

```

#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <TimeLib.h> // Librería para obtener la fecha y hora

// Datos de tu red WiFi
const char* ssid = "Ruz";
const char* password = "12345678";

// Dirección IP del broker MQTT
const char* mqtt_server = "54.166.117.242";

// Instancia del cliente WiFi
WiFiClient espClient;
// Instancia del cliente MQTT
PubSubClient client(espClient);

void setup_wifi() {
    delay(10);
    // Conexión a la red WiFi
    Serial.println();
    Serial.print("Conectando a ");
    Serial.println(ssid);

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.println("Conectado a la red WiFi");
    Serial.println("Dirección IP: ");
    Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length) {
    // Manejar mensajes recibidos desde el broker
    Serial.print("Mensaje recibido [");
    Serial.print(topic);
    Serial.print("] ");
    for (int i = 0; i < length; i++) {
        Serial.print((char)payload[i]);
    }
}

```

```

    }
    Serial.println();
}

void reconnect() {
    // Reconexión al broker MQTT
    while (!client.connected()) {
        Serial.print("Intentando conexión MQTT...");
        if (client.connect("NodeMCU_Client")) {
            Serial.println("Conectado al broker MQTT");
            client.subscribe("iot"); // Suscribirse al tópico deseado
        } else {
            Serial.print("Error al conectar, rc=");
            Serial.print(client.state());
            Serial.println(" Intentando de nuevo en 5 segundos");
            delay(5000);
        }
    }
}

void setup() {
    Serial.begin(115200);
    setup_wifi();
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);

    // Inicializar la librería Time
    configTime(0, 0, "pool.ntp.org"); // Configurar el servidor de tiempo
    NTP
    while (!time(nullptr)) {
        Serial.println("Esperando por la hora del servidor de tiempo...");
        delay(1000);
    }
}

void loop() {
    if (!client.connected()) {
        reconnect();
    }
    client.loop();

    // Obtener la fecha y hora actual
    time_t now = time(nullptr);

```

```

String message = "Hola desde NodeMCU - " + String(ctime(&now));

// Envío de un mensaje al broker MQTT
client.publish("iot", message.c_str());
Serial.println("Mensaje enviado: " + message);

    delay(5000); // Espera de 5 segundos antes de enviar el siguiente
mensaje
}


```

```

#include <DHT.h> // Incluimos la librería DHT

#define DHTPIN D1 // Definimos el pin GPIO al que está conectado el
sensor DHT22
#define DHTTYPE DHT22 // Definimos el tipo de sensor DHT que estamos
utilizando

DHT dht(DHTPIN, DHTTYPE); // Creamos una instancia del objeto DHT

void setup() {
    Serial.begin(9600); // Inicializamos la comunicación serial
    dht.begin();        // Inicializamos el sensor DHT
}

void loop() {
    // Esperamos unos segundos entre lecturas
    delay(2000);

    // Leemos la temperatura en grados Celsius
    float temperatura = dht.readTemperature();
    // Leemos la humedad relativa
    float humedad = dht.readHumidity();

    // Verificamos si la lectura fue exitosa
    if (isnan(temperatura) || isnan(humedad)) {
        Serial.println("Error al leer el sensor DHT22!");
        return;
    }

    // Mostramos los datos por el puerto serial
    Serial.print("Temperatura: ");

```

```
Serial.print(temperatura);  
Serial.print(" °C\t");  
Serial.print("Humedad: ");  
Serial.print(humedad);  
Serial.println("%");  
}
```

```
#include <ESP8266WiFi.h>  
#include <PubSubClient.h>  
#include <TimeLib.h>  
#include <DHT.h>  
  
// Datos de tu red WiFi  
const char* ssid = "Ruz";  
const char* password = "12345678";  
  
// Dirección IP del broker MQTT  
const char* mqtt_server = "54.166.117.242";  
  
// Pin al que está conectado el sensor DHT22  
#define DHTPIN D1  
#define DHTTYPE DHT22  
  
// Instancias de los objetos  
WiFiClient espClient;  
PubSubClient client(espClient);  
DHT dht(DHTPIN, DHTTYPE);  
  
void setup_wifi() {  
  delay(10);  
  Serial.println();  
  Serial.print("Conectando a ");  
  Serial.println(ssid);  
  WiFi.begin(ssid, password);  
  while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
  }  
  Serial.println("");  
  Serial.println("Conectado a la red WiFi");  
  Serial.println("Dirección IP: ");  
  Serial.println(WiFi.localIP());  
}  
  
void callback(char* topic, byte* payload, unsigned int length) {
```

```

Serial.print("Mensaje recibido [");
Serial.print(topic);
Serial.print("] ");
for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
}
Serial.println();
}

void reconnect() {
    while (!client.connected()) {
        Serial.print("Intentando conexión MQTT...");
        if (client.connect("NodeMCU_Client")) {
            Serial.println("Conectado al broker MQTT");
            client.subscribe("iot");
        } else {
            Serial.print("Error al conectar, rc=");
            Serial.print(client.state());
            Serial.println(" Intentando de nuevo en 5 segundos");
            delay(5000);
        }
    }
}

void setup() {
    Serial.begin(115200);
    setup_wifi();
    client.setServer(mqtt_server, 1883);
    client.setCallback(callback);
    dht.begin();
    configTime(0, 0, "pool.ntp.org");
    while (!time(nullptr)) {
        Serial.println("Esperando por la hora del servidor de tiempo...");
        delay(1000);
    }
}

void loop() {
    if (!client.connected()) {
        reconnect();
    }
    client.loop();

    time_t now = time(nullptr);
    String message = "Temperatura: " + String(dht.readTemperature()) + " °C - Humedad: " +
String(dht.readHumidity()) + "% - " + String(ctime(&now));
    client.publish("iot", message.c_str());
    Serial.println("Mensaje enviado: " + message);
}

```

```
    delay(5000);  
}
```