

# **Documentación del Código: Control de Dispositivos y Lectura de Sensores con MQTT y WiFi**

**Maiker Stanley Cañon – u20202193179**

**Daniel Santiago Vargas Monje – u20202191528**

**Profesor:  
Juan Castro**

**UNIVERSIDAD SURCOLOMBIANA  
NEIVA-HUILA**

**01 junio del 2025**

## Explicación archivo - sketch\_connect\_mqtt

Este script está diseñado para controlar dispositivos (como un ventilador y una bombilla) y leer datos de temperatura y humedad desde un sensor DHT22 utilizando un ESP32. El ESP32 se conecta a una red WiFi y se comunica con un servidor MQTT para recibir comandos para encender y apagar dispositivos, así como enviar datos de sensores.

### Librerías Utilizadas

- **WiFi.h:** Biblioteca para la conexión WiFi con el ESP32.
- **PubSubClient.h:** Biblioteca para la implementación de MQTT en el ESP32.
- **Wire.h:** Biblioteca de comunicación I2C.
- **Adafruit\_Sensor.h:** Biblioteca base de Adafruit para la interfaz con los sensores.
- **DHT.h / DHT\_U.h:** Bibliotecas específicas para el sensor DHT22 que permiten leer los valores de temperatura y humedad.

### Definición de Variables

- **ledPin1 y ledPin2:** Definen los pines de los LEDs que controlarán dispositivos (ventilador y bombilla, respectivamente).
- **DHTPIN:** Define el pin digital donde está conectado el sensor DHT22.
- **DHTTYPE:** Especifica el tipo de sensor DHT (en este caso, DHT22).
- **dht:** Instancia del objeto DHT\_Unified, que se utiliza para leer los datos del sensor DHT.
- **ssid y password:** Datos de conexión WiFi.
- **mqtt\_server:** Dirección IP del servidor MQTT donde el ESP32 se conectará.
- **espClient y client:** Definen el cliente MQTT que utilizará el ESP32 para enviar y recibir mensajes.
- **temperature y humidity:** Variables para almacenar los valores de temperatura y humedad leídos del sensor DHT22.

### Función setup()

La función setup() se ejecuta una sola vez al encender el dispositivo. Esta función inicializa la comunicación serie, el sensor DHT, establece la conexión WiFi, configura el servidor MQTT y los pines de salida para controlar los dispositivos (LEDs).

1. Se inicializa el sensor DHT22 con dht.begin().

2. Se establece la conexión WiFi con `setup_wifi()`.
3. Se configura la conexión con el servidor MQTT mediante `client.setServer()` y se asigna un "callback" con `client.setCallback()`.
4. Se configuran los pines de los LEDs para ser salidas con `pinMode()`.

### **Función `setup_wifi()`**

Esta función se encarga de conectar el ESP32 a una red WiFi.

1. Inicia la conexión WiFi utilizando el SSID y la contraseña proporcionados.
2. La función espera hasta que la conexión WiFi sea exitosa y luego imprime la dirección IP asignada.

### **Función `callback()`**

La función `callback()` es llamada cada vez que el ESP32 recibe un mensaje MQTT en los temas suscritos. Dependiendo del mensaje recibido, se toman acciones específicas:

1. Si el mensaje es `{'bombilla':'ON'}`, se enciende el LED conectado al pin `ledPin1`.
2. Si el mensaje es `{'bombilla':'OFF'}`, se apaga el LED conectado al pin `ledPin1`.
3. Si el mensaje es `{'ventilador':'ON'}`, se enciende el LED conectado al pin `ledPin2`.
4. Si el mensaje es `{'ventilador':'OFF'}`, se apaga el LED conectado al pin `ledPin2`.
5. Si el mensaje recibido es "on" o "off" en el tema `python/mqtt`, se controla el encendido/apagado del LED correspondiente.

### **Función `reconnect()`**

La función `reconnect()` intenta reconectar el ESP32 al servidor MQTT si la conexión se pierde. Si la reconexión es exitosa, se suscribe al tema `python/mqtt`. Si la reconexión falla, la función espera 5 segundos y vuelve a intentar.

### **Función `loop()`**

La función `loop()` se ejecuta repetidamente. Realiza las siguientes acciones:

1. Verifica si el cliente MQTT está conectado. Si no está conectado, llama a `reconnect()`.
2. Si han pasado más de 5 segundos desde el último mensaje (`millis()`), lee los datos de temperatura y humedad del sensor DHT22.
  - Si la lectura es exitosa, los valores de temperatura y humedad se almacenan en las variables correspondientes.
  - Si los datos son válidos (no son -999), se crea un mensaje JSON con los datos de temperatura y humedad y se publica en el tema `python/mqtt`.

- Si la lectura falla, se muestra un mensaje de error en el monitor serie y se omite el envío del mensaje MQTT.

### **Resumen de Funcionamiento**

1. **Conexión a WiFi:** El ESP32 se conecta a la red WiFi utilizando los datos proporcionados (SSID y contraseña).
2. **Conexión a MQTT:** Una vez conectado a WiFi, el ESP32 se conecta al servidor MQTT especificado. Si la conexión MQTT se pierde, se intenta reconectar.
3. **Control de dispositivos:** Los dispositivos (bombilla y ventilador) se controlan a través de mensajes MQTT. Los mensajes pueden ser enviados desde otro dispositivo o servidor, y en función de los mensajes, se encienden o apagan los dispositivos conectados al ESP32.
4. **Lectura de sensores:** Cada 5 segundos, el ESP32 lee los datos del sensor DHT22 y publica los valores de temperatura y humedad en el servidor MQTT. Estos datos están disponibles para ser consumidos por otros dispositivos o aplicaciones suscritos al tema `python/mqtt`.