

# Manual de Pruebas: Conexión de ESP32 a una API

## Introducción

Este documento describe el procedimiento para conectar un ESP32 a una API RESTful, utilizando el módulo WiFi para enviar datos de sensores en formato JSON a un servidor. Se utilizará un ejemplo básico que envía datos de temperatura y humedad.

## Requisitos Previos

### Hardware Necesario

- Placa ESP32 DevKit v1.
- Cable USB para conectar el ESP32 a un ordenador.
- Sensor de temperatura y humedad (opcional, se pueden simular los datos).

### Software Necesario

- Arduino IDE (o PlatformIO).
- Bibliotecas de ESP32 y WiFi.
- Acceso a la API RESTful (en este caso, [https://api.gonaiot.com/plata/datos\\_dispositivos](https://api.gonaiot.com/plata/datos_dispositivos)).

## Configuración del Entorno

1. **Instalación del Arduino IDE:**
  - Descarga e instala el Arduino IDE desde aquí.
2. **Configuración del ESP32 en Arduino IDE:**
  - Abre el Arduino IDE.
  - Ve a **Archivo > Preferencias** y en "Gestor de URLs adicionales de tarjetas", agrega:  
  

```
arduino  
Copiar código  
https://dl.espressif.com/dl/package\_esp32\_index.json
```
  - Luego, ve a **Herramientas > Placa > Gestor de placas**, busca "ESP32" e instala el paquete correspondiente.
3. **Instalación de Bibliotecas:**
  - Asegúrate de que las bibliotecas `WiFi` y `HTTPClient` estén disponibles (vienen preinstaladas con el soporte de ESP32).

# Configuración del Código

## Código de Ejemplo

```
cpp
Copiar código
#include <WiFi.h>
#include <HTTPClient.h>

// Definir las credenciales Wi-Fi
const char* ssid = "tu_SSID"; // Reemplaza con tu SSID
const char* password = "tu_PASSWORD"; // Reemplaza con tu contraseña

// URL del servidor donde se envían los datos
const char* serverName =
"https://api.gonaiot.com/plata/datos_dispositivos";

// Variables de prueba (simulando datos de sensores)
float temperatura = 24.5; // Ejemplo de valor de temperatura
float humedad = 60.2;      // Ejemplo de valor de humedad

void setup() {
    Serial.begin(115200);
    WiFi.begin(ssid, password);
    Serial.println("Conectando a WiFi...");

    while (WiFi.status() != WL_CONNECTED) {
        delay(1000);
        Serial.println("Intentando conectarse...");
    }

    Serial.println("Conectado a la red WiFi");
}

void loop() {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        http.begin(serverName);
        http.addHeader("Content-Type", "application/json");

        String httpRequestData = "{\"temperatura\":\"" + String(temperatura) +
        "\", \"humedad\":\"" + String(humedad) + "\"}";
        int httpResponseCode = http.POST(httpRequestData);

        if (httpResponseCode > 0) {
            String response = http.getString();
            Serial.println(httpResponseCode);
            Serial.println(response);
        } else {
            Serial.print("Error en la solicitud POST: ");
            Serial.println(httpResponseCode);
        }

        http.end();
    } else {
```

```
        Serial.println("Error en la conexión Wi-Fi");
    }

    delay(10000); // Esperar antes de la siguiente solicitud
}
```

## Instrucciones de Configuración

1. **Modificar Credenciales Wi-Fi:**
  - Cambia `tu_SSID` y `tu_PASSWORD` en el código por tus credenciales de red Wi-Fi.
2. **Subir el Código:**
  - Conecta el ESP32 a tu ordenador mediante un cable USB.
  - Selecciona la placa ESP32 adecuada en **Herramientas > Placa**.
  - Selecciona el puerto correcto en **Herramientas > Puerto**.
  - Haz clic en el botón de subir (Upload) en el Arduino IDE.

## Ejecución y Pruebas

1. **Abrir el Monitor Serial:**
  - Abre el Monitor Serial en el Arduino IDE (Ctrl + Shift + M) para visualizar la salida.
2. **Comprobar Conexión:**
  - Observa el mensaje "Conectado a la red WiFi" en el Monitor Serial, lo que indica que el ESP32 se ha conectado correctamente a tu red.
3. **Verificar Envío de Datos:**
  - El ESP32 enviará datos de temperatura y humedad cada 10 segundos.
  - Observa los códigos de respuesta y cualquier mensaje de error en el Monitor Serial.

## Verificar Almacenamiento en la Base de Datos

Para confirmar que los datos enviados se almacenan correctamente en la base de datos de la API, sigue estos pasos:

1. **Acceso a la API:**
  - Asegúrate de que tienes acceso a la API de `gonaiot.com` y los permisos necesarios para realizar consultas a la base de datos.
2. **Consulta de Datos:**
  - Realiza una consulta a la API para recuperar los datos almacenados. Esto puede hacerse mediante una solicitud GET a la misma URL utilizada para enviar los datos.
  - Puedes usar herramientas como Postman, curl o un navegador web para enviar la solicitud.

### Ejemplo de solicitud GET (usando curl):

```
bash
Copiar código
curl -X GET "https://api.gonaiot.com/plata/datos_dispositivos"
```

### 3. Verificación:

- Revisa la respuesta de la API para asegurarte de que los datos de temperatura y humedad coincidan con los que enviaste desde el ESP32.
- Si los datos aparecen en la respuesta, significa que se almacenaron correctamente.

## Manejo de Errores

### Errores de Conexión

- **Problemas de Conexión Wi-Fi:**
  - Si el ESP32 no puede conectarse a la red Wi-Fi, verifica:
    - Que las credenciales de Wi-Fi sean correctas.
    - Que el ESP32 esté dentro del rango de la red.

### Errores de Respuesta de la API

- **Códigos de Respuesta HTTP:**
  - Un código de respuesta mayor que 0 indica éxito. Los códigos de error deben ser manejados adecuadamente (por ejemplo, 400, 401, 404, 500).
  - En el código de ejemplo, se implementa un manejo básico de errores de respuesta, que proporciona información específica según el código recibido.