

**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**  
**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**  
**ESCUELA PROFESIONAL DE INGENIERÍA DE SOFTWARE**



**ASIGNATURA:**

**INTERNET DE LAS COSAS**

**DOCENTE:**

**YESSICA ROSAS CUEVA**

**GRUPO 01:**

- Villacis Alvear David
- Tomayquispe Ramos Jorge Luis
- Carlos Augusto Wong Gómez
- Kevin Anderson Gonzalez Cabezas
- Quiroz Pita, Alexander Andrey

**2025**

## Índice

<b>1. Introducción.....</b>	<b>3</b>
<b>2.Problemática.....</b>	<b>3</b>
<b>3. Objetivos.....</b>	<b>3</b>
<b>4. Materiales y Componentes.....</b>	<b>3</b>
<b>5. Metodología.....</b>	<b>4</b>
5.1 Diseño del Sistema.....	4
5.2 Comunicación de Datos.....	4
5.3 Desarrollo de Software.....	4
<b>6. Resultados Esperados.....</b>	<b>4</b>
<b>8. Conclusión.....</b>	<b>4</b>
<b>9. Simulación Con un SENSOR.....</b>	<b>5</b>

## 1. Introducción

En este proyecto, se propone el desarrollo de un sistema basado en ESP32 para la medición y monitoreo en tiempo real de la calidad del aire y la temperatura, permitiendo la recolección de datos para su análisis y toma de decisiones.

## 2. Problemática

La contaminación del aire y las variaciones de temperatura representan un desafío significativo para la salud humana y el medio ambiente. La exposición prolongada a gases contaminantes puede provocar enfermedades respiratorias y afectar la calidad de vida. Además, los cambios bruscos de temperatura influyen en el bienestar de las personas. Sin embargo, la falta de sistemas accesibles y eficientes para el monitoreo en tiempo real dificulta la toma de decisiones informadas y la implementación de medidas preventivas.

## 3. Objetivos

- Implementar un sistema de monitoreo ambiental con sensores de calidad del aire y temperatura utilizando ESP32.
- Desarrollar una interfaz de visualización de datos en tiempo real mediante una aplicación web o móvil.
- Integrar conectividad Wi-Fi/Bluetooth para el envío de datos a una plataforma en la nube. MQTT.

## 4. Materiales y Componentes

- **Microcontrolador:** ESP32.
- **Sensor de Calidad del Aire:** MQ-135 o CCS811.
- **Sensor de Temperatura y Humedad:** DHT22 o BME280.
- **Pantalla opcional:** OLED SSD1306 para visualización local de datos y SERVO.
- **Módulo de conexión:** Wi-Fi y MQTT para envío de datos.
- **Fuente de alimentación:** Batería Li-ion con módulo de carga TP4056 (opcional).

## 5. Metodología

### 5.1 Diseño del Sistema

Se utilizará el ESP32 como unidad central de procesamiento, que capturará datos de los sensores y los transmitirá a una plataforma de monitoreo.

### 5.2 Comunicación de Datos

- **Wi-Fi/MQTT:** Para enviar datos a una plataforma en la nube como Thingspeak, Firebase o un servidor local.

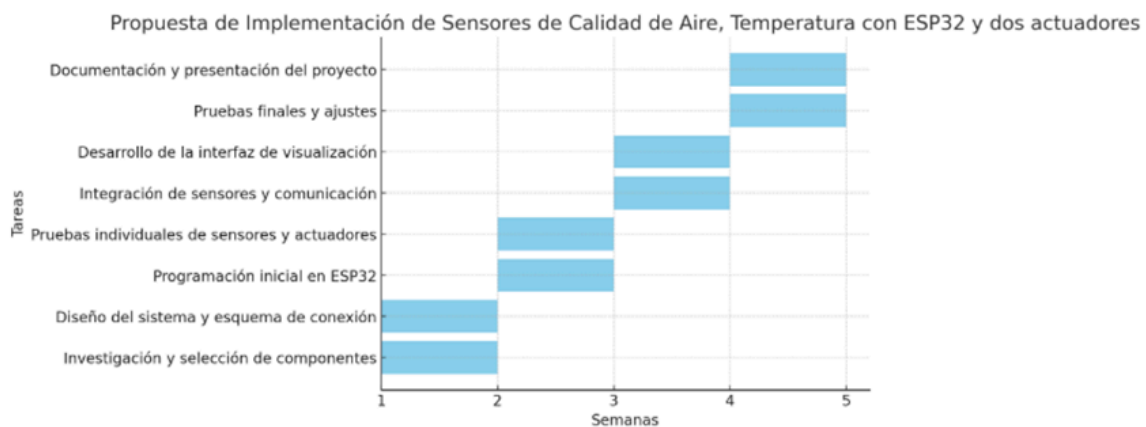
### 5.3 Desarrollo de Software

- Programación en **Arduino IDE** o **MicroPython**.
- Uso de bibliotecas para la integración de sensores y comunicación con la nube.
- Diseño de una interfaz gráfica para la visualización de datos en tiempo real.

### 6. Resultados Esperados

- Sistema funcional capaz de medir y visualizar la calidad del aire y temperatura en tiempo real.
- Plataforma de monitoreo accesible desde computadoras.
- Posibilidad de expandir el proyecto con alertas por umbrales de contaminación.
- Predicción de datos, se podrá trabajar con análisis de otro país.

### 7. Cronograma para la implementación



### 8. Conclusión

Este sistema proporcionará información útil sobre las condiciones ambientales en distintos entornos, lo que permitirá tomar medidas preventivas para mejorar la calidad del aire y monitorear variaciones de temperatura de manera eficiente

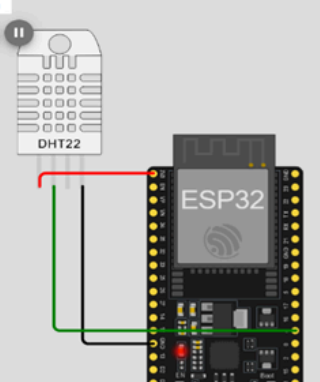
## 9. Simulación Con un SENSOR

WOKWI [SAVE](#) [SHARE](#) Docs [SIGN UP](#)

sketch.ino [diagram.json](#) [libraries.txt](#) Library Manager

```
1 #include "DHT.h"
2
3 // Configuración del pin y el tipo de sensor
4 #define DHTPIN 4 // Pin al que está conectado el sensor (GP104)
5 #define DHTTYPE DHT22 // Modelo del sensor: DHT22 (o DHT11 si cambias el sensor)
6
7 DHT dht(DHTPIN, DHTTYPE);
8
9 void setup() {
10   Serial.begin(115200); // Inicializa la comunicación serie
11   dht.begin(); // Inicializa el sensor DHT
12   Serial.println("DHT22 Sensor iniciado");
13 }
14
15 void loop() {
16   // Lee la temperatura y la humedad
17   float humedad = dht.readHumidity();
18   float temperatura = dht.readTemperature();
19
20   // Verifica si la lectura es válida
21   if (!isnan(humedad) || !isnan(temperatura)) {
22     Serial.println("Error al leer el sensor DHT!");
23     return;
24   }
25
26   // Muestra los valores en el monitor serie
27   Serial.print("Humedad: ");
28   Serial.print(humedad);
29   Serial.print("% Temperatura: ");
30   Serial.print(temperatura);
31   Serial.println("°C");
32
33   // Espera 2 segundos antes de la próxima lectura
34   delay(2000);
35 }
36
```

Simulation



load:0x40080400,len:2972  
entry 0x400805dc  
DHT22 Sensor iniciado  
Humedad: 40.00% Temperatura: 24.00°C  
Humedad: 40.00% Temperatura: 24.00°C  
Humedad: 40.00% Temperatura: 24.00°C  
Humedad: 40.00% Temperatura: 24.00°C  
Humedad: 40.00% Temperatura: 24.00°C