Taller 4. MQTT

Diego Iván Perea Montealegre (2185751) [diego.perea@uao.edu.co](mailto: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:

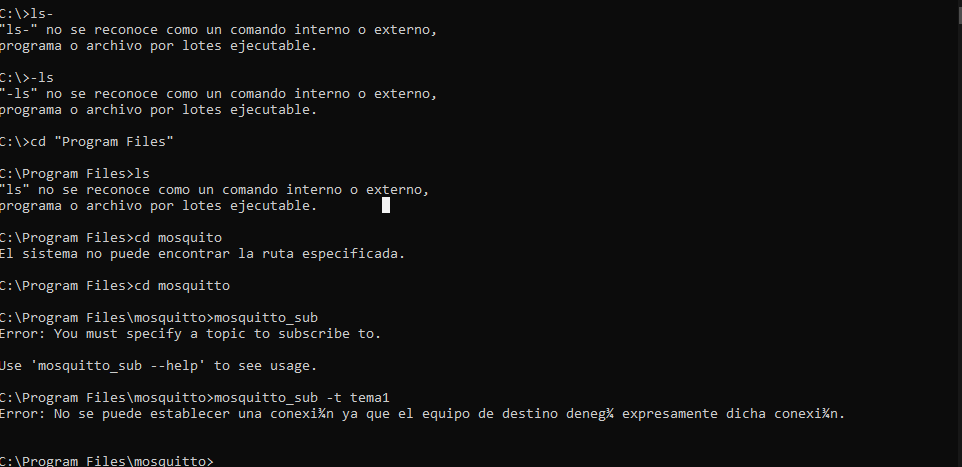


Figura 1. Error de conexión al ejecutar el tema

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

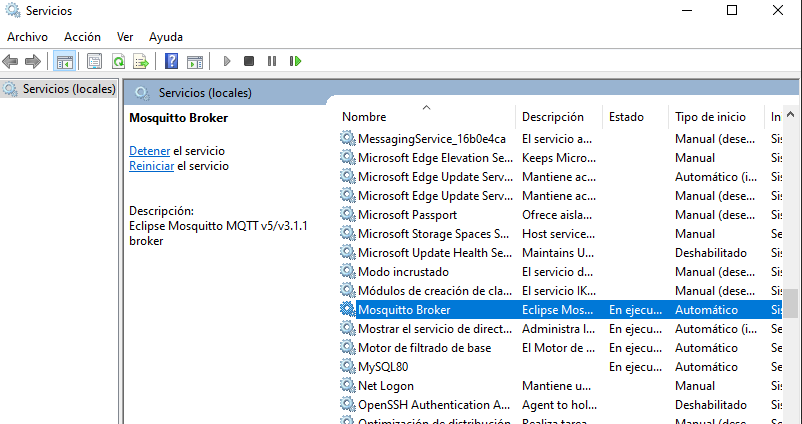


Figura 2. Activación del mosquito Bróker

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.

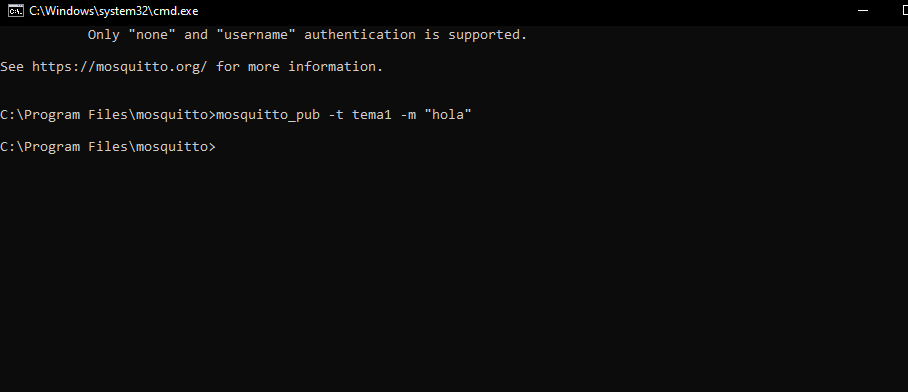


Figura 3. Publicación del tema en mosquito

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

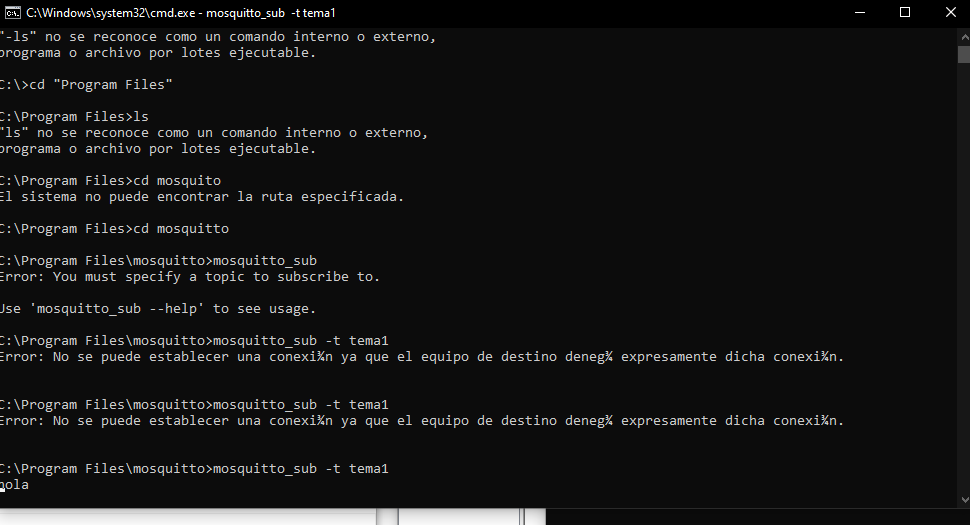


Figura 4. Conexión exitosa de publicación

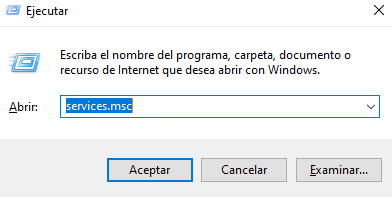
Para poder ver los servicios del sistema se usa la combinación de teclas (WIN+R) y se escribe el comando “services.msc” y así podemos realizar la activación o detención del Bróker Mosquitto. 

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.

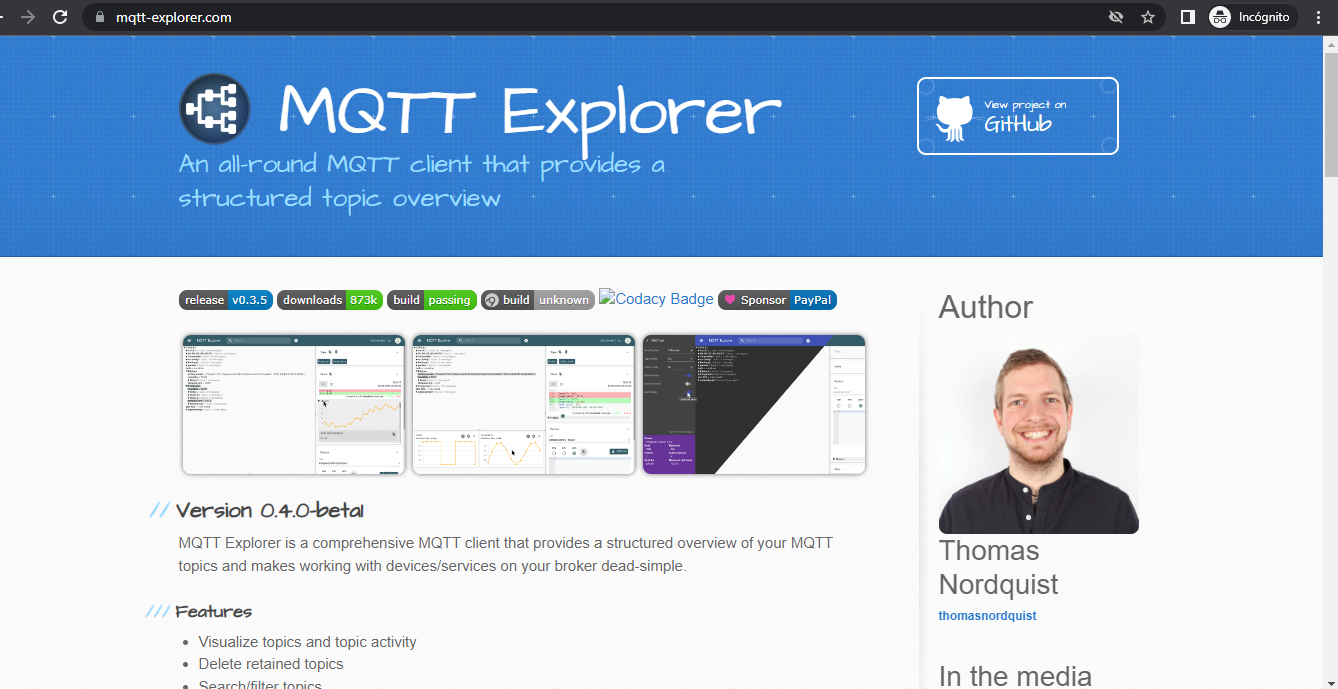


Figura 6 . Página de MQTT explorer.

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.

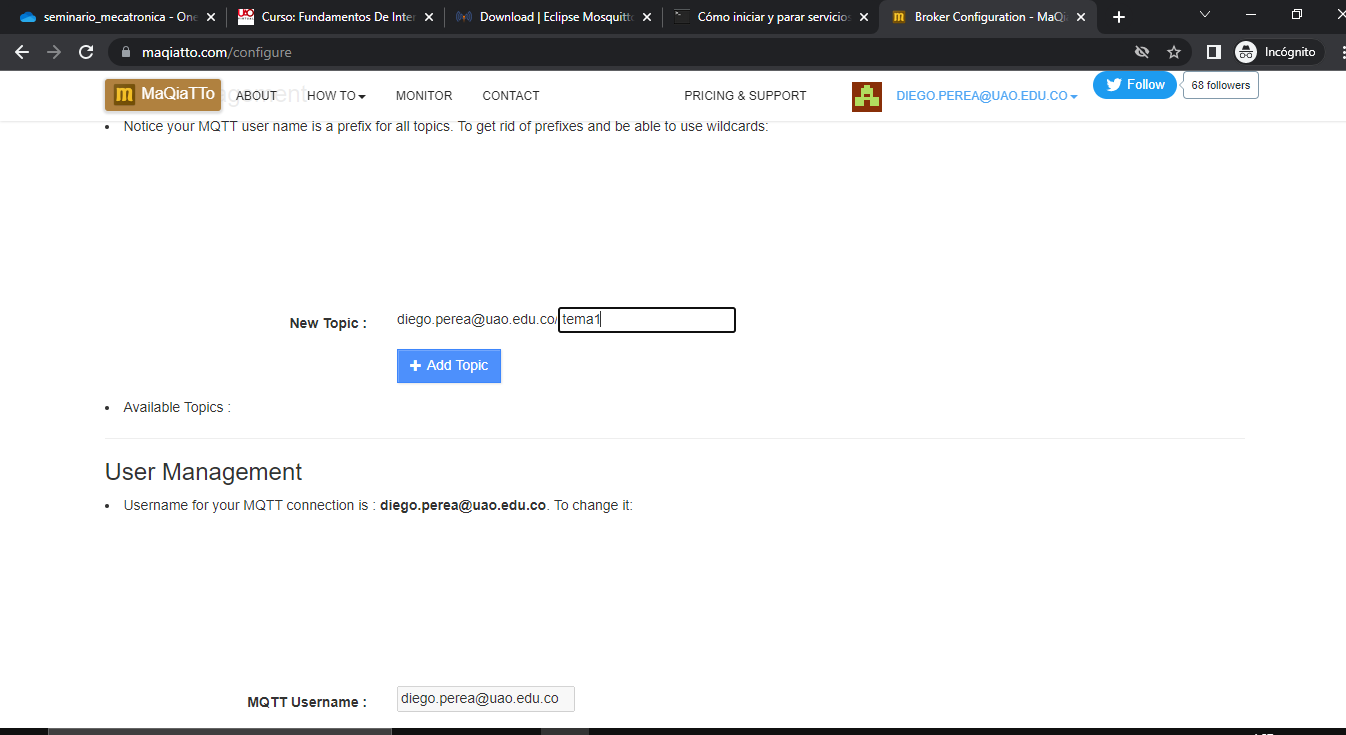


Figura 7. MQTT creación de tópico

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

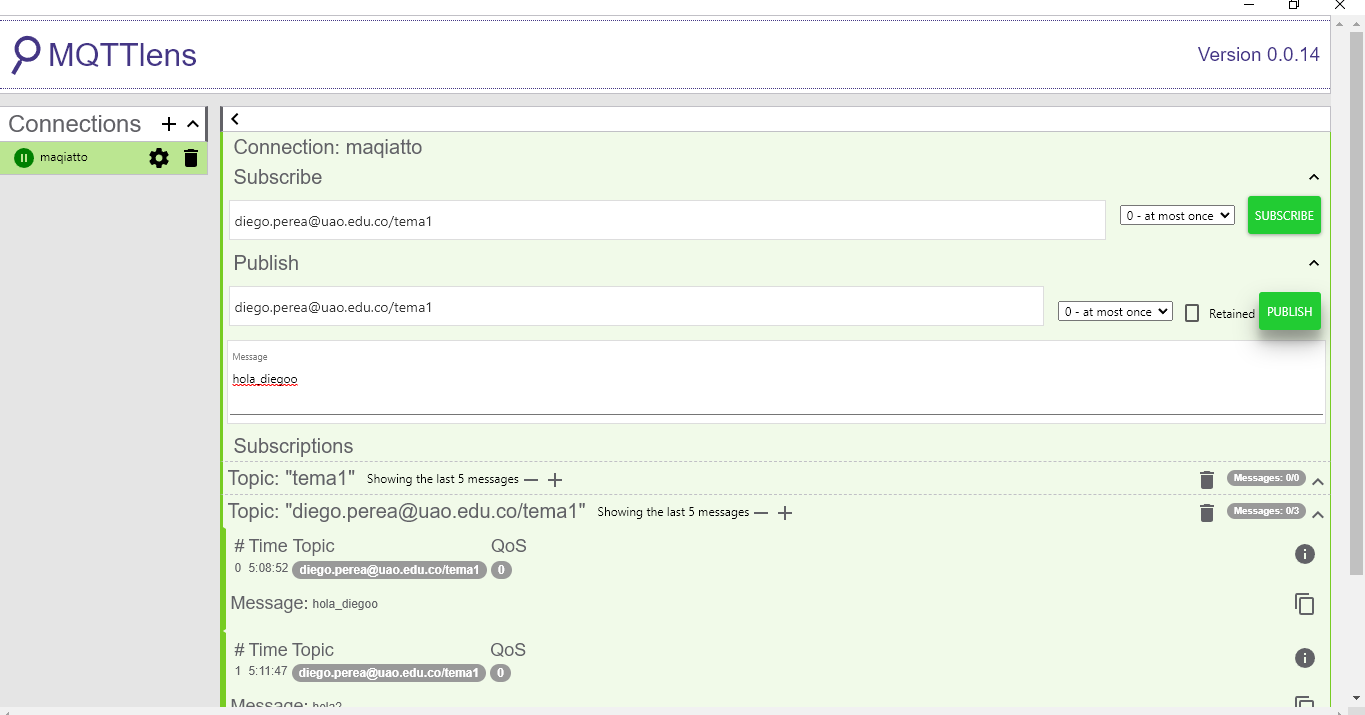


Figura 8. Extensión MQTT lens para la publicación del tópico.

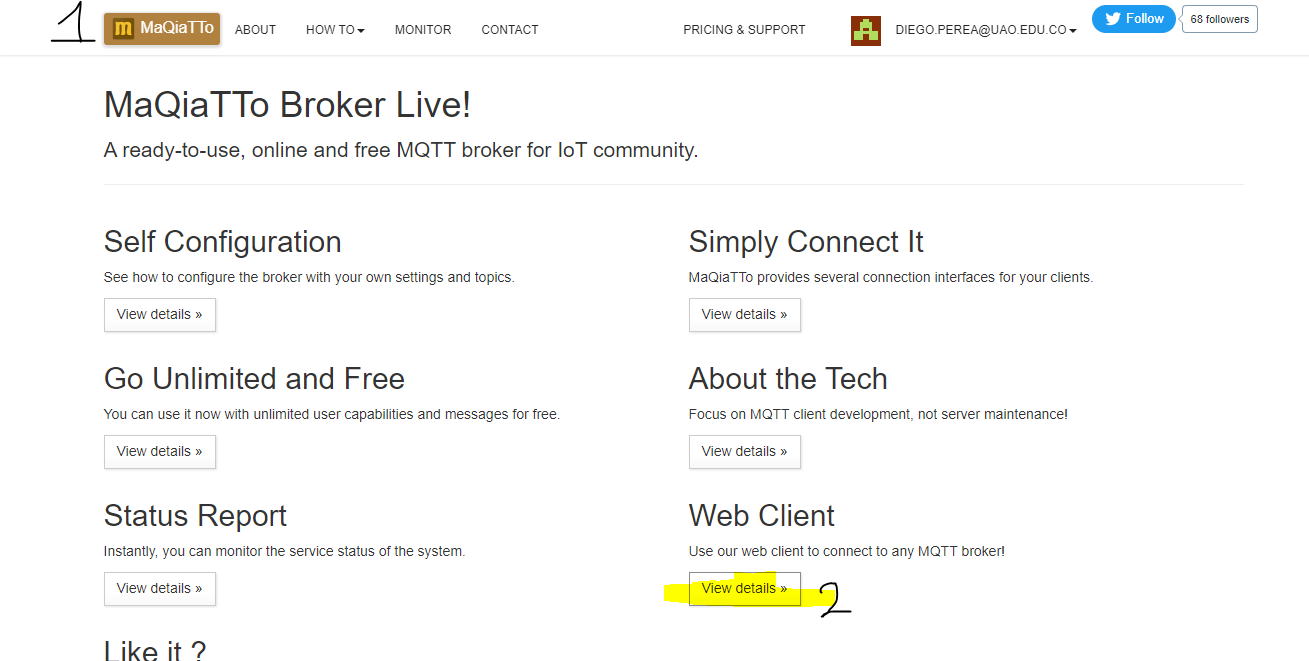


Figura 9. Maquiatto.com

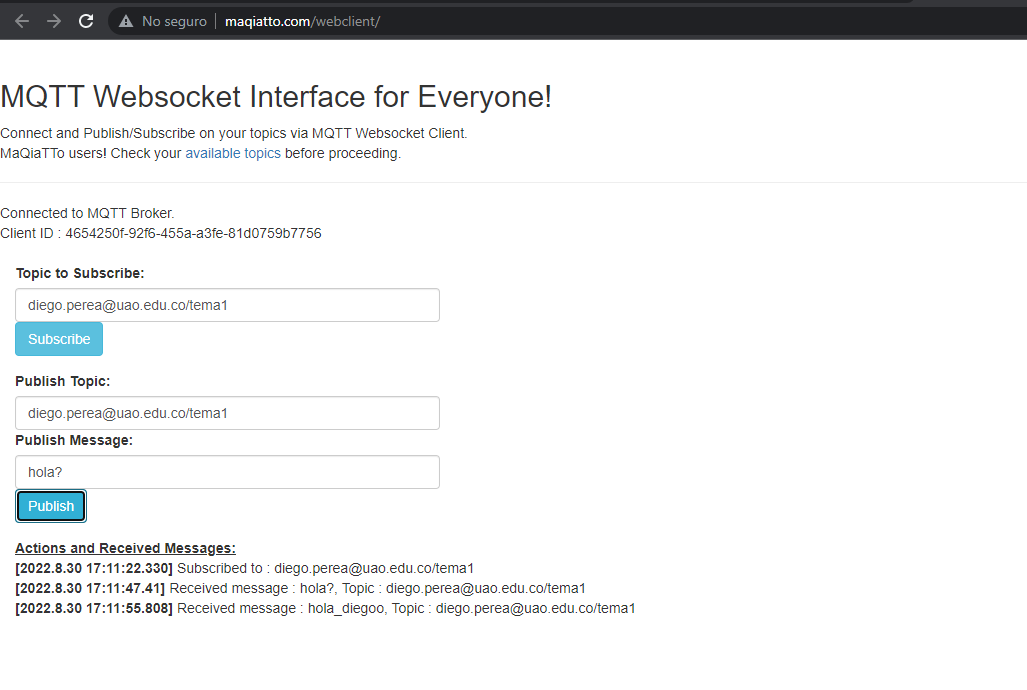


Figura 10. maquiatto web

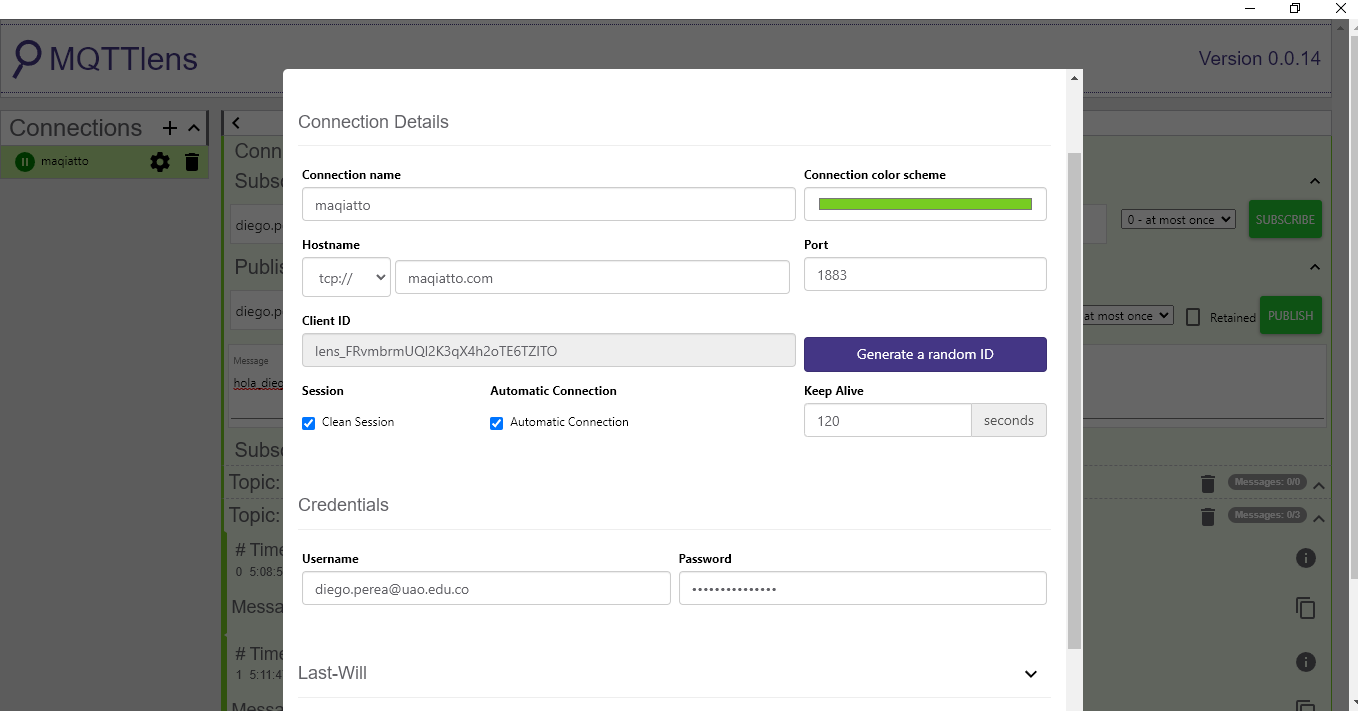


Figura 11. User y password en MQTT LENS

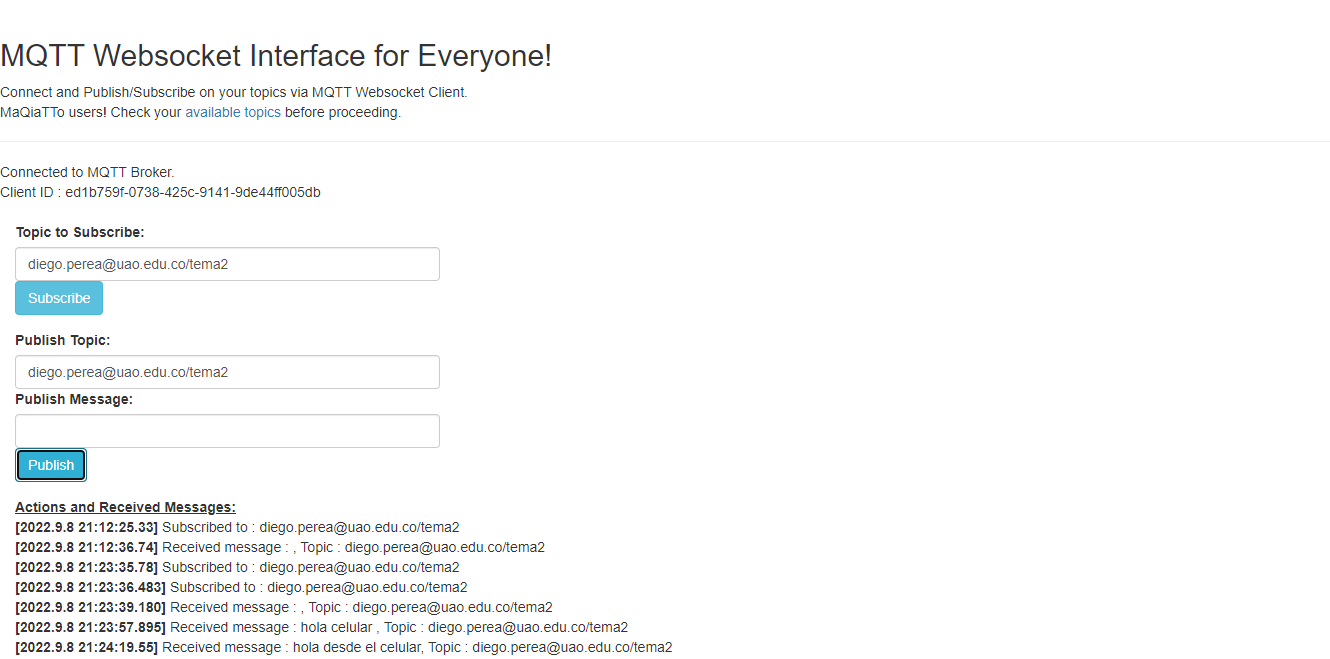
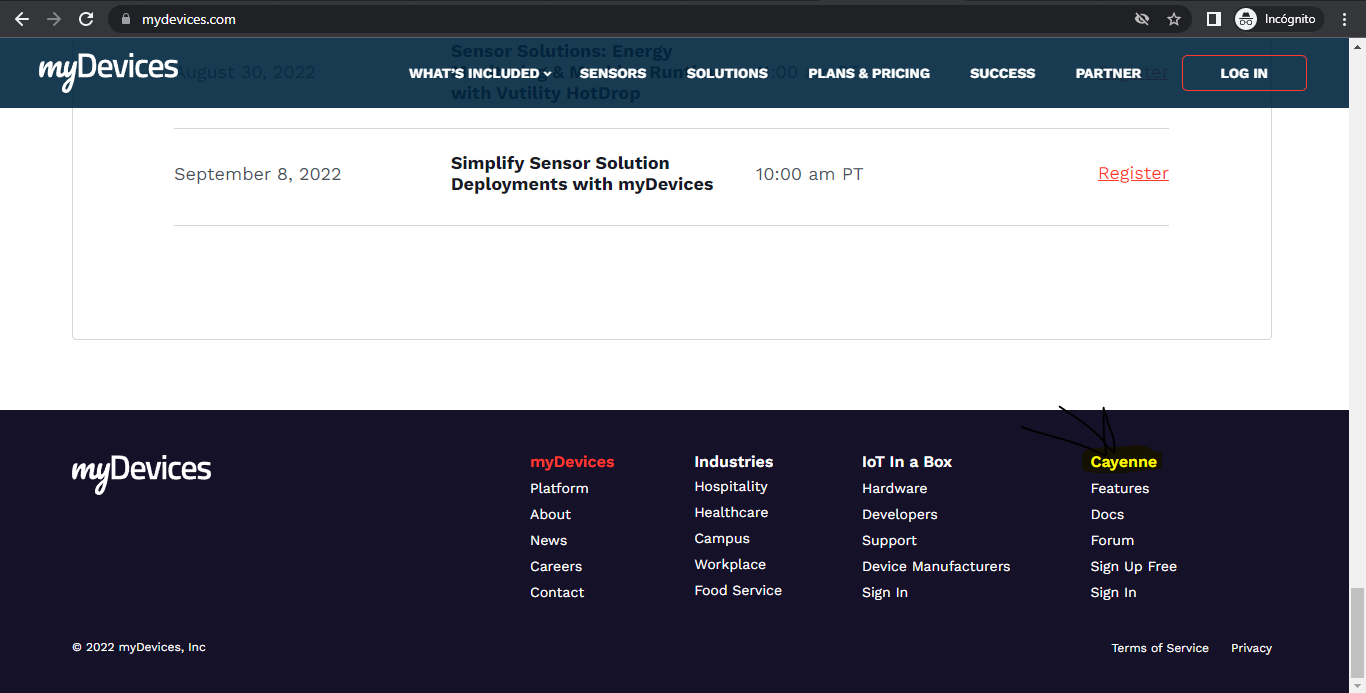


Figura 12. Conexión con Maqiatto con el celular

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

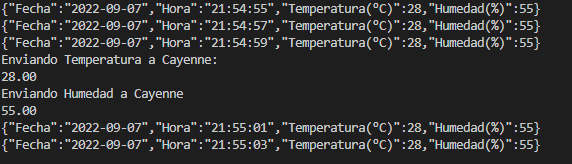


Figura 14. Envío de datos a cayenne

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

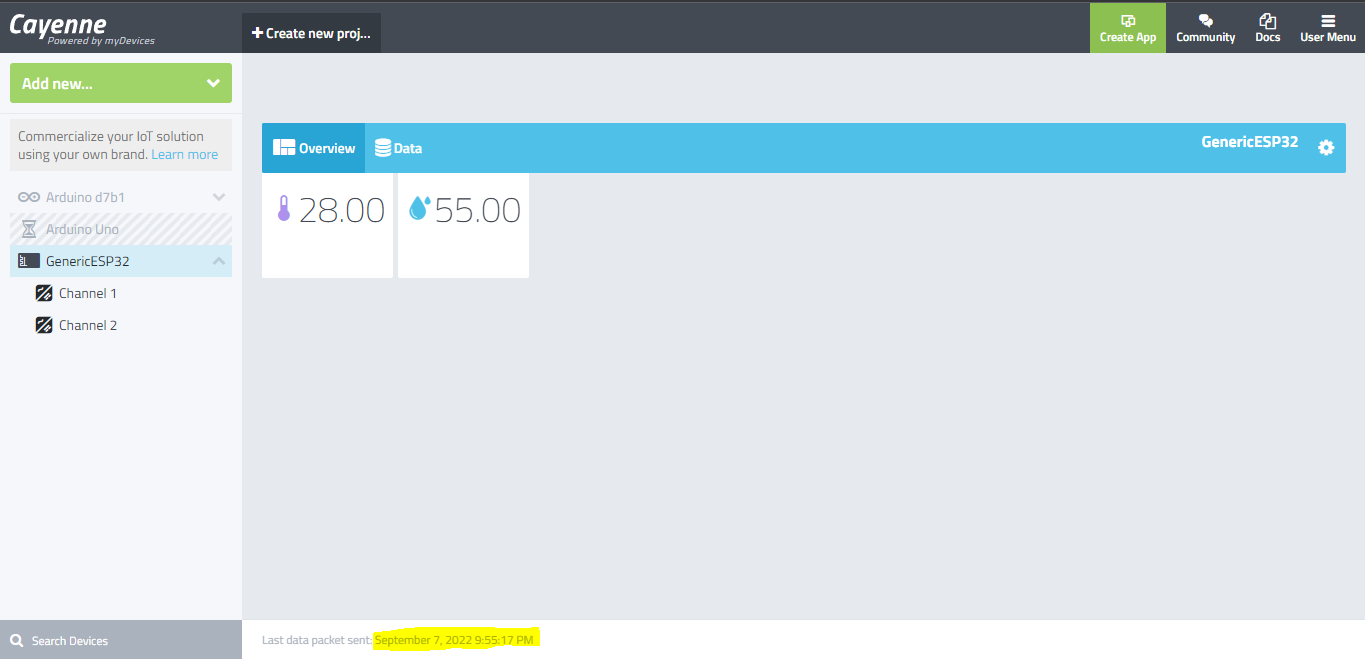


Figura 15. Datos obtenido de los sensores en ESP32

#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 cayene

#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();

}

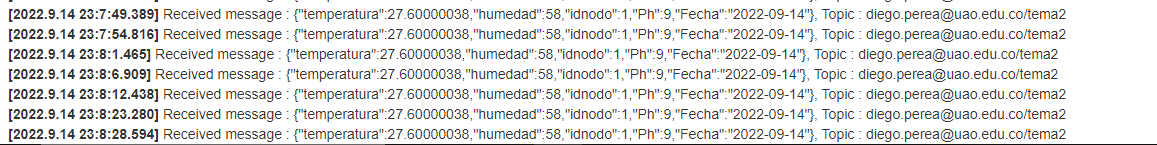
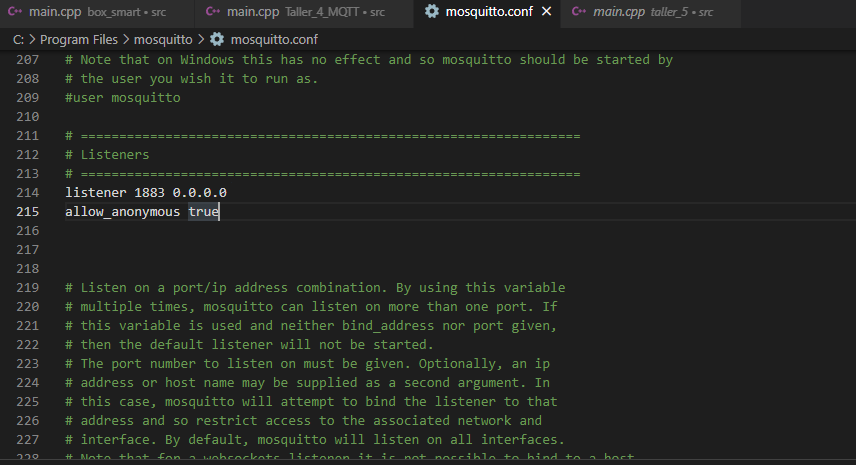
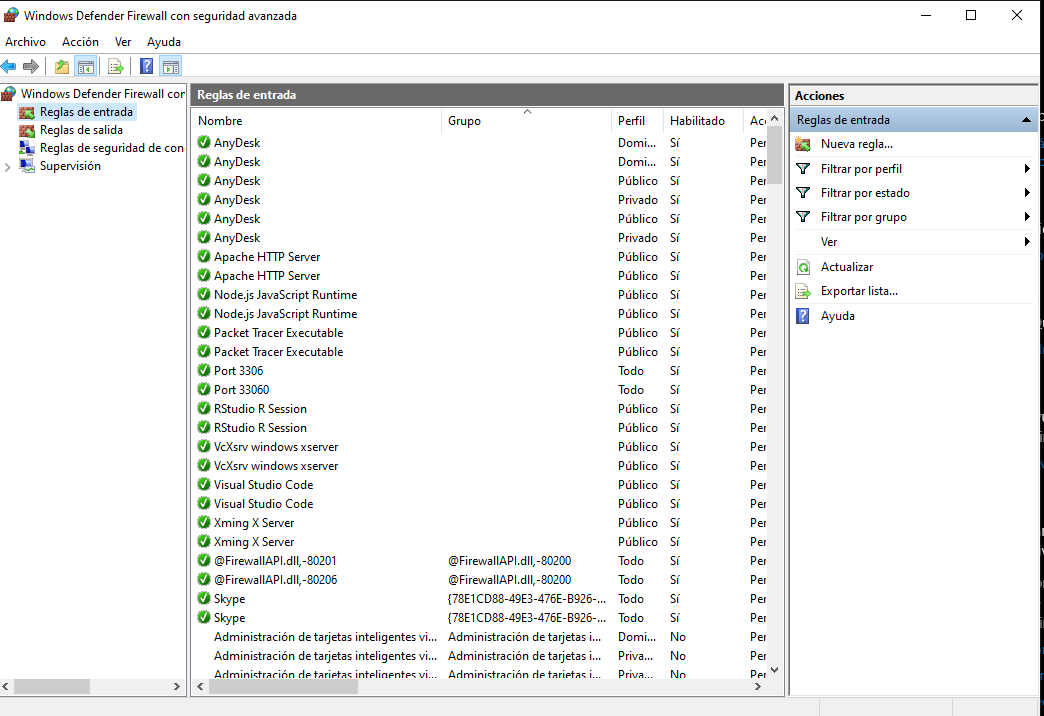
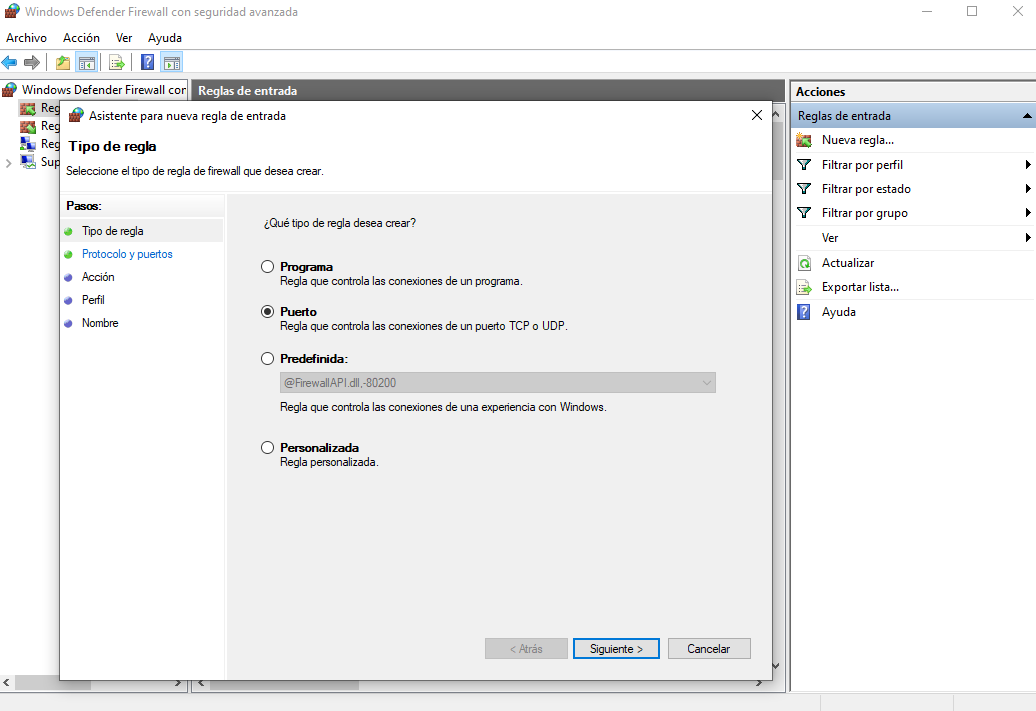


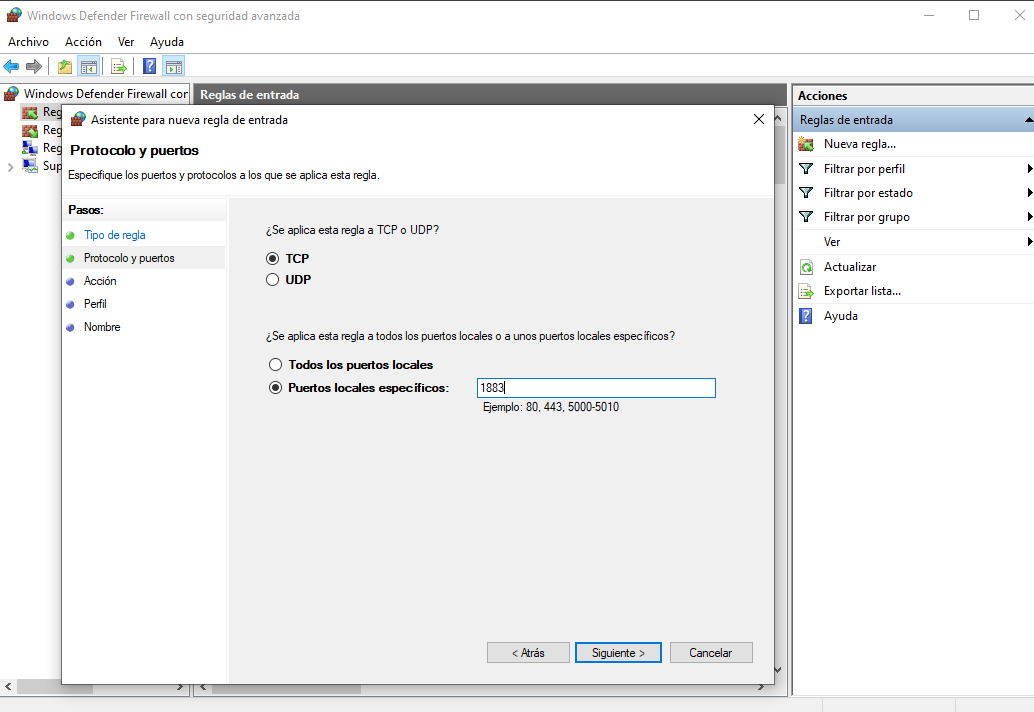
Figura 17. Recepción de datos a MQTT maqiatto

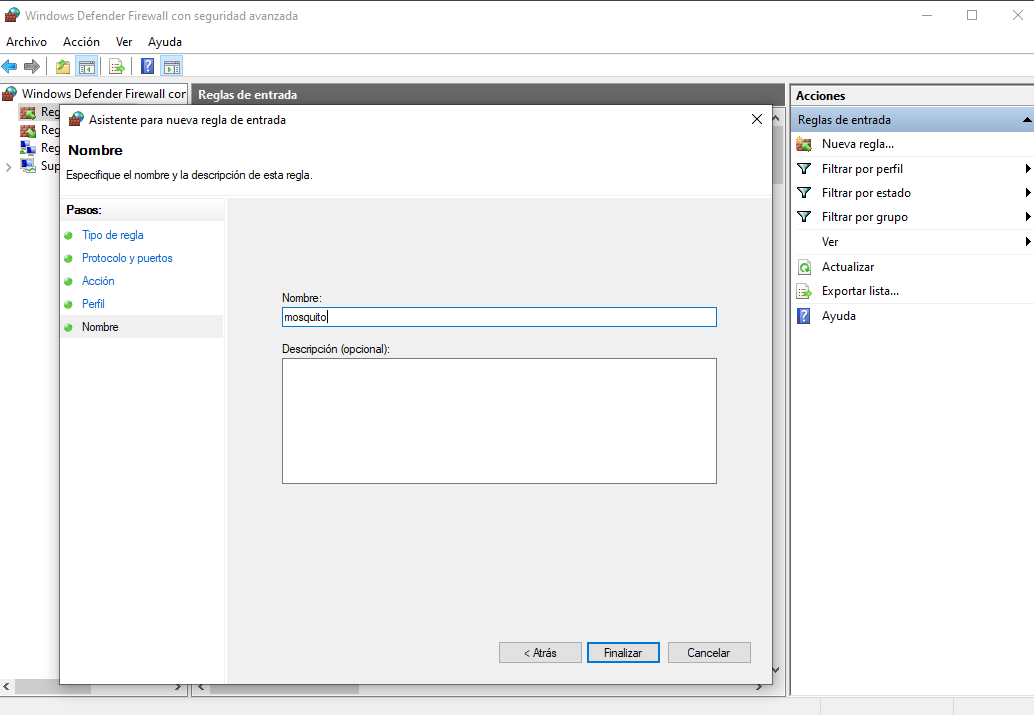
Se realiza la conexión Mosquitto:











Código de mosquitto

#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

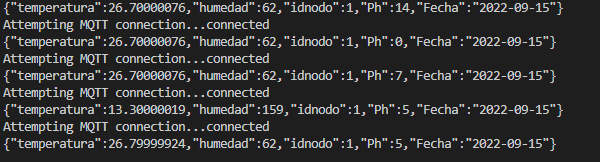


Figura 19. Envío de datos a mosquitto

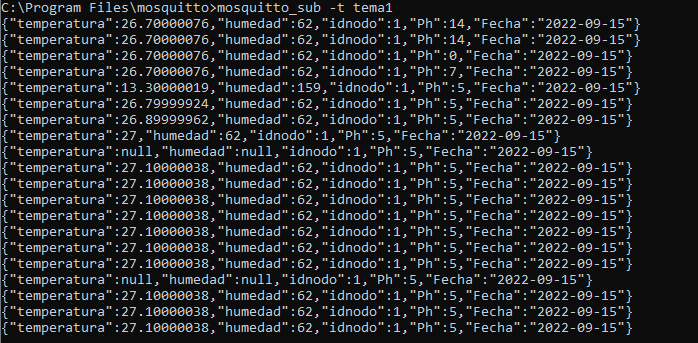


Figura 20. Visualización de datos en mosquitto

**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). [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]