

Taller 4. MQTT

Diego Iván Perea Montealegre (2185751) diego.perea@uao.edu.co

Facultad de Ingeniería, Universidad Autónoma de Occidente

Cali, Valle del Cauca

Primero se realiza la parte de Mosquitto, en donde se tenía que ir a la carpeta de mosquito con la ayuda de la “cmd”, en donde se ponía los temas o los tópicos de cada uno de los elementos, pero hay que aclarar que se debe ejecutar el bróker local Mosquitto, debido a que si este no se ejecuta nos dará error como se muestra en la siguiente figura en donde se realiza una suscripción:

```
C:\>ls-
"ls-" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\>-ls
"-ls" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\>cd "Program Files"

C:\Program Files>ls
"ls" no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Program Files>cd mosquito
El sistema no puede encontrar la ruta especificada.

C:\Program Files>cd mosquitto

C:\Program Files\mosquitto>mosquitto_sub
Error: You must specify a topic to subscribe to.

Use 'mosquitto_sub --help' to see usage.

C:\Program Files\mosquitto>mosquitto_sub -t tema1
Error: No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión.

C:\Program Files\mosquitto>
```

Figura 1. Error de conexión al ejecutar el tema

Activación del mosquito bróker para dar las publicaciones y suscripciones en el sistema

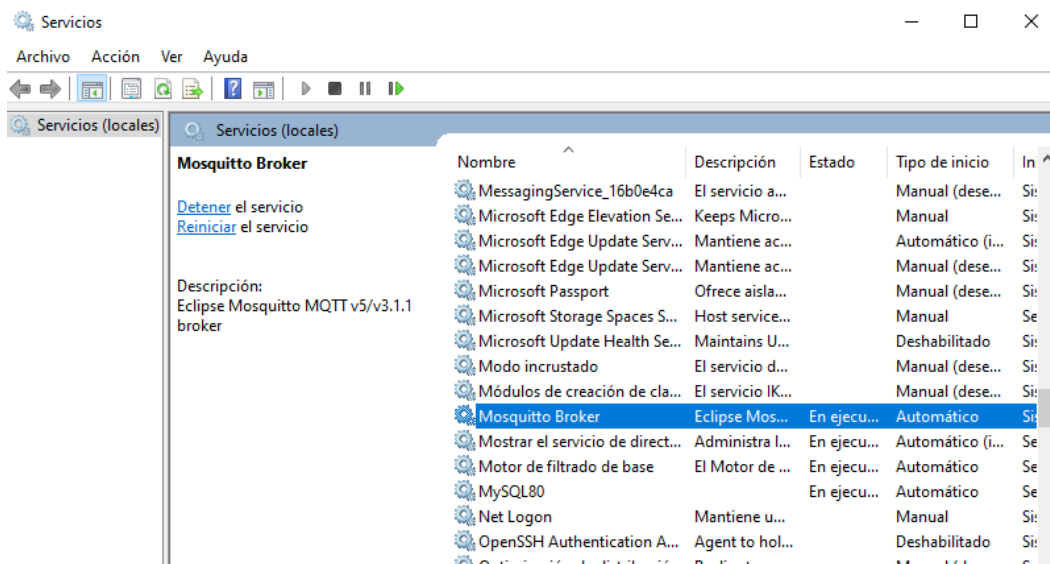


Figura 2. Activación del mosquito Bróker

Para que haya conexión con el tema de mosquito se da una publicación con el mensaje “hola”, en donde esto se visualiza en el otro cmd de la suscripción.

```
C:\Windows\system32\cmd.exe
Only "none" and "username" authentication is supported.
See https://mosquitto.org/ for more information.

C:\Program Files\mosquitto>mosquitto_pub -t tema1 -m "hola"
C:\Program Files\mosquitto>
```

Figura 3. Publicación del tema en mosquito

Ejecutando la publicación del mosquito y dando el tema escrito con éxito se pudo conectar el mosquito

```
C:\Windows\system32\cmd.exe - mosquitto_sub -t tema1
'-ls' no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\>cd "Program Files"

C:\Program Files>ls
'-ls' no se reconoce como un comando interno o externo,
programa o archivo por lotes ejecutable.

C:\Program Files>cd mosquitto
El sistema no puede encontrar la ruta especificada.

C:\Program Files>cd mosquitto

C:\Program Files\mosquitto>mosquitto_sub
Error: You must specify a topic to subscribe to.

Use 'mosquitto_sub --help' to see usage.

C:\Program Files\mosquitto>mosquitto_sub -t tema1
Error: No se puede establecer una conexión ya que el equipo de destino deneg% expresamente dicha conexión.

C:\Program Files\mosquitto>mosquitto_sub -t tema1
Error: No se puede establecer una conexión ya que el equipo de destino deneg% expresamente dicha conexión.

C:\Program Files\mosquitto>mosquitto_sub -t tema1
hola
```

Figura 4. Conexión exitosa de publicación

Para poder ver los servicios del sistema se usa la combinación de teclas (WIN+R) y se escribe el comando “services.msc” y así podemos realizar la activación o detención del Bróker Mosquitto.

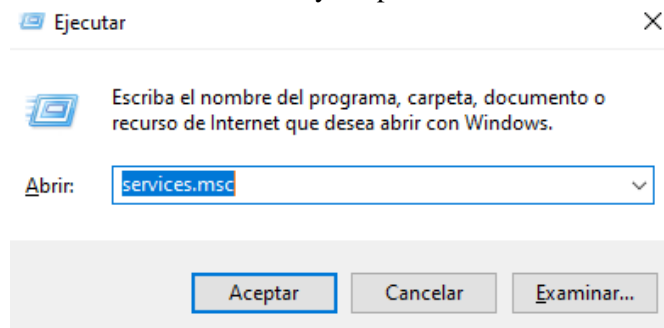


Figura 5. Servicios con cmd

Otra forma de conexiones se realiza con [MQTT explorer](#), en el que se usa de modo diferente, pero se usa los modos de publicación y suscripción necesarios para dar la conexión, entre las dos partes.

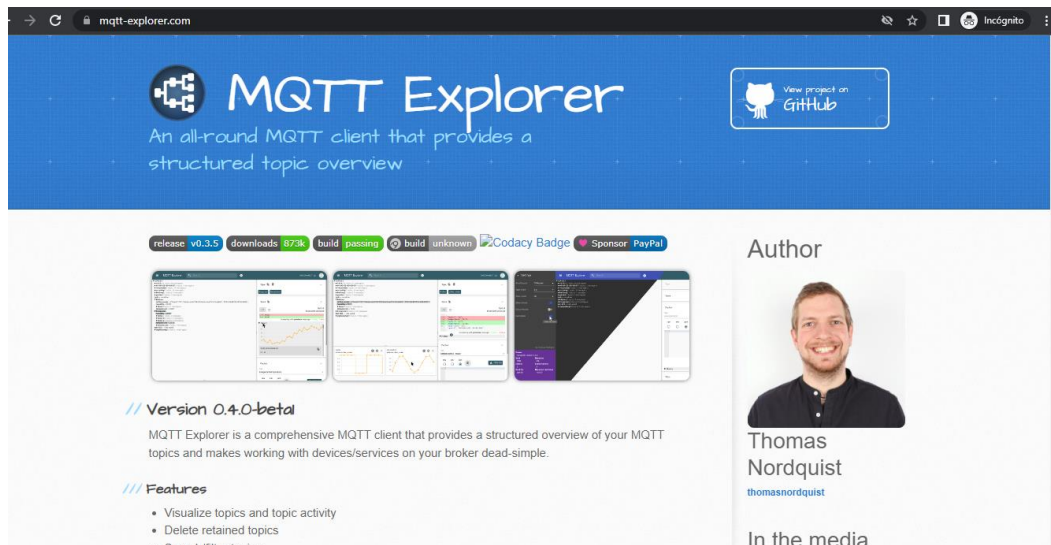


Figura 6 . Página de MQTT explorer.

Para que se de conexiones de publicaciones se crea la suscripción con el nombre de “tema1”, para así tener referencia a que tópico se tiene que publicar el mensaje o dato enviado.

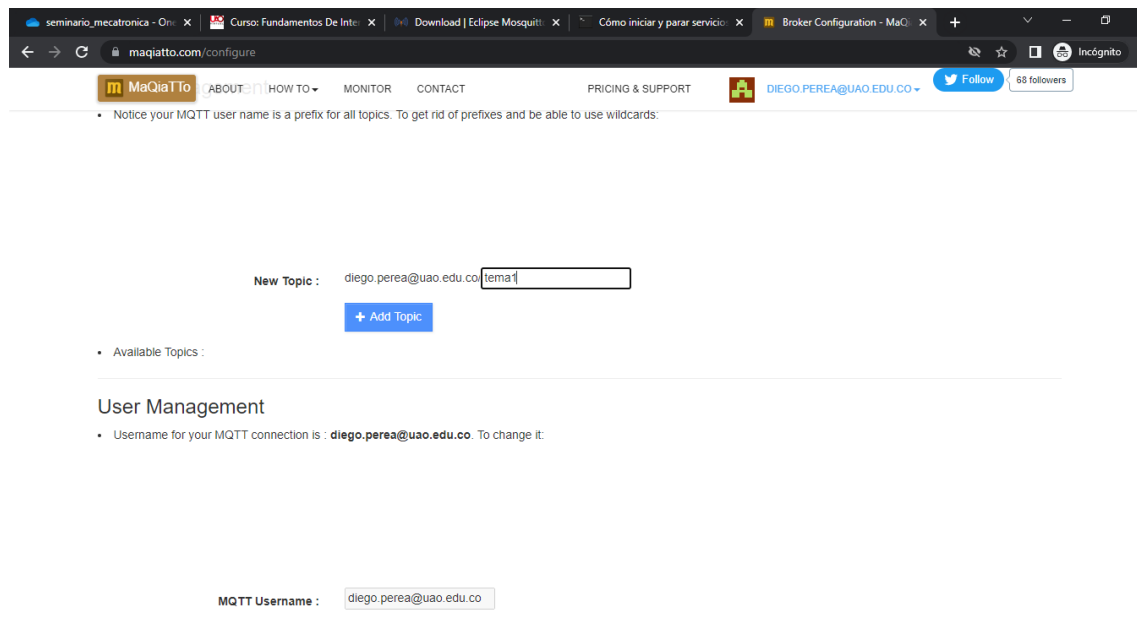


Figura 7. MQTT creación de tópico

Forma visual de realizar la suscripción del tema con la extensión de MQTTlens.

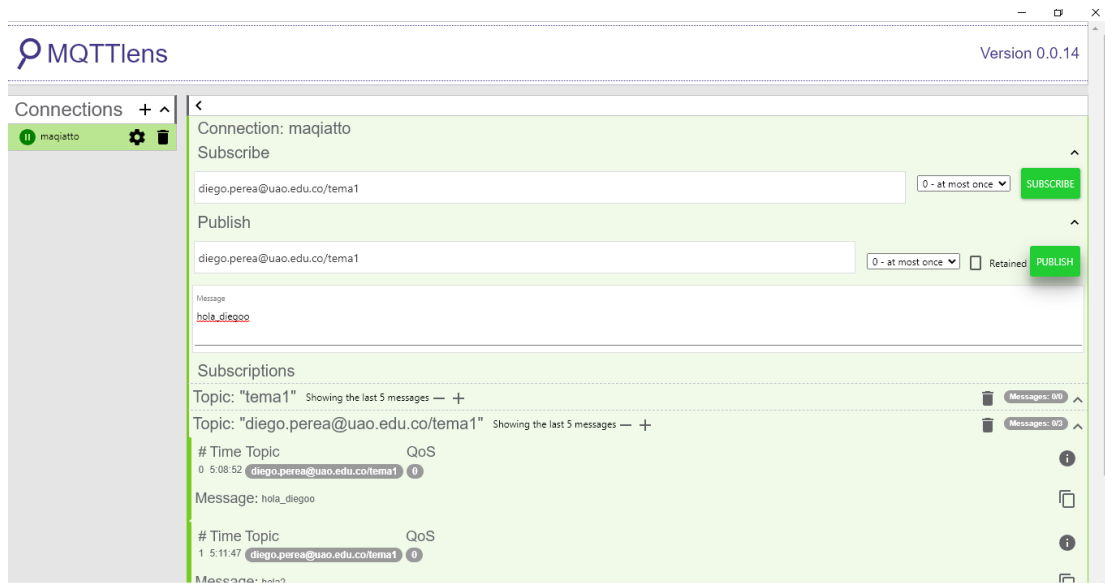


Figura 8. Extensión MQTT lens para la publicación del t pico.

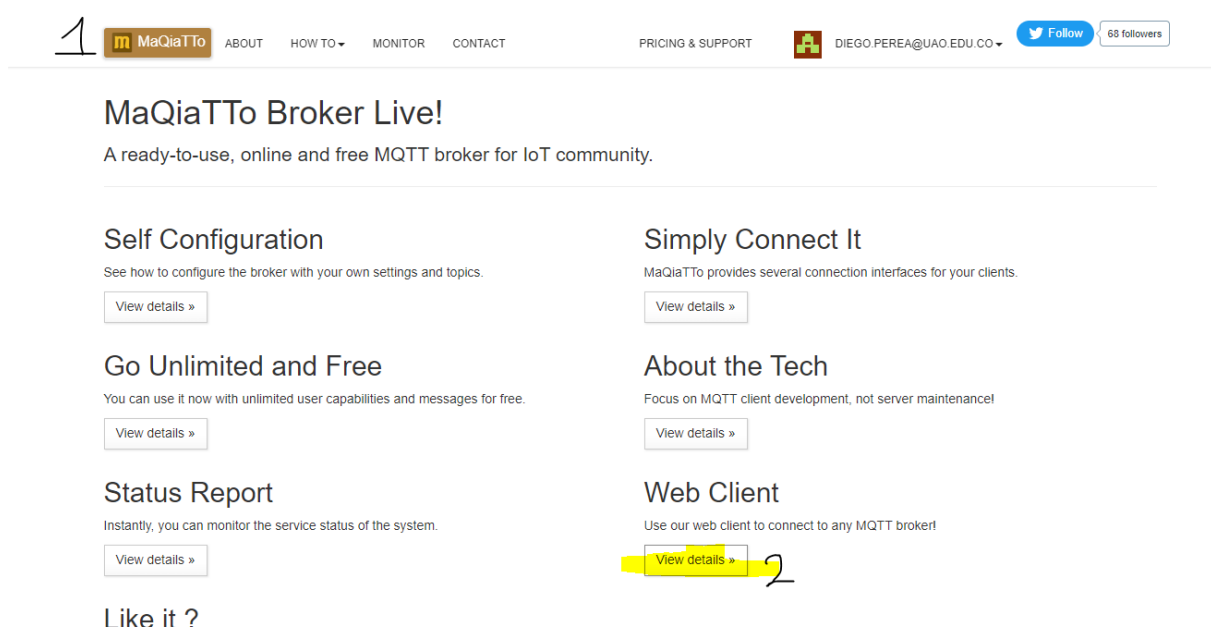


Figura 9. Maquiatto.com

← → ↻ No seguro maqiatto.com/webclient/

MQTT Websocket Interface for Everyone!

Connect and Publish/Subscribe on your topics via MQTT Websocket Client.
MaQiaTTto users! Check your [available topics](#) before proceeding.

Connected to MQTT Broker.
Client ID : 4654250f-92f6-455a-a3fe-81d0759b7756

Topic to Subscribe:

diego.perea@uao.edu.co/tema1

Subscribe

Publish Topic:

diego.perea@uao.edu.co/tema1

Publish Message:

hola?

Publish

Actions and Received Messages:

[2022.8.30 17:11:22.330] Subscribed to : diego.perea@uao.edu.co/tema1
[2022.8.30 17:11:47.41] Received message : hola?, Topic : diego.perea@uao.edu.co/tema1
[2022.8.30 17:11:55.808] Received message : hola_diegoo, Topic : diego.perea@uao.edu.co/tema1

Figura 10. maqiatto web

MQTTlens Version 0.0.14

Connections + ^

maqiatto

Connection Details

Connection name: maqiatto

Connection color scheme: [Green bar]

Hostname: tcp:// maqiatto.com

Port: 1883

Client ID: lens_FRvmbmUQl2K3qX4h2oTE6TZITO

Generate a random ID

Session: ☒ Clean Session

Automatic Connection: ☒ Automatic Connection

Keep Alive: 120 seconds

Credentials

Username: diego.perea@uao.edu.co

Password: [Redacted]

Last-Will

Figura 11. User y password en MQTT LENS

MQTT Websocket Interface for Everyone!

Connect and Publish/Subscribe on your topics via MQTT Websocket Client.
MaQiaTTto users! Check your [available topics](#) before proceeding.

Connected to MQTT Broker.

Client ID : ed1b759f-0738-425c-9141-9de44ff005db

Topic to Subscribe:

Subscribe

Publish Topic:

Publish Message:

Publish

Actions and Received Messages:

[2022.9.8 21:12:25.33] Subscribed to : diego.perea@uao.edu.co/tema2
[2022.9.8 21:12:36.74] Received message : , Topic : diego.perea@uao.edu.co/tema2
[2022.9.8 21:23:35.78] Subscribed to : diego.perea@uao.edu.co/tema2
[2022.9.8 21:23:36.483] Subscribed to : diego.perea@uao.edu.co/tema2
[2022.9.8 21:23:39.180] Received message : , Topic : diego.perea@uao.edu.co/tema2
[2022.9.8 21:23:57.895] Received message : hola celular , Topic : diego.perea@uao.edu.co/tema2
[2022.9.8 21:24:19.55] Received message : hola desde el celular, Topic : diego.perea@uao.edu.co/tema2

Figura 12. Conexión con Maqiatto con el celular

Para ver los datos mediante plataforma bróker iot se utiliza cayence que ayuda a la facilidad de las conexiones

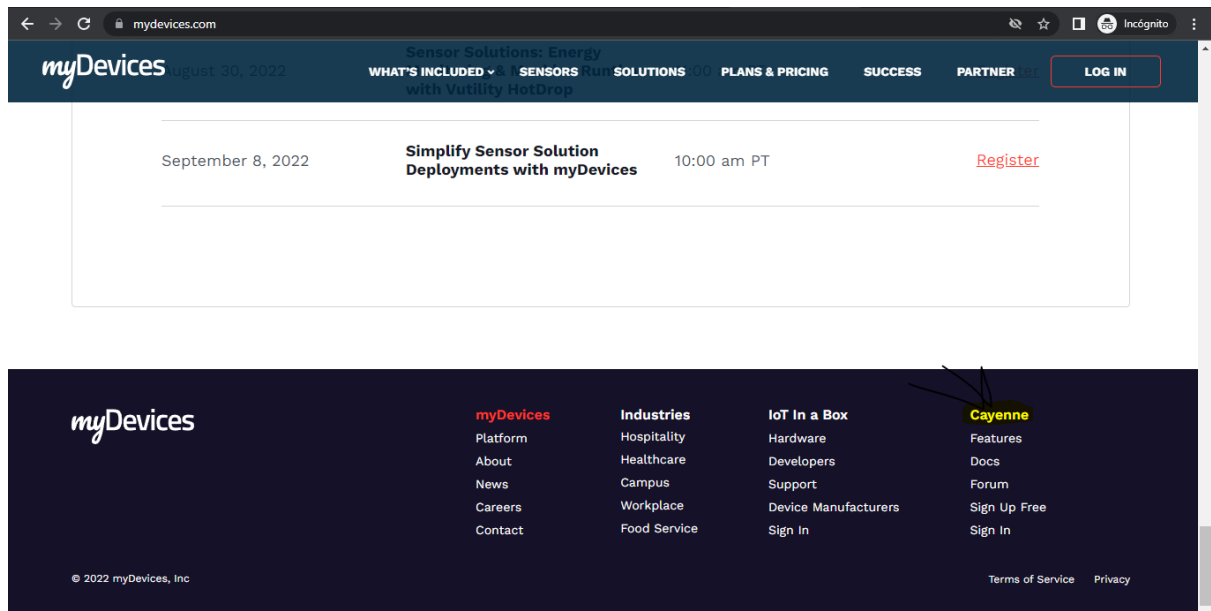


Figura 13. Página de mydevices

Para realizarlo se hizo en Cayenne

```

{"Fecha":"2022-09-07","Hora":"21:54:55","Temperatura(°C)":28,"Humedad(%)":55}
{"Fecha":"2022-09-07","Hora":"21:54:57","Temperatura(°C)":28,"Humedad(%)":55}
{"Fecha":"2022-09-07","Hora":"21:54:59","Temperatura(°C)":28,"Humedad(%)":55}
Enviando Temperatura a Cayenne:
28.00
Enviando Humedad a Cayenne
55.00
{"Fecha":"2022-09-07","Hora":"21:55:01","Temperatura(°C)":28,"Humedad(%)":55}
{"Fecha":"2022-09-07","Hora":"21:55:03","Temperatura(°C)":28,"Humedad(%)":55}

```

Figura 14. Envío de datos a cayenne

Se puede observar en el resultado que el vio de datos es inmediato a la plataforma de Cayenne

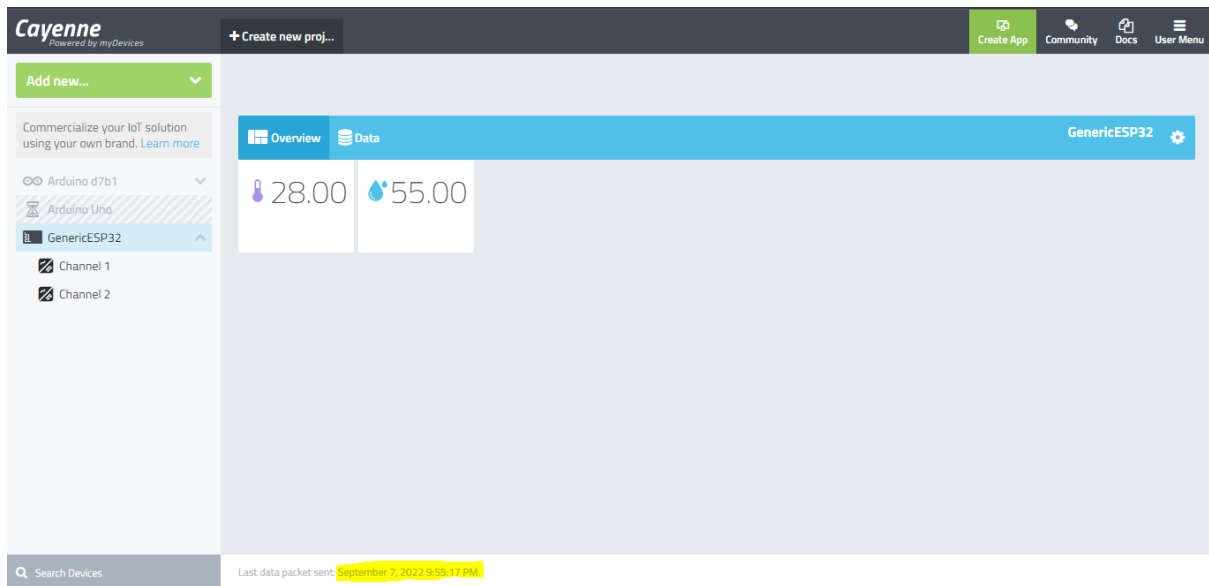


Figura 15. Datos obtenido de los sensores en ESP32

```

#include <Arduino.h>
#include <ArduinoJson.h>
//LIBRERIAS PARA FECHA Y HORA
#include <WiFi.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
//LIBRERIAS PARA DHT11 (TEMPERATURA Y HUMEDAD)
#include <Adafruit_Sensor.h>
#include <DHT.h>
//libreria cayenne
#include <CayenneMQTTESP32.h> //Librería de Cayenne MQTT
#define CAYENNE_PRINT Serial

//DEFINICION DE PINES DHT11
#define DHTPIN 4 // 4 = PIN D4
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

```

```

//CONFIG PARA ----FECHA Y HORA-----
// Replace with your network credentials
const char* ssid = "**you**name**wifi";
const char* wifipassword = "**you*password*wifi";

// Define NTP Client to get time
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);

// Variables to save date and time
String formattedDate;
String dayStamp;
String timeStamp;

//potenciometro ph
const int portPin=34;
int valor=0;

//info CAYENE-----
// Parámetros de conexión a Cayenne. Esto debe obtenerse del Tablero de
Cayenne.
char username[] = "5475eb90-28b1-11ed-baf6-35fab7fd0a**";
char password[] = "46dd0481c83ab7f4ad4aa2654369e7692bc45b**";
char clientID[] = "3c8d0360-2e4c-11ed-baf6-35fab7fd0a**";
//-----

void setup() {
// Initialize Serial Monitor
Serial.begin(9600);
//CODIGO----FECHA Y HORA-----
WiFi.mode(WIFI_STA);
WiFi.begin(ssid, wifipassword);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
}
// Initialize a NTPClient to get time
timeClient.begin();
// Set offset time in seconds to adjust for your timezone, for example:
// COLOMBIA -5 , entonces -5*3600 -> -18000
timeClient.setTimeOffset(-18000); //Thailand +7 = 25200

//configuracion CAYENE
Cayenne.begin(username, password, clientID, ssid, wifipassword);
}

```



```

void loop() {
  while(!timeClient.update()) {
    timeClient.forceUpdate();
  }
  // The formattedDate comes with the following format:
  // 2018-05-28T16:00:13Z
  // We need to extract date and time
  formattedDate = timeClient.getFormattedDate();
  // Extract date
  int splitT = formattedDate.indexOf("T");
  dayStamp = formattedDate.substring(0, splitT);
  //Serial.print("DATE: ");
  //Serial.println(dayStamp);
  // Extract time
  timeStamp = formattedDate.substring(splitT+1, formattedDate.length()-1);
  //Serial.print("HOUR: ");
  //Serial.println(timeStamp);

  //CODIGO----TEMPERATURA Y HUMEDAD-----
  float h= dht.readHumidity();
  float t =dht.readTemperature();
  //potenciometro ph
  valor=analogRead(portPin)/292.5;
  //-----

  //----CODIGO JSON-----
  String variable;

  DynamicJsonDocument doc(1024);

  doc["Fecha"] = dayStamp;
  doc["Hora"] = timeStamp;
  doc["Temperatura(°C)"] = t;
  doc["Humedad(%)"] = h;
  //doc["Ph"] = valor;

  serializeJson(doc, variable);
  Serial.println(variable);
  Cayenne.loop();
  delay(1000);
}

// Función de Cayenne para enviar datos del sensor al canal 1.
CAYENNE_OUT(1)

```

```

{
  float t = dht.readTemperature();//Se lee la temperatura y se asigna el
valor a "t".
  //Lectura de temperatura se enviara a Cayenne en el canal 1.
  //Envió de lecturas cada 10 segundos.
  Cayenne.virtualWrite(1, t);
  //Se imprimen los siguientes datos en el monitor serie.
  CAYENNE_PRINT.println("Enviando Temperatura a Cayenne:");
  CAYENNE_PRINT.println(t);
}

// Función de Cayenne para enviar datos del sensor al canal 2.
CAYENNE_OUT(2)
{
  float h = dht.readHumidity(); //Se lee la humedad y se asigna el valor a "h"
  //Lectura de Humedad se enviaran a Cayenne en el canal 2.
  //Envió de lecturas cada 10 segundos.
  Cayenne.virtualWrite(2, h);
  //Se imprimen los siguientes datos en el monitor serie.
  CAYENNE_PRINT.println("Enviando Humedad a Cayenne");
  CAYENNE_PRINT.println(h);
}

```

Figura 16. Código de cayenne en ESP32

Se realiza la conexión a MQTT maqiatto:

```

#include <Arduino.h>
#include <ArduinoJson.h>

#include <PubSubClient.h>

//LIBRERIAS PARA FECHA Y HORA
#include <WiFi.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
//LIBRERIAS PARA DHT11 (TEMPERATURA Y HUMEDAD)
#include <Adafruit_Sensor.h>
#include <DHT.h>

//DEFINICION DE PINES DHT11
#define DHTPIN 4    // 4 = PIN D4
#define DHTTYPE    DHT11
DHT dht(DHTPIN, DHTTYPE);
//potenciometro ph
const int portPin=34;
int valorPh=0;

#define mqttUser "diego.perea@uao.edu.co" //user maqiatto

```

```

#define mqttPass "***" //contraseña maqiatto
#define mqttPort 1883
const char* ssid = "***yor**name**wifi"; // name wifi
const char* password = "****"; //contraseña wifi
char mqttBroker[] = "maqiatto.com"; //ip del servidor
char mqttClientId[] = "tema2"; //cualquier nombre
char inTopic[] = "diego.perea@uao.edu.co/tema2";

// Define NTP Client to get time
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);

// Variables to save date and time
String formattedDate;
String dayStamp;
String timeStamp;

void callback(char* topic, byte* payload, unsigned int length) {
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");
    for (int i=0;i<length;i++) {
        Serial.print((char)payload[i]);
    }
    Serial.println();
}

WiFiClient BClient;
PubSubClient client(BClient);
void reconnect() {
    // Loop until we're reconnected
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        // Attempt to connect
        if (client.connect("diego.perea@uao.edu.co/tema2", mqttUser, mqttPass)) {
            Serial.println("connected");
            // Once connected, publish an announcement...
            float h= dht.readHumidity();
            float t =dht.readTemperature();
            //potenciometro ph
            valorPh=analogRead(portPin)/292.5;
            //-----
            String variable;
            StaticJsonDocument<256> doc;
            doc["temperatura"] = t;
            doc["humedad"] = h;
            doc["idnodo"] = 1;
            doc["Ph"]=valorPh;

```

```

doc["Fecha"] = dayStamp;
serializeJson(doc, variable);
int lon = variable.length()+1;
Serial.println(variable);
char datojson[lon];
variable.toCharArray(datojson, lon);
client.publish(inTopic,datojson);
client.disconnect();
delay(5000);
// ... and resubscribe
//client.subscribe("topic2");
} else {
Serial.print("failed, rc=");
Serial.print(client.state());
Serial.println(" try again in 5 seconds");
// Wait 5 seconds before retrying
delay(5000);
}
}
}

void setup_wifi() {
  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  // Initialize a NTPClient to get time
  timeClient.begin();
  // Set offset time in seconds to adjust for your timezone, for example:
  // COLOMBIA -5 , entonces -5*3600 -> -18000
  timeClient.setTimeOffset(-18000); //Thailand +7 = 25200
}

void setup()
{
  Serial.begin(9600); //Serial connection

```

```

    setup_wifi(); //WiFi connection
    client.setServer( mqttBroker, mqttPort );
    client.setCallback( callback );
    Serial.println("Setup done");
    delay(1500);
}

void loop()
{
    while(!timeClient.update()) {
        timeClient.forceUpdate();
    }
    // The formattedDate comes with the following format:
    // 2018-05-28T16:00:13Z
    // We need to extract date and time
    formattedDate = timeClient.getFormattedDate();
    // Extract date
    int splitT = formattedDate.indexOf("T");
    dayStamp = formattedDate.substring(0, splitT);
    //Serial.print("DATE: ");
    //Serial.println(dayStamp);
    // Extract time
    timeStamp = formattedDate.substring(splitT+1, formattedDate.length()-1);

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

```

```

[2022.9.14 23:7:49.389] Received message : {"temperatura":27.60000038,"humedad":58,"idnodo":1,"Ph":9,"Fecha":"2022-09-14"}, Topic : diego.perea@uao.edu.co/tema2
[2022.9.14 23:7:54.816] Received message : {"temperatura":27.60000038,"humedad":58,"idnodo":1,"Ph":9,"Fecha":"2022-09-14"}, Topic : diego.perea@uao.edu.co/tema2
[2022.9.14 23:8:1.465] Received message : {"temperatura":27.60000038,"humedad":58,"idnodo":1,"Ph":9,"Fecha":"2022-09-14"}, Topic : diego.perea@uao.edu.co/tema2
[2022.9.14 23:8:6.909] Received message : {"temperatura":27.60000038,"humedad":58,"idnodo":1,"Ph":9,"Fecha":"2022-09-14"}, Topic : diego.perea@uao.edu.co/tema2
[2022.9.14 23:8:12.438] Received message : {"temperatura":27.60000038,"humedad":58,"idnodo":1,"Ph":9,"Fecha":"2022-09-14"}, Topic : diego.perea@uao.edu.co/tema2
[2022.9.14 23:8:23.280] Received message : {"temperatura":27.60000038,"humedad":58,"idnodo":1,"Ph":9,"Fecha":"2022-09-14"}, Topic : diego.perea@uao.edu.co/tema2
[2022.9.14 23:8:28.594] Received message : {"temperatura":27.60000038,"humedad":58,"idnodo":1,"Ph":9,"Fecha":"2022-09-14"}, Topic : diego.perea@uao.edu.co/tema2

```

Figura 17. Recepción de datos a MQTT maqiatto

Se realiza la conexión Mosquitto:

```
C++ main.cpp box_smart + src C++ main.cpp Taller_4 MQTT + src mosquitto.conf X C++ main.cpp taller_5 + src
C: > Program Files > mosquitto > mosquitto.conf
207 # Note that on Windows this has no effect and so mosquitto should be started by
208 # the user you wish it to run as.
209 #user mosquitto
210
211 # =====
212 # Listeners
213 # =====
214 listener 1883 0.0.0.0
215 allow_anonymous true
216
217
218
219 # Listen on a port/ip address combination. By using this variable
220 # multiple times, mosquitto can listen on more than one port. If
221 # this variable is used and neither bind_address nor port given,
222 # then the default listener will not be started.
223 # The port number to listen on must be given. Optionally, an ip
224 # address or host name may be supplied as a second argument. In
225 # this case, mosquitto will attempt to bind the listener to that
226 # address and so restrict access to the associated network and
227 # interface. By default, mosquitto will listen on all interfaces.
228 # Note that for a websockets listener it is not possible to bind to a host
```

Windows Defender Firewall con seguridad avanzada

Archivo Acción Ver Ayuda

Windows Defender Firewall con seguridad avanzada

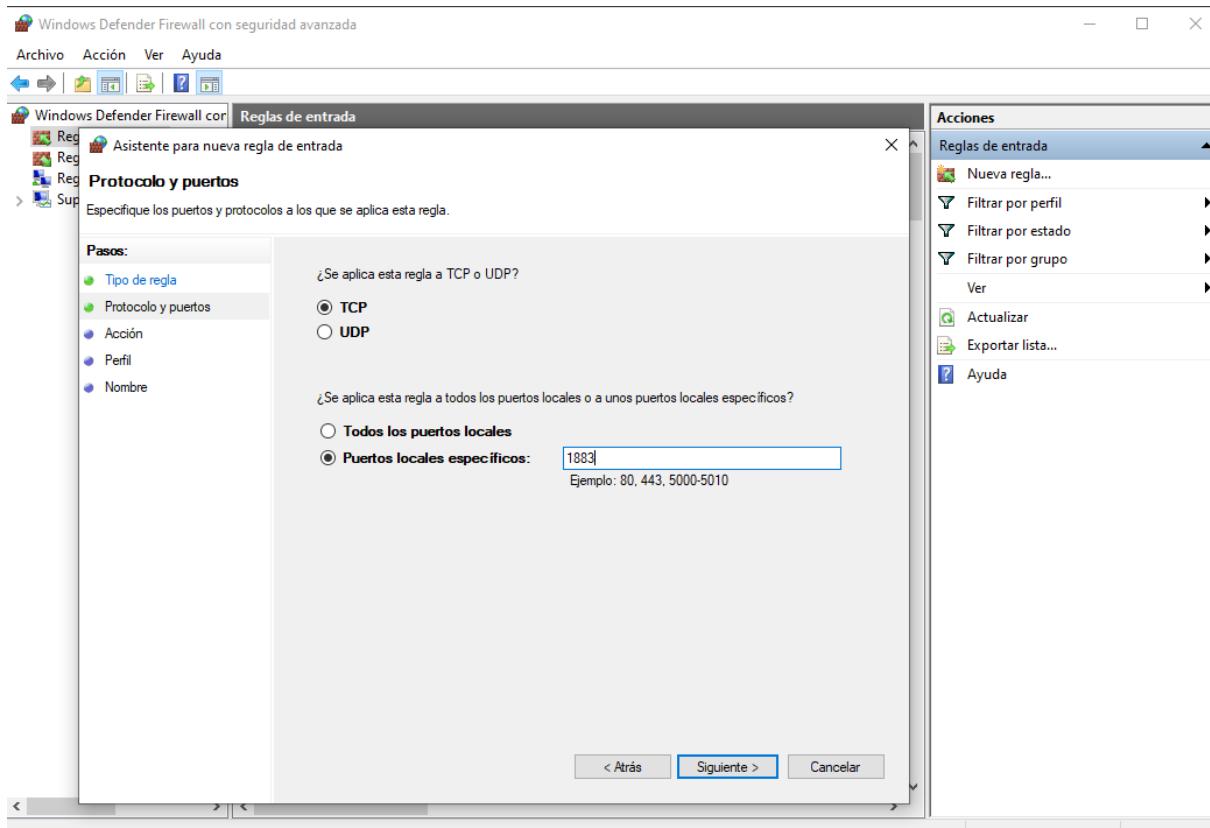
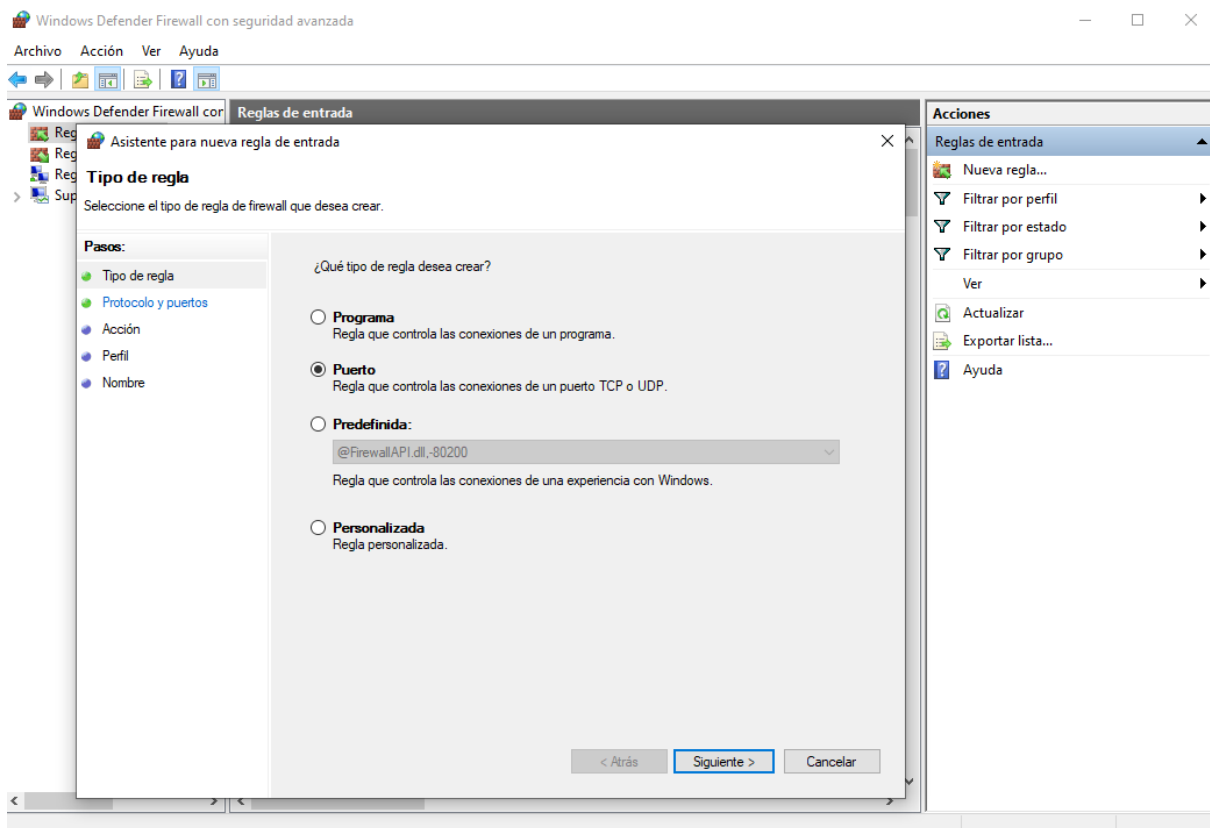
Reglas de entrada

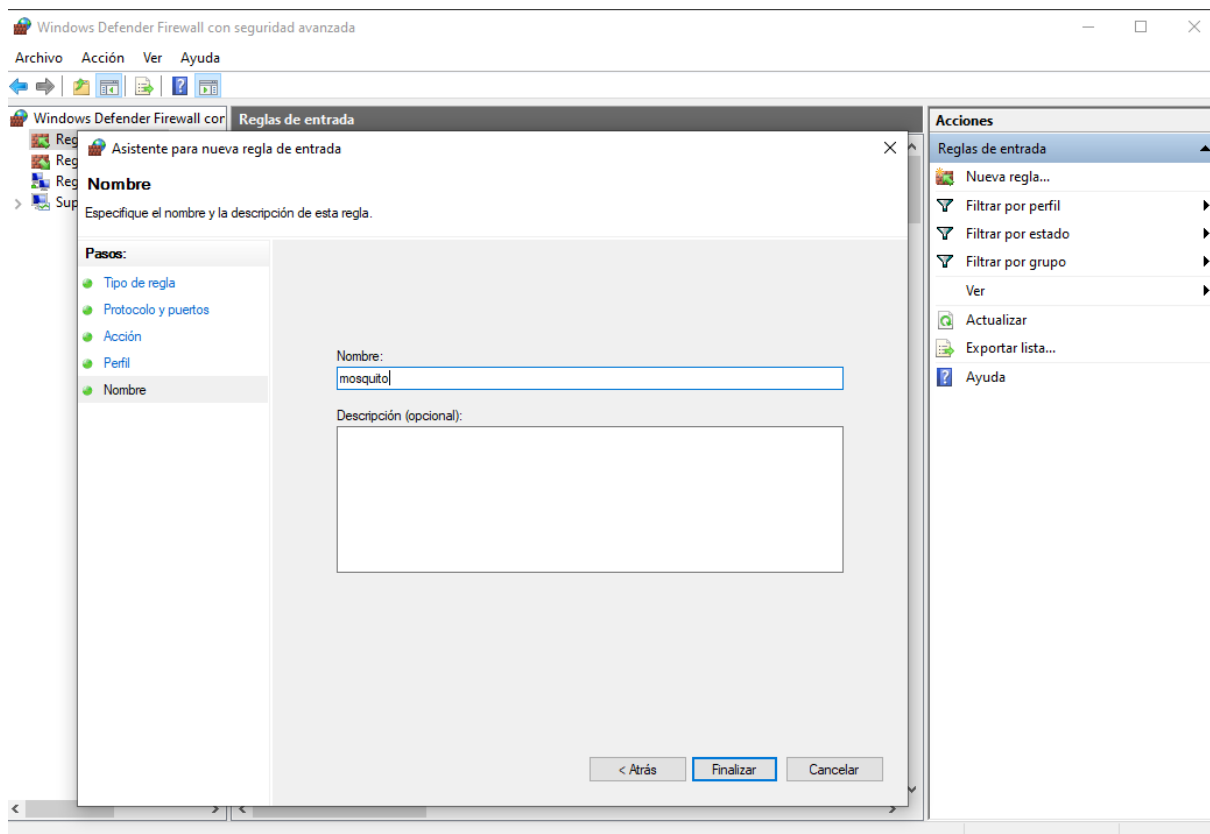
Nombre	Grupo	Perfil	Habilitado	Acción
AnyDesk		Domestic	Sí	Permitir
AnyDesk		Domestic	Sí	Permitir
AnyDesk		Público	Sí	Permitir
AnyDesk		Privado	Sí	Permitir
AnyDesk		Público	Sí	Permitir
AnyDesk		Privado	Sí	Permitir
Apache HTTP Server		Público	Sí	Permitir
Apache HTTP Server		Público	Sí	Permitir
Node.js JavaScript Runtime		Público	Sí	Permitir
Node.js JavaScript Runtime		Público	Sí	Permitir
Packet Tracer Executable		Público	Sí	Permitir
Packet Tracer Executable		Público	Sí	Permitir
Port 3306		Todo	Sí	Permitir
Port 33060		Todo	Sí	Permitir
RStudio R Session		Público	Sí	Permitir
RStudio R Session		Público	Sí	Permitir
VcXsrv windows xserver		Público	Sí	Permitir
VcXsrv windows xserver		Público	Sí	Permitir
Visual Studio Code		Público	Sí	Permitir
Visual Studio Code		Público	Sí	Permitir
Xming X Server		Público	Sí	Permitir
Xming X Server		Público	Sí	Permitir
@FirewallAPI.dll, -80201	@FirewallAPI.dll, -80200	Todo	Sí	Permitir
@FirewallAPI.dll, -80206	@FirewallAPI.dll, -80200	Todo	Sí	Permitir
Skype	{78E1CD88-49E3-476E-B926-...	Todo	Sí	Permitir
Skype	{78E1CD88-49E3-476E-B926-...	Todo	Sí	Permitir
Administración de tarjetas inteligentes vi...	Administración de tarjetas i...	Domestic	No	Permitir
Administración de tarjetas inteligentes vi...	Administración de tarjetas i...	Privado	No	Permitir
Administración de tarjetas inteligentes vi...	Administración de tarjetas i...	Privado	No	Permitir

Acciones

Reglas de entrada

- Nueva regla...
- Filtrar por perfil
- Filtrar por estado
- Filtrar por grupo
- Ver
- Actualizar
- Exportar lista...
- Ayuda





Código de mosquito

```
#include <Arduino.h>

#include <ArduinoJson.h>
#include <WiFi.h>
#include <PubSubClient.h>
//LIBRERIAS PARA DHT11 (TEMPERATURA Y HUMEDAD)
#include <Adafruit_Sensor.h>
#include <DHT.h>
//LIBRERIAS PARA FECHA Y HORA
#include <WiFi.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
//DEFINICION DE PINES DHT11
#define DHTPIN 4    // 4 = PIN D4
#define DHTTYPE    DHT11
DHT dht(DHTPIN, DHTTYPE);
//potenciometro ph
const int portPin=34;
int valorPh=0;

// Define NTP Client to get time
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);
```



```

// Variables to save date and time
String formattedDate;
String dayStamp;
String timeStamp;

#define mqttUser ""
#define mqttPass ""
#define mqttPort 1883
const char* ssid = "**name**wifi";//name wifi
const char* password = "**"; // clave de wifi
char mqttBroker[] = "192.168.*.*"; //ip del servidor
char mqttClientId[] = "tema1"; //cualquier nombre
char inTopic[] = "tema1";//topcico a suscribirse

void callback(char* topic, byte* payload, unsigned int length) {
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");
    for (int i=0;i<length;i++) {
        Serial.print((char)payload[i]);
    }
    Serial.println();
}

WiFiClient BClient;
PubSubClient client(BClient);
void reconnect() {
    // Loop until we're reconnected
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        // Attempt to connect
        if (client.connect("", mqttUser, mqttPass)) {
            Serial.println("connected");
            // Once connected, publish an announcement...
            // Once connected, publish an announcement...
            float h= dht.readHumidity();
            float t =dht.readTemperature();
            //potenciometro ph
            valorPh=analogRead(portPin)/292.5;
            //-----
            String variable;
            StaticJsonDocument<256> doc;
            doc["temperatura"] = t;
            doc["humedad"] = h;
            doc["idnodo"] = 1;
            doc["Ph"]=valorPh;
            doc["Fecha"] = dayStamp;
            serializeJson(doc, variable);

```

```

    int lon = variable.length()+1;
    Serial.println(variable);
    char datojson[lon];
    variable.toCharArray(datojson, lon);
    client.publish(inTopic,datojson);
    client.disconnect();
    delay(5000);
    // ... and resubscribe
    //client.subscribe("topic2");
  } else {
    Serial.print("failed, rc=");
    Serial.print(client.state());
    Serial.println(" try again in 5 seconds");
    // Wait 5 seconds before retrying
    delay(5000);
  }
}
}
}

void setup_wifi() {
  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  // Initialize a NTPClient to get time
  timeClient.begin();
  // Set offset time in seconds to adjust for your timezone, for example:
  // COLOMBIA -5 , entonces -5*3600 -> -18000
  timeClient.setTimeOffset(-18000); //Thailand +7 = 25200
}

void setup()
{
  Serial.begin(9600); //Serial connection
  setup_wifi(); //WiFi connection
  client.setServer(mqttBroker, mqttPort );
  client.setCallback( callback );
  Serial.println("Setup done");
}

```

```

    delay(1500);
}

void loop(){
    while(!timeClient.update()) {
        timeClient.forceUpdate();
    }
    // The formattedDate comes with the following format:
    // 2018-05-28T16:00:13Z
    // We need to extract date and time
    formattedDate = timeClient.getFormattedDate();
    // Extract date
    int splitT = formattedDate.indexOf("T");
    dayStamp = formattedDate.substring(0, splitT);
    //Serial.print("DATE: ");
    //Serial.println(dayStamp);
    // Extract time
    timeStamp = formattedDate.substring(splitT+1, formattedDate.length()-1);
    if (!client.connected()) {
        reconnect();
    }
    client.loop();
}
}

```

Figura 18. Código mosquitto

```

{"temperatura":26.70000076,"humedad":62,"idnodo":1,"Ph":14,"Fecha":"2022-09-15"}
Attempting MQTT connection...connected
{"temperatura":26.70000076,"humedad":62,"idnodo":1,"Ph":0,"Fecha":"2022-09-15"}
Attempting MQTT connection...connected
{"temperatura":26.70000076,"humedad":62,"idnodo":1,"Ph":7,"Fecha":"2022-09-15"}
Attempting MQTT connection...connected
{"temperatura":13.30000019,"humedad":159,"idnodo":1,"Ph":5,"Fecha":"2022-09-15"}
Attempting MQTT connection...connected
{"temperatura":26.79999924,"humedad":62,"idnodo":1,"Ph":5,"Fecha":"2022-09-15"}

```

Figura 19. Envío de datos a mosquitto

[illegible]

Figura 20. Visualización de datos en mosquitto

Bibliografía

- [1]"3.1.4 Características Generales de Cada Capa", Cidecame.uaeh.edu.mx, 2022. [Online]. Available: [http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro27/314_caractersticas_generales_de_cada_capa.html#:~:text=En%20esta%20capa%20se%20ubican,tarjetas%20y%20repetidores%20\(hub\).](http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro27/314_caractersticas_generales_de_cada_capa.html#:~:text=En%20esta%20capa%20se%20ubican,tarjetas%20y%20repetidores%20(hub).) [Accessed: 10- Sep- 2022]
- [2]F. Ayala, "Dashboard con Cayenne IOT + ESP32 y DHT22 - UNIT Electronics", UNIT Electronics, 2022. [Online]. Available: <https://blog.uelectronics.com/tarjetas-desarrollo/dashboard-de-temperatura-y-humedad-con-cayenne-iot-esp32-y-dht22/>. [Accessed: 10- Sep- 2022]
- [3]"GitHub - arduino-libraries/Bridge: Bridge Library for Arduino", GitHub, 2022. [Online]. Available: <https://github.com/arduino-libraries/Bridge>. [Accessed: 10- Sep- 2022]
- [4]2022. [Online]. Available: <https://askarduino.com/q/83582/conecte-esp32-a-traves-de-mqtt>. [Accessed: 10- Sep- 2022]
- [5]"ESP32: Sending JSON messages over MQTT - techtutorialsx", techtutorialsx, 2022. [Online]. Available: <https://techtutorialsx.com/2017/04/29/esp32-sending-json-messages-over-mqtt/>. [Accessed: 10- Sep- 2022]
- [6]Youtube.com, 2022. [Online]. Available: <https://www.youtube.com/watch?v=5tG3JXFYrUo>. [Accessed: 10- Sep- 2022]
- [7]Youtube.com, 2022. [Online]. Available: <https://www.youtube.com/watch?v=l31o3GfaSNU>. [Accessed: 10- Sep- 2022]

[8]"ESP32 y Broker MQTT | Tienda y Tutoriales Arduino", Prometec.net, 2022. [Online]. Available: <https://www.prometec.net/esp32-mqtt/>. [Accessed: 10- Sep- 2022]

[9]"Can't connect ESP32 to MQTT Broker (Mosquitto) - ESP32 Forum", Esp32.com, 2022. [Online]. Available: <https://esp32.com/viewtopic.php?t=2605>. [Accessed: 10- Sep- 2022]

[10]"GitHub - plapointe6/EspMQTTClient: Wifi and MQTT handling for ESP8266 and ESP32", GitHub, 2022. [Online]. Available: <https://github.com/plapointe6/EspMQTTClient>. [Accessed: 10- Sep- 2022]

[11]"ESP32/8266 With MQTT (Mosquitto) - Bryce Automation", Bryce Automation, 2022. [Online]. Available: <https://bryceautomation.com/index.php/2021/08/09/esp32-8266-with-mqtt/>. [Accessed: 10- Sep- 2022]