

# Planificación del Proyecto

---

Última modificación: 27/01/25

## Semana 1: Configuración inicial y prueba de sensores

### Objetivos:

1. Configurar el ESP32 y asegurarte de que los sensores (**DS18B20** y **MQ-135**) funcionan correctamente.
2. Implementar la lectura de datos de los sensores.
3. Configurar el almacenamiento local básico (usando la memoria flash del ESP32 o SPIFFS).

### Tareas:

1. Configurar el entorno de desarrollo:
  - Instalar Arduino IDE / PlatformIO.
  - Descargar librerías necesarias (DallasTemperature, OneWire, etc.).
2. Conectar y probar los sensores:
  - Configurar el DS18B20 (leer temperatura).
  - Configurar el MQ-135 (medir niveles de CO<sub>2</sub>).
  - Mostrar las lecturas en el monitor serie.
3. Implementar almacenamiento local:
  - Guardar lecturas de temperatura y CO<sub>2</sub> en un archivo en la memoria flash del ESP32 (modo offline).

### Entregable al final de la semana:

- ESP32 leyendo datos de ambos sensores.
  - Datos almacenados localmente en el ESP32.
- 

## Semana 2: Control de actuadores y lógica de umbrales

### Objetivos:

1. Conectar y controlar los actuadores (ventilador y servo de la compuerta).
2. Implementar la lógica de control basada en los umbrales de temperatura y CO<sub>2</sub>.
3. Diseñar la estructura básica del programa (lectura, lógica, control, almacenamiento).

### **Tareas:**

1. Conectar los actuadores:
  - Configurar el ventilador (mediante transistor o módulo relé).
  - Configurar el servo para abrir/cerrar la compuerta.
2. Implementar la lógica de control:
  - Encender el ventilador si la temperatura supera los 8 °C.
  - Abrir la compuerta con el servo si el CO<sub>2</sub> supera los 10,000 ppm.
3. Refinar el almacenamiento:
  - Registrar las acciones de los actuadores junto con las lecturas de los sensores.

### **Entregable al final de la semana:**

- Actuadores funcionando según los valores de los sensores.
  - Lógica de control completamente implementada.
- 

## **Semana 3: Comunicación MQTT y plataforma IoT**

### **Objetivos:**

1. Configurar la comunicación con la plataforma IoT usando MQTT.
2. Crear un dashboard básico para visualizar los datos y recibir alertas.

### **Tareas:**

1. Implementar MQTT:
  - Configurar conexión Wi-Fi del ESP32.
  - Publicar lecturas de sensores en un broker MQTT (como HiveMQ o Mosquitto).
2. Diseñar el dashboard IoT:
  - Usar herramientas como Node-RED, Adafruit IO o Blynk.
  - Mostrar datos en tiempo real (temperatura, CO<sub>2</sub>, estado de actuadores).
  - Configurar alertas (por ejemplo, notificaciones push o correo electrónico).
3. Probar la sincronización:
  - Enviar datos locales almacenados al dashboard una vez que se establezca la conexión (modo offline a online).

### **Entregable al final de la semana:**

- Datos enviados al dashboard IoT en tiempo real.
  - Dashboard funcional con visualización y alertas básicas.
-

## Semana 4: Integración y pruebas finales

### Objetivos:

1. Integrar todas las partes del sistema (sensores, actuadores, almacenamiento, comunicación).
2. Realizar pruebas completas en una maqueta o entorno simulado.
3. Documentar el sistema.

### Tareas:

1. Pruebas de integración:
  - Validar que los sensores, actuadores y comunicación MQTT funcionen juntos.
  - Simular escenarios críticos (por ejemplo, temperatura alta, niveles altos de CO<sub>2</sub>).
2. Optimización:
  - Mejorar la eficiencia energética (usando modos de bajo consumo del ESP32).
  - Refinar la lógica de control y manejo de datos.
3. Documentación:
  - Crear diagramas eléctricos y de flujo del software.
  - Documentar cómo instalar y operar el sistema.

### Entregable al final de la semana:

- Prototipo funcional con todas las características implementadas.
  - Documentación básica del sistema.
- 

## Resumen de Entregables Semanales

Semana	Entregables Principales
1	Sensores configurados, datos almacenados localmente.
2	Actuadores funcionando según lógica de umbrales.
3	Comunicación MQTT y dashboard IoT básico.
4	Prototipo integrado, probado y documentado.