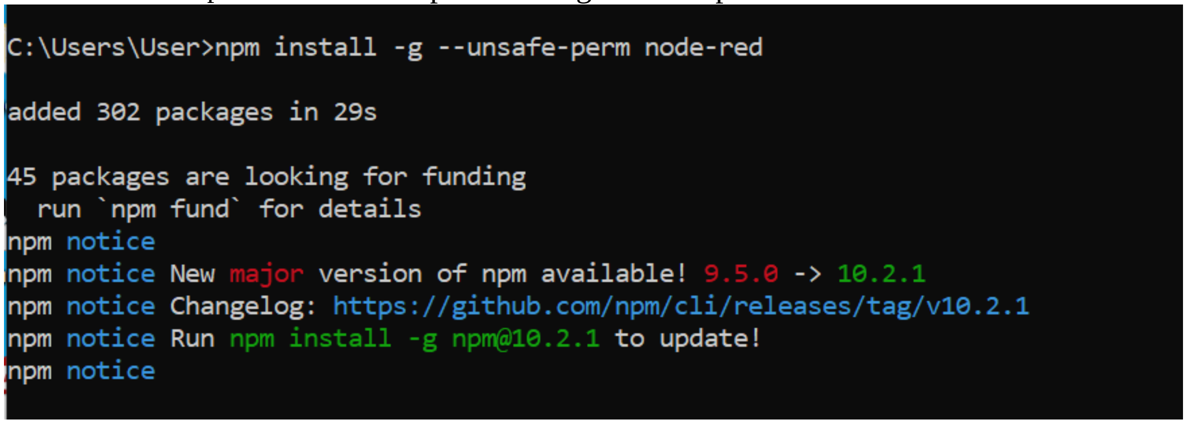
Actividad 4 IoT

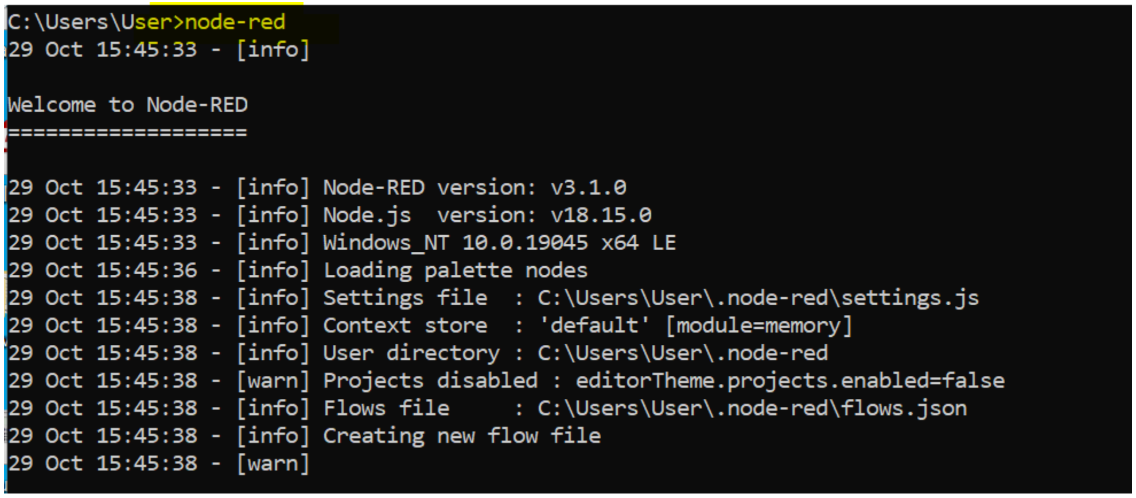
Diego Iván Perea Montealegre (2238513) diego.perea@uao.edu.co
Facultad de Ingeniería, Universidad Autónoma de Occidente
Cali, Valle del Cauca

Se aplica el comando npm install -g --unsafe-perm node-red en cmd :



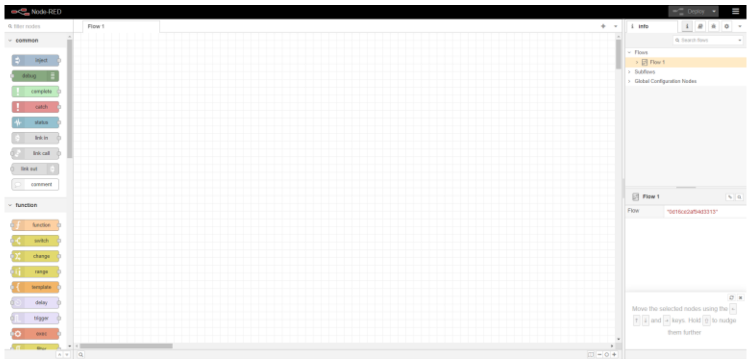
```
C:\Users\User>npm install -g --unsafe-perm node-red

added 302 packages in 29s

45 packages are looking for funding
  run `npm fund` for details
npm notice
npm notice New major version of npm available! 9.5.0 -> 10.2.1
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.2.1
npm notice Run npm install -g npm@10.2.1 to update!
npm notice
```

Se abre con el comando node-red :

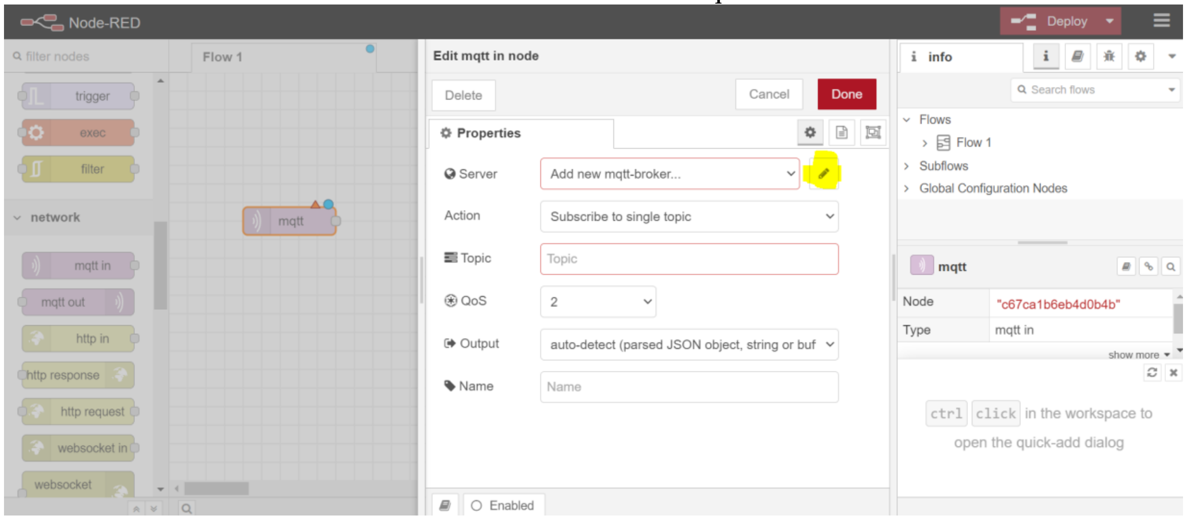


```
C:\Users\User>node-red
29 Oct 15:45:33 - [info]

Welcome to Node-RED
===================

29 Oct 15:45:33 - [info] Node-RED version: v3.1.0
29 Oct 15:45:33 - [info] Node.js  version: v18.15.0
29 Oct 15:45:33 - [info] Windows_NT 10.0.19045 x64 LE
29 Oct 15:45:36 - [info] Loading palette nodes
29 Oct 15:45:38 - [info] Settings file  : C:\Users\User\.node-red\settings.js
29 Oct 15:45:38 - [info] Context store  : 'default' [module=memory]
29 Oct 15:45:38 - [info] User directory : C:\Users\User\.node-red
29 Oct 15:45:38 - [warn] Projects disabled : editorTheme.projects.enabled=false
29 Oct 15:45:38 - [info] Flows file     : C:\Users\User\.node-red\flows.json
29 Oct 15:45:38 - [info] Creating new flow file
29 Oct 15:45:38 - [warn]
```

Y dnando la dirección de localhost:1880 se visualiza node red:

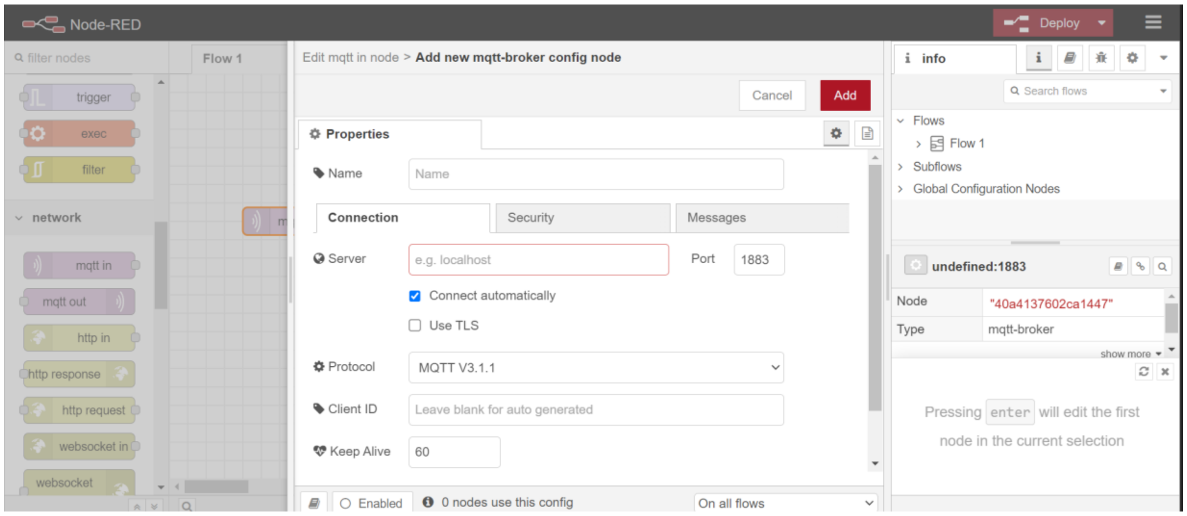


Configurar MQTT
dar en editar en el mqtt


Edit mqtt in node

Server: Add new mqtt-broker...
Action: Subscribe to single topic
Topic: Topic
QoS: 2
Output: auto-detect (parsed JSON object, string or buf
Name: Name

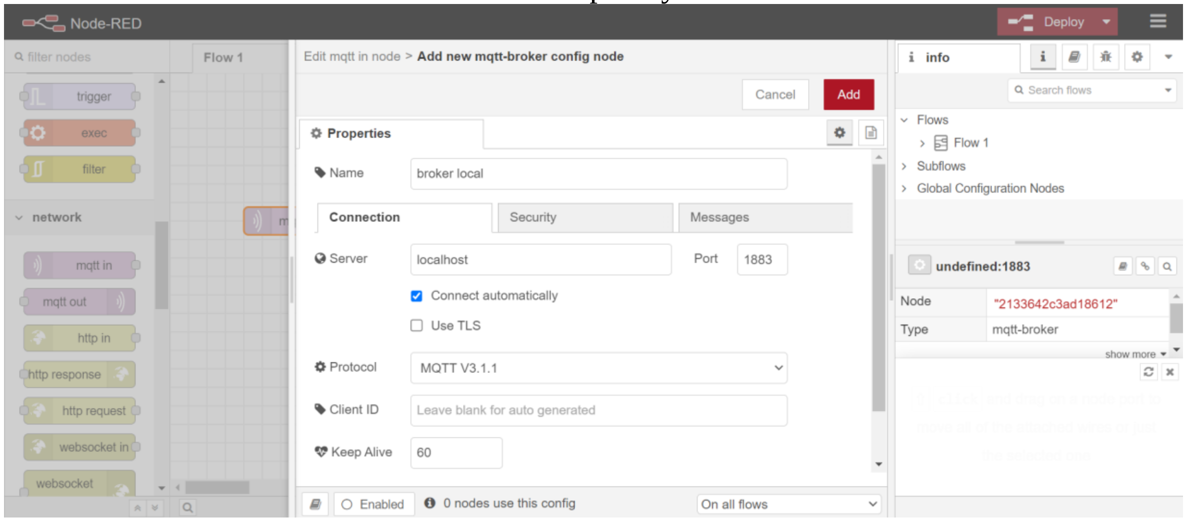
Node: "c67ca1b6eb4d0b4b"
Type: mqtt in


Edit mqtt in node > Add new mqtt-broker config node

Name: Name
Connection | Security | Messages
Server: e.g. localhost — Port: 1883
☑ Connect automatically
☐ Use TLS
Protocol: MQTT V3.1.1
Client ID: Leave blank for auto generated
Keep Alive: 60

Node: "40a4137602ca1447"
Type: mqtt-broker

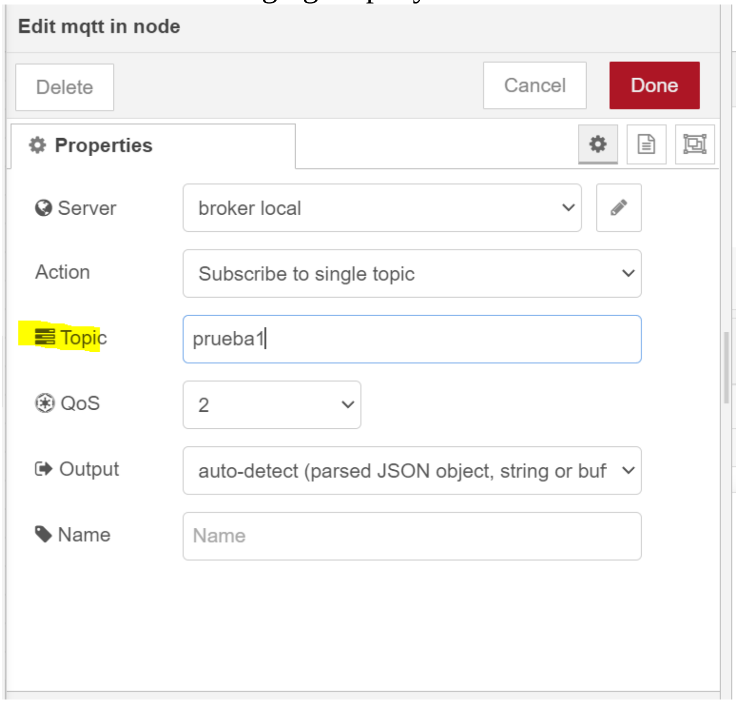
Si se usa local se pone y se da Add:


Edit mqtt in node > Add new mqtt-broker config node

Name: broker local
Connection | Security | Messages
Server: localhost — Port: 1883
☑ Connect automatically
☐ Use TLS
Protocol: MQTT V3.1.1
Client ID: Leave blank for auto generated
Keep Alive: 60

Node: "2133642c3ad18612"
Type: mqtt-broker

Se agrega topic y Done :



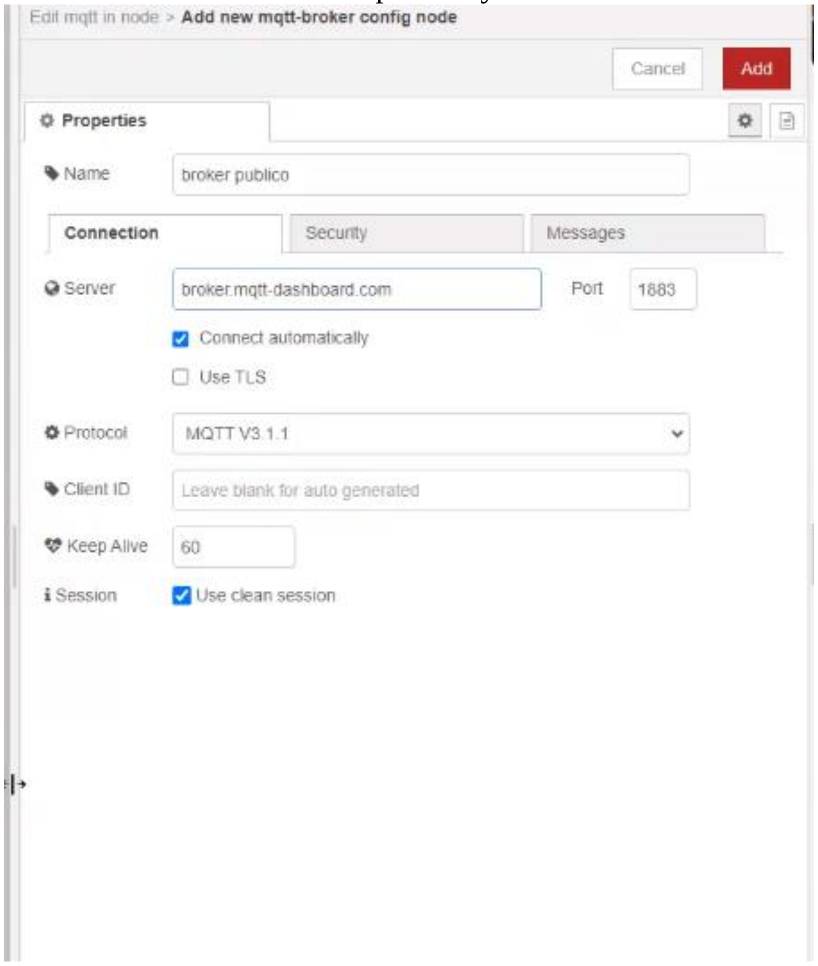
Edit mqtt in node

Delete | Cancel | Done

Properties

- Server: broker local
- Action: Subscribe to single topic
- Topic: prueba1
- QoS: 2
- Output: auto-detect (parsed JSON object, string or buf
- Name: Name

Pero si se tiene publico y se da Add:



Edit mqtt in node > Add new mqtt-broker config node

Cancel | Add

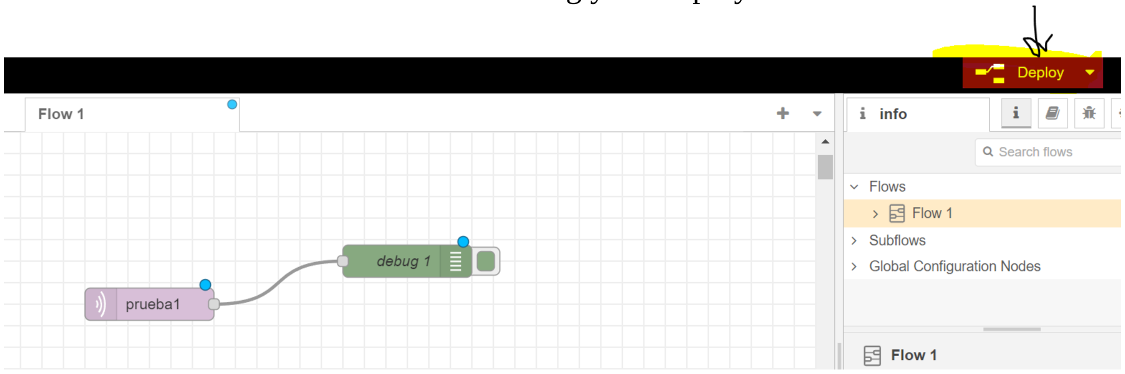
Properties

- Name: broker publico

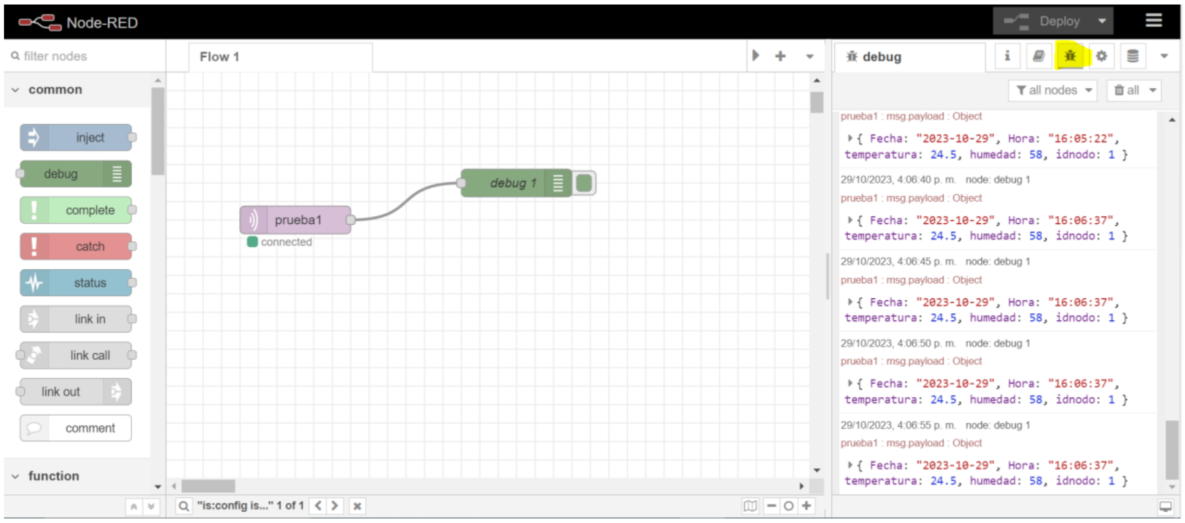
Connection | Security | Messages

- Server: broker.mqtt-dashboard.com
- Port: 1883
- [✓] Connect automatically
- [ ] Use TLS
- Protocol: MQTT V3.1.1
- Client ID: Leave blank for auto generated
- Keep Alive: 60
- Session: [✓] Use clean session

Se conecta debug y dar deploy :



Dar en debug y ejecutar código de ESP32



```
prueba1 : msg.payload : Object
▸{ Fecha: "2023-10-29", Hora: "16:05:22",
temperatura: 24.5, humedad: 58, idnodo: 1 }
29/10/2023, 4:06:40 p. m.   node: debug 1
prueba1 : msg.payload : Object
▸{ Fecha: "2023-10-29", Hora: "16:06:37",
temperatura: 24.5, humedad: 58, idnodo: 1 }
29/10/2023, 4:06:45 p. m.   node: debug 1
prueba1 : msg.payload : Object
▸{ Fecha: "2023-10-29", Hora: "16:06:37",
temperatura: 24.5, humedad: 58, idnodo: 1 }
29/10/2023, 4:06:50 p. m.   node: debug 1
prueba1 : msg.payload : Object
▸{ Fecha: "2023-10-29", Hora: "16:06:37",
temperatura: 24.5, humedad: 58, idnodo: 1 }
29/10/2023, 4:06:55 p. m.   node: debug 1
prueba1 : msg.payload : Object
▸{ Fecha: "2023-10-29", Hora: "16:06:37",
temperatura: 24.5, humedad: 58, idnodo: 1 }
```

Codigo ESP32 Platformio:

```
#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 14    // 4 = PIN D4
```

```
#define DHTTYPE    DHT11
DHT dht(DHTPIN, DHTTYPE);


// 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 = "*PASSWORD_WIFI*"; // clave de wifi
char mqttBroker[] = "192.168.**.*"; //ip del servidor

char mqttClientId[] = "prueba1"; //cualquier nombre
char inTopic[] = "prueba1";//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();

  String variable;
  StaticJsonDocument<256> doc;

  doc["Fecha"] = dayStamp;
  doc["Hora"] = timeStamp;
  doc["temperatura"] = t;
  doc["humedad"] = h;
  doc["idnodo"] = 1;



  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();
}
```