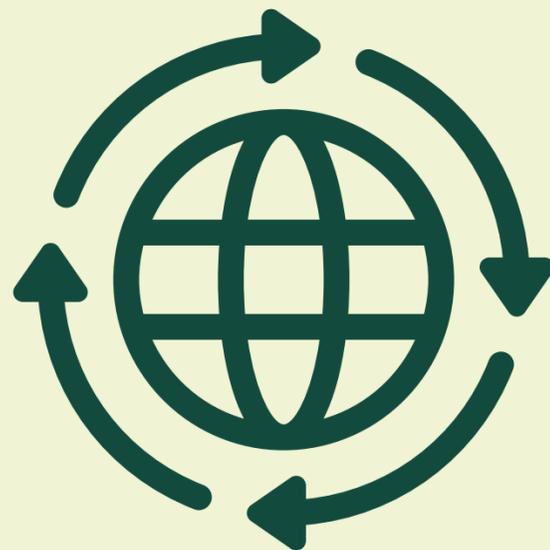
```
Extractor de aire activado
```

```
Intentando conexión MQTT...Falló, rc=-2 Intentando de nuevo en 5 segundos
```

# RESULTADOS

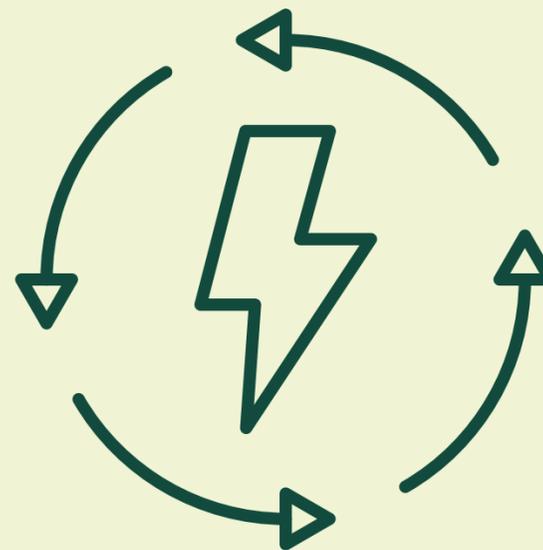
## Cobertura

- Área de Cobertura: **3589.08 [m<sup>2</sup>]**
- Distancia Máxima: **33.8 [m]**



## Consumo Energético

- Tiempo de Evaluación: **5 [h]**
- Consumo Energético: **0.01021 kWh**



## Costos

- Costo por dispositivo **menor a 35 USD**

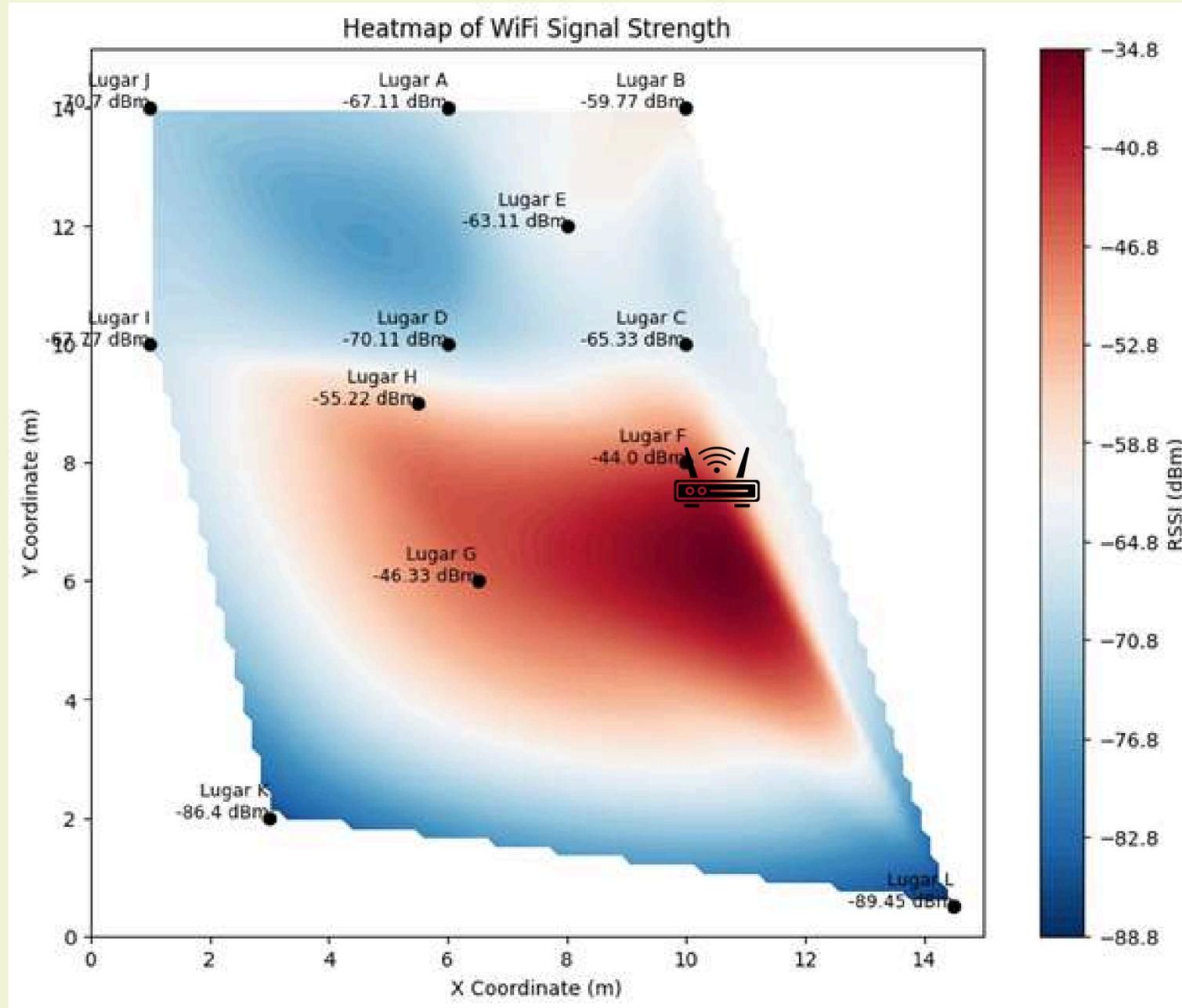


Lugar	Distancia [m]	RSSI [dBm]
Lugar A	8,94	- 67.11
Lugar B	8,3	- 59.77
Lugar C	4,8	- 65.33
Lugar D	6,40	- 70.11
Lugar E	6,32	- 63.11
Lugar F	0,90	- 44.00
Lugar G	4,47	- 46.33
Lugar H	5.65	- 55.22
Lugar I	10,3	- 67.77
Lugar J	12,4	- 70.70
Lugar K	20,35	- 86.40
Lugar L	33,8	- 89.45

# Cobertura

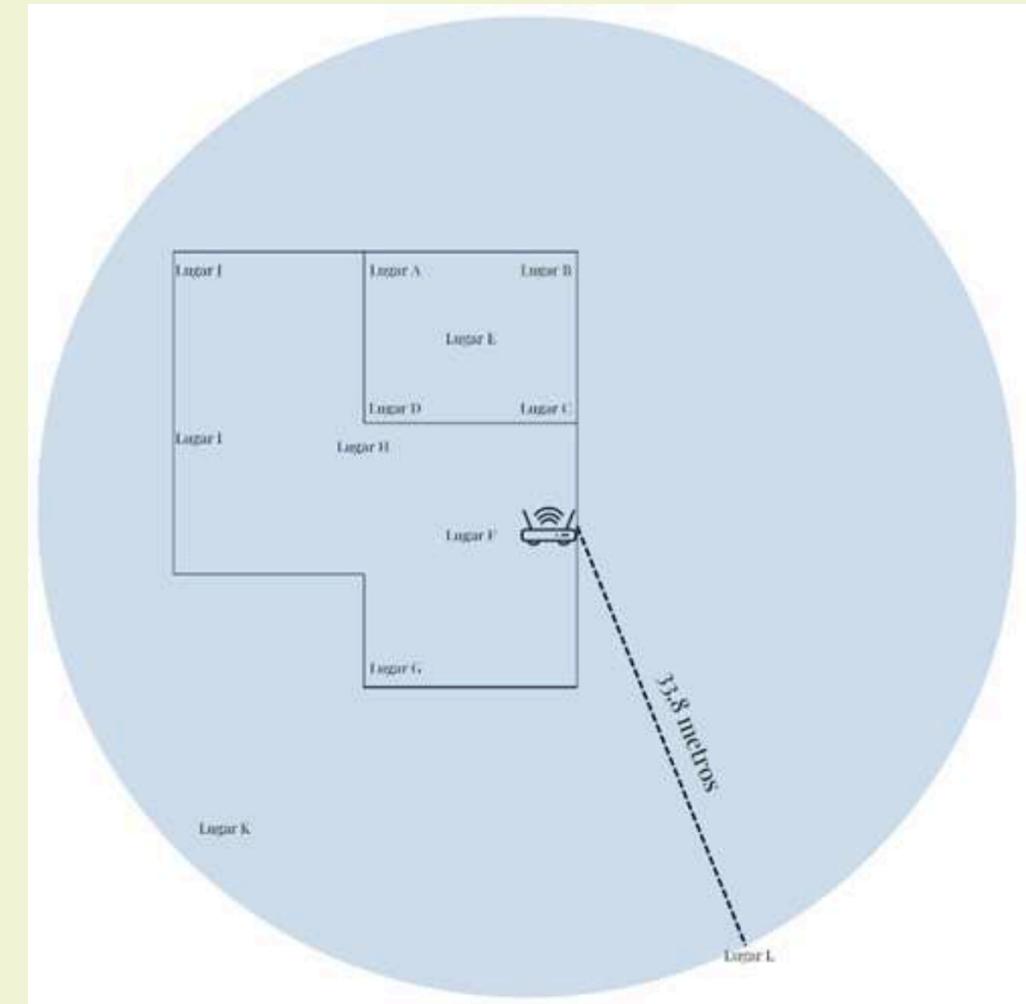
Para lograr calificar la **calidad de la señal**, es necesario clasificarla. Utilizando los niveles de **RSSI**, se tiene que:

- **0 dBm a -50 dBm: Señal excelente**
- **-51 dBm a -60 dBm: Señal muy buena**
- **-61 dBm a -70 dBm: Señal aceptable**
- **-71 dBm a -80 dBm: Señal débil**
- **-81 dBm a -90 dBm: Señal muy débil**



**Distancia Máxima: 33.8 [m]**

# Cobertura



**Área de Cobertura: 3589.08 [m<sup>2</sup>]**



# Consumo Energético

Dispositivo	Intensidad de Corriente [A]	Voltaje [V]	Potencia [W]	Consumo energético unitario [Wh]	Cantidad	Consumo energético [Wh]
Módulo ESP8266	0.079	5,12	0.40448	20.224	5	10.112
SCD41	0,001	3,31	0.00331	0.01655	1	0.01655
Sensor analógico capacitivo de humedad del suelo	0,0025	3,26	0.00815	0.04075	1	0.04075
DS18B20	0,0015	3,33	0.004995	0.024975	1	0.024975
Relé	0,0005	3,28	0.00164	0.0082	2	0.0164
<b>Total</b>						<b>10,211</b>

Tiempo de Evaluación: **5 [h]**

Costo KWh: **\$0.092**

Consumo Energético:  
**0.01021 kWh**

Costo del consumo energético: **\$0.0009189**





# Costos

Costo del prototipo: \$134,16

Dispositivo	Cantidad	Costo Unitario [\$]	Costo total [\$]
ESP8266	5	5	25
Capacitive Soil Moisture Sensor v1.2	1	5.69	5.69
Sensirion SCD41	1	30	30
Sensor DS18B20	1	7.99	7.99
Relé	2	3.99	7.98
Protoboard	5	5.90	29.50
Cable de carga Micro USB	5	5.60	28.00
<b>Total</b>			<b>134,16</b>

# CONCLUSIONES



①

**Diseñar la arquitectura del sistema con sensores y actuadores.**

- Se logró diseñar y estructurar una arquitectura de Smart Farming con dispositivos **IoT** y cobertura **WiFi** de aproximadamente **3589 [m<sup>2</sup>]**.

②

**Desarrollar el prototipo físico integrando sensores y actuadores.**

- Se desarrolló el **prototipo físico** y se implementó una infraestructura de comunicación **WiFi** con acceso a datos mediante un servidor **MQTT**.
- Los dispositivos **IoT** se integraron de manera funcional, pero no se pudo probar la electroválvula debido a retrasos en la entrega.

# CONCLUSIONES



- 3 Implementar la comunicación WiFi con servidor MQTT.**
  - Las pruebas del sistema mostraron **buena cobertura y calidad de señal.**
  - La calidad de la señal **WiFi** fue generalmente **buena o aceptable** para el funcionamiento del sistema.
- 4 Evaluar el sistema en su hardware y red de comunicación.**
  - El mapa de calor mostró variaciones en la **intensidad de la señal**, con rangos de -44.00 [dBm] a -89.45 [dBm].
  - El sistema consume **10.2106 [Wh]** en 5 horas, con el ESP8266 siendo el mayor consumidor de energía.

# RECOMENDACIONES



## Mejora de Cobertura

**Instalar** repetidores o **reubicar** el router para **mejorar la señal** en áreas débiles.

## Optimización Energética

Explorar opciones para **reducir** aún más el **consumo energético** del sistema.

## Expansión del Sistema

**Ampliar el sistema** con más sensores y actuadores e integrar nuevas tecnologías.

## Protección Física

Diseñar el dispositivo para resistir **condiciones externas** en pruebas al aire libre.



11/09/2024

 0960546408  
 Runita de corazon  
 Runita\_de\_shungo

***Como dijo mi ex, hasta aquí llegamos, muchas gracias.***

A top-down view of fresh vegetables in a black wire basket. The basket contains several types of lettuce, including light green leaf lettuce and dark red leaf lettuce, along with a bunch of fresh spinach. Numerous bright orange carrots with their green tops are also visible. The text '¿PREGUNTAS?' is overlaid in the center in a bold, white, sans-serif font, enclosed within a thin white rectangular border.

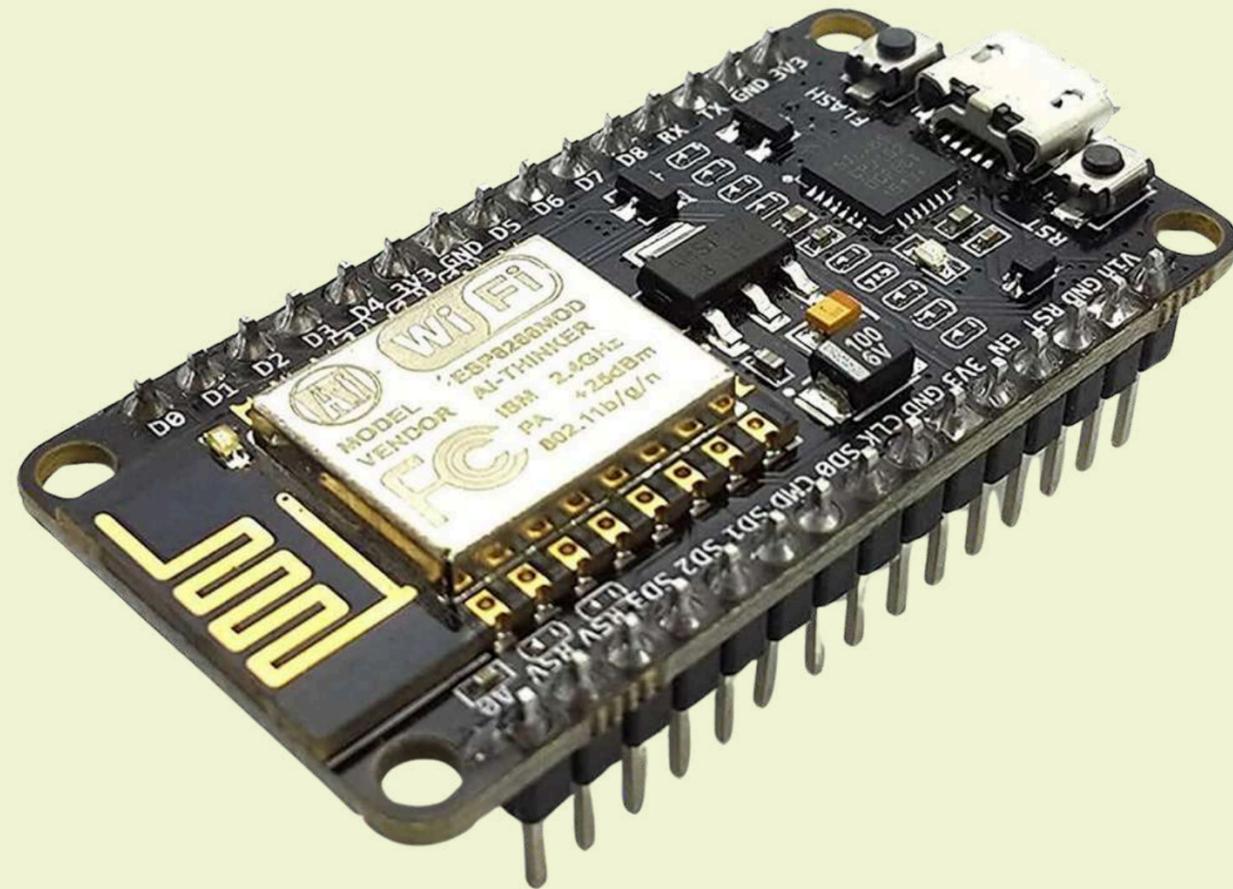
**¿PREGUNTAS?**

# ¿Por qué utilicé la metodología PPDIOO?



- La metodología PPDIOO es perfecta para proyectos de redes como el Smart Farming, ya que tiene una **estructura clara y fases definidas**.
- Facilita la **optimización del rendimiento y la cobertura**, algo esencial para un sistema que busca **maximizar la eficiencia**.
- La fase de operación y **optimización** asegura que el sistema sea **escalable** y sostenible a largo plazo.

# ¿Por qué utilicé ESP8266?



- El ESP8266 tiene **WiFi integrado**, lo que simplifica la comunicación sin necesidad de módulos adicionales.
- Es una opción **económica y eficiente** en términos de energía, ajustándose al presupuesto y las necesidades del proyecto.
- Tiene el rendimiento suficiente para las tareas del sistema, mientras que **consume menos energía** que placas como la Raspberry Pi, que tienen mayores capacidades pero no son necesarias para este proyecto.

# ¿Por qué utilicé Mosquitto como servidor MQTT?



- Es **liviano y eficiente**, ideal para dispositivos IoT que manejan recursos limitados, como el **ESP8266**.
- El protocolo MQTT está **orientado** específicamente para la comunicación **M2M** (Machine to Machine), mientras que Kafka está orientado a grandes volúmenes de datos, lo que sería innecesario y complejo para este proyecto.
- Es **fácil de configurar y consume pocos recursos**, lo que lo hace perfecto para un sistema con restricciones de hardware.

