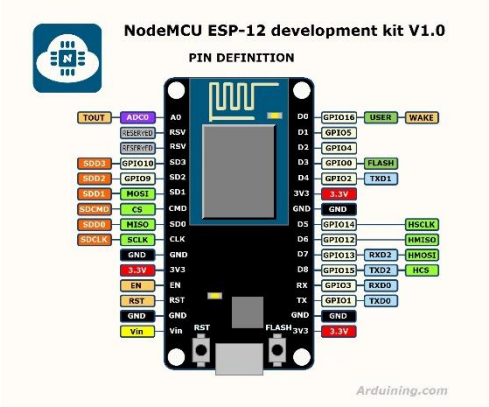
ESP-12E NodeMcu Lua WiFi Development Board



NodeMcu Lua **ESP-12E** WIFI Development Board
Wireless **802.11 b/g/n standard**
Support STA / AP / STA + AP three operating modes
Built-in **TCP / IP protocol** stack to support multiple TCP Client connections (5 MAX)
D0 ~ D8, SD1 ~ SD3: used as GPIO, PWM, IIC, etc., port driver capability 15mA
AD0: 1 channel ADC
Input**: 4.5V ~ 9V** (10VMAX), USB-powered
Current: continuous transmission: ≈70mA (200mA MAX), Standby: <200uA
Transfer rate:110-460800bps
Support UART / **GPIO** data communication interface
Remote firmware upgrade (OTA)
Support Smart Link Smart Networking
Working temperature: -40 ℃ ~ + 125 ℃
Drive Type: Dual high-power H-bridge driver

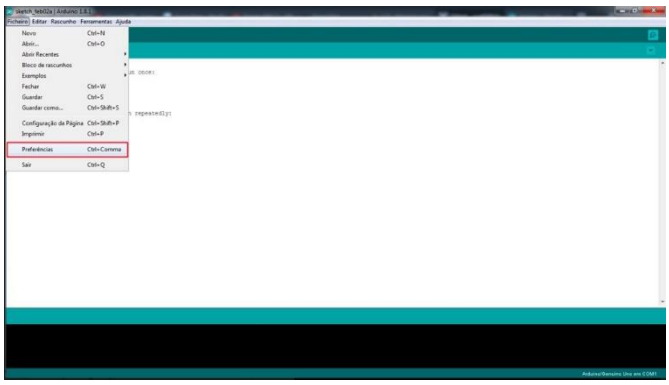
1. INSTALAÇÃO Placa ESP-12E NodeNCU no Arduíno IDE

a. Configuração do lick para a placa ESP

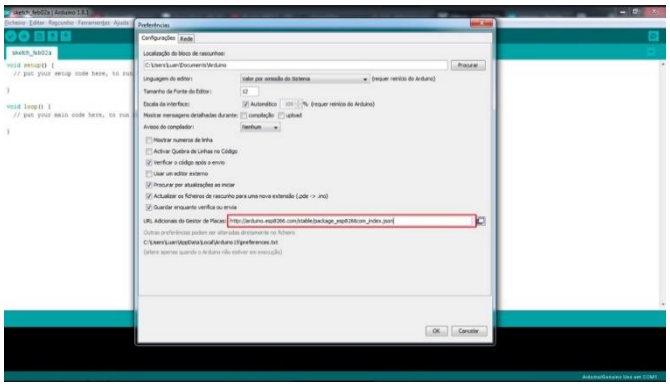
Link a ser colocado no IDE do Arduíno (https://github.com/esp8266/Arduino):

https://arduino.esp8266.com/stable/package_esp8266com_index.json

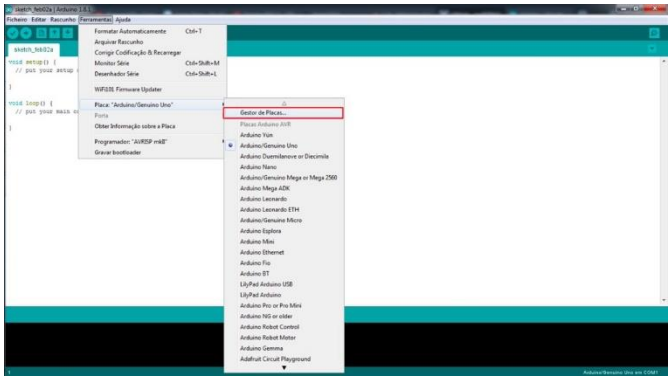
1: Preferências



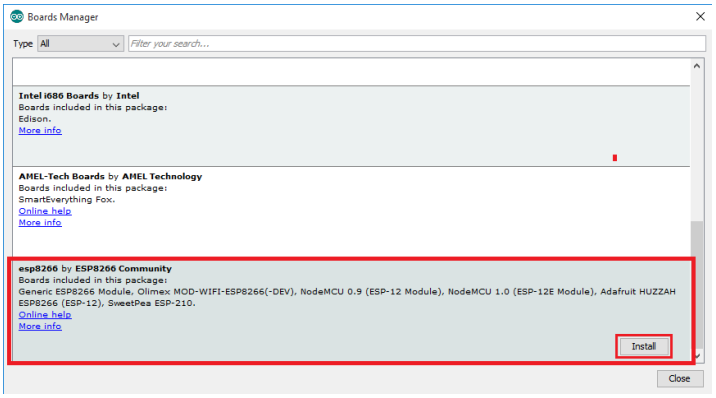
2: Copy e Cole LINK



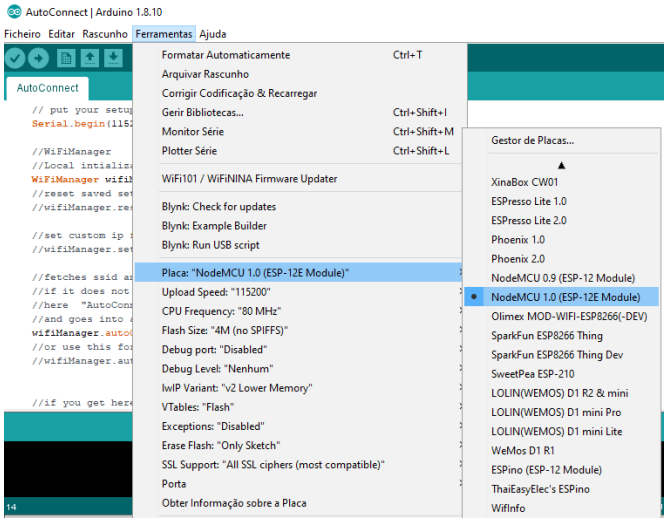
3: Gestão de placas



4: instalar ESP8266 community

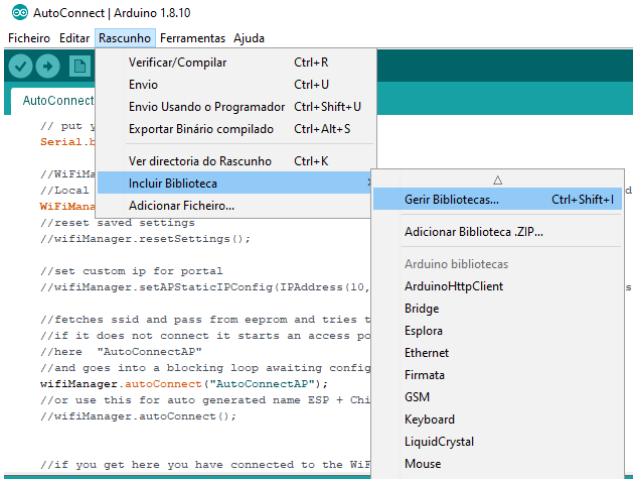


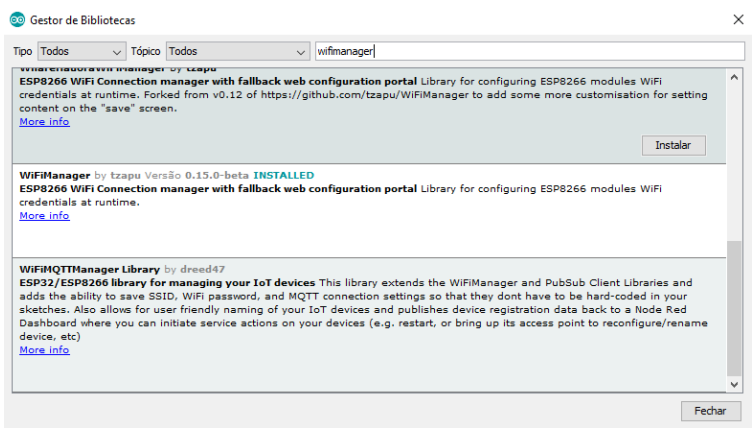
5: Escolher “**NodeMCU 1.0 ESP-12E**”



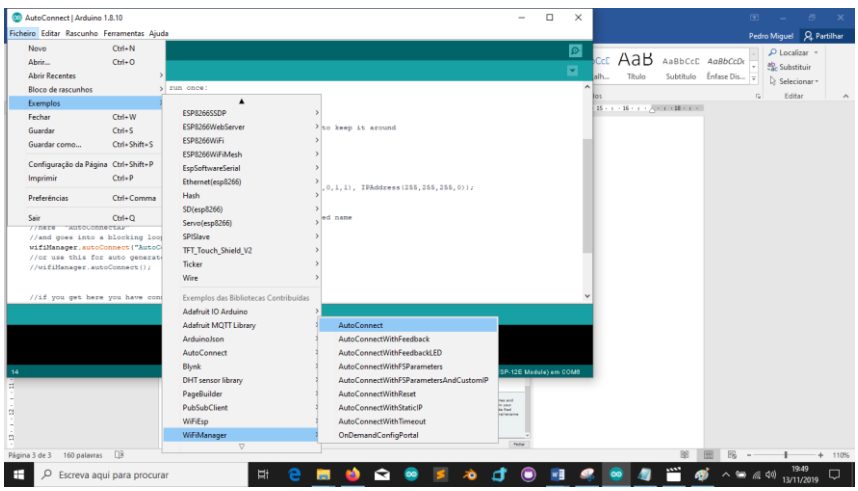
b. Biblioteca WiFi ESP-12E

Adicionar biblioteca “**WiFiManager**”





EXEMPLOS:

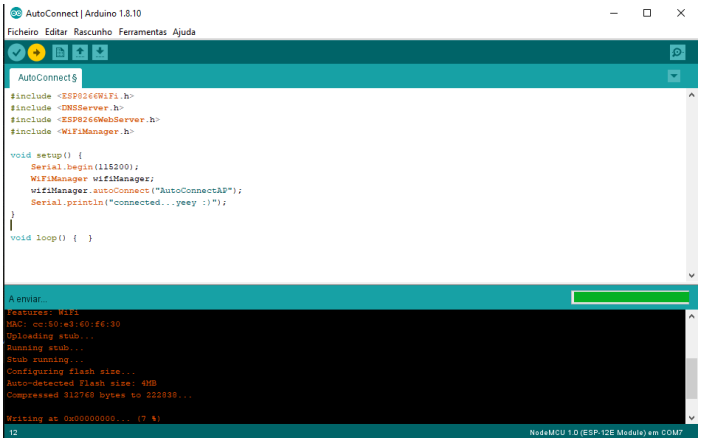


2. Testar o 1º Exemplo – AutoConnect

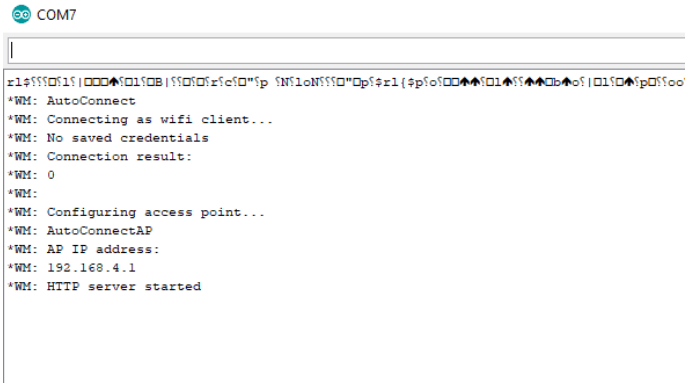
a) Limpar o código do exemplo

```
#include <ESP8266WiFi.h>
#include <DNSServer.h>
#include <ESP8266WebServer.h>
#include <WiFiManager.h>

void setup() {
   Serial.begin(115200);
   WiFiManager wifiManager;
   wifiManager.autoConnect("AutoConnectAP");
   Serial.println("connected...yeey :)");
}

void loop() { }
```

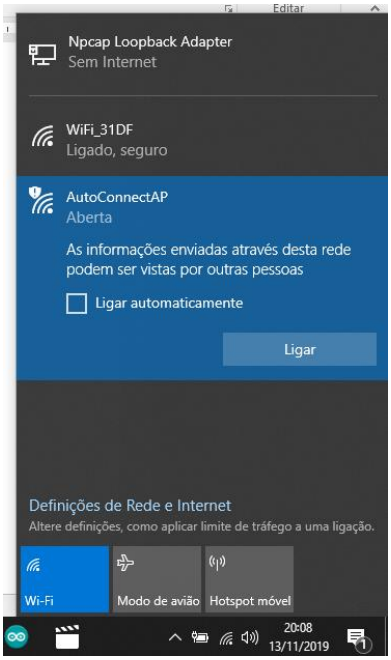
b) Enviar para NodeMCU ESP-12E



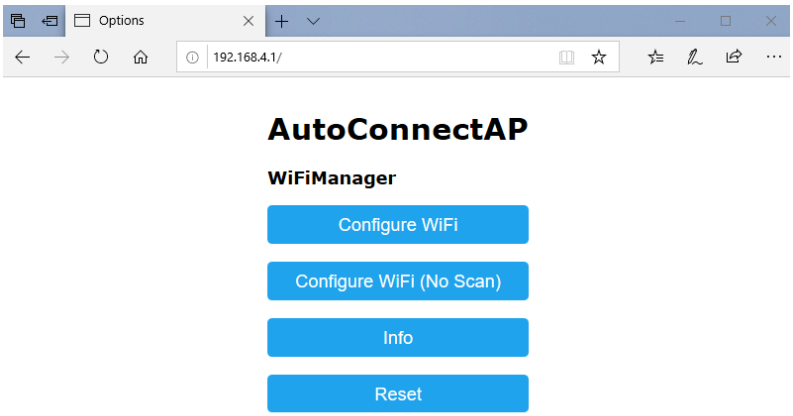
c) Verificar se gravou e o ip atribuído pelo ESP-12E



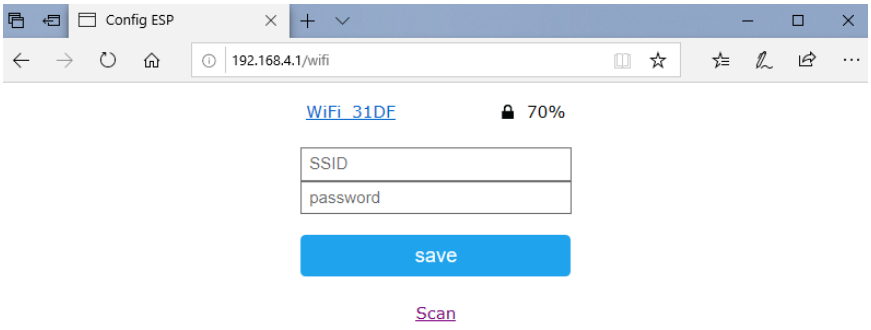
d) Escolher e ligar a rede SSID dado pelo ESP-12E



e) Navegador colocar ip (192.168.4.1)

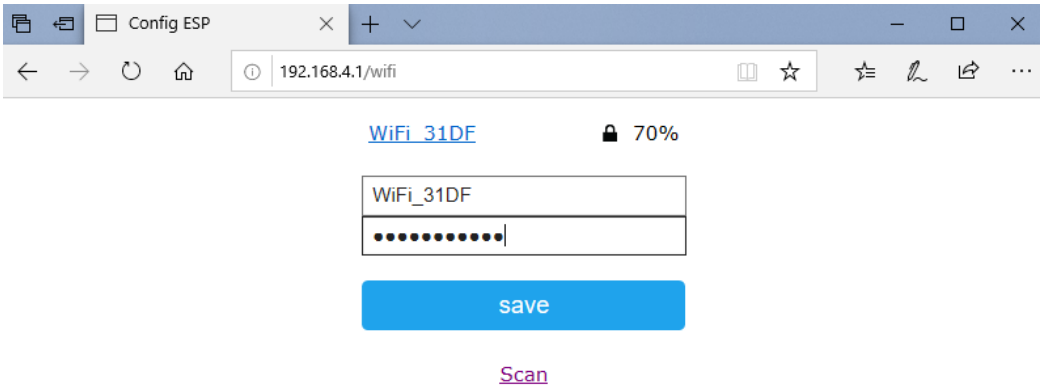


f) Escolher Configurar WiFi



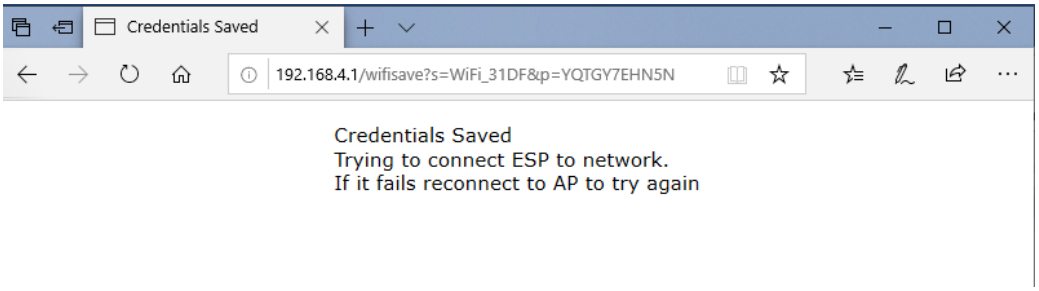
g) Colocar SSID e a Password da rede pretendida.

- SSID - Password - Clicar SAVE



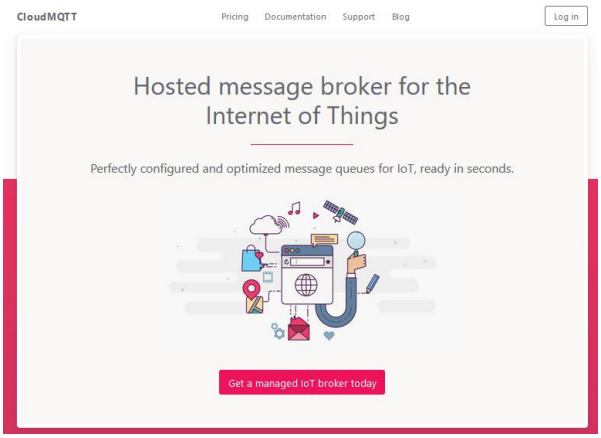
h) FIM da Configuração de rede

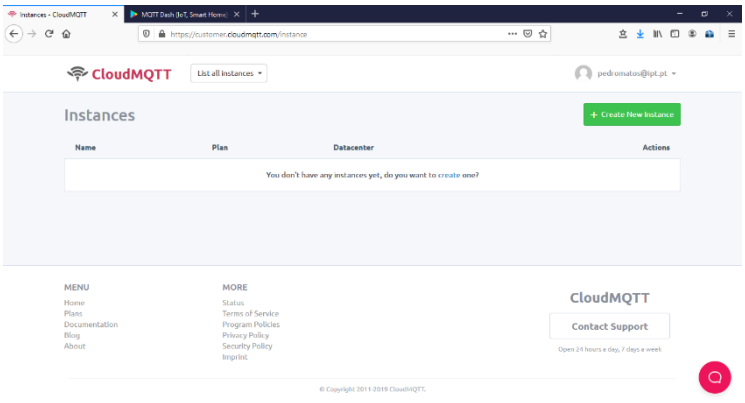
- Desligar e tornar a ligar, está gravada na Flash do ESP-12E



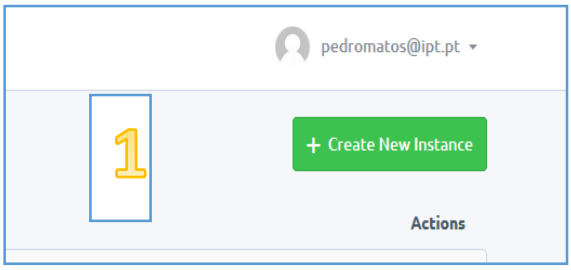
3. Testar o 2º Exemplo – MQTT (PubSubClient)

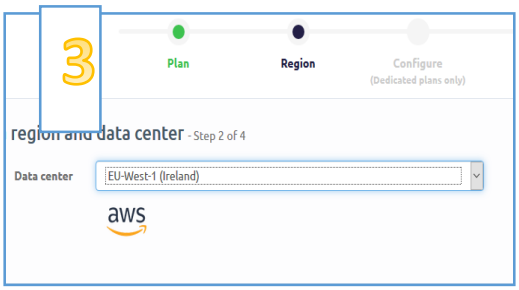
a) Criar conta CloudMQTT

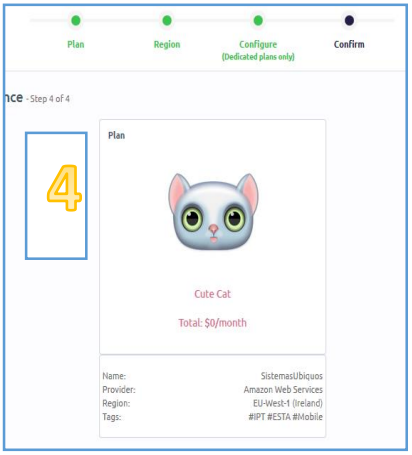




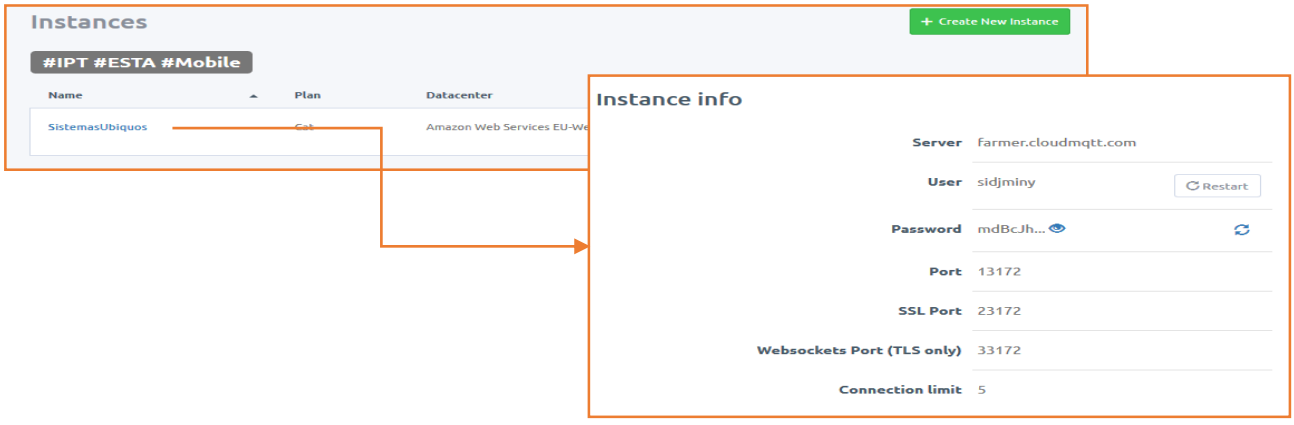
- **Criar um projeto (instance)**



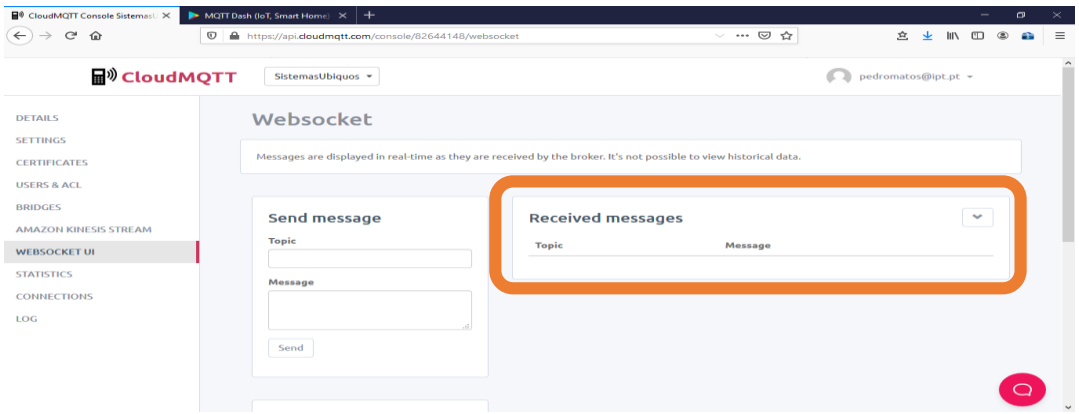




- **Dados da Autenticação**



- Dados inseridos com o MQTT ESP-12E



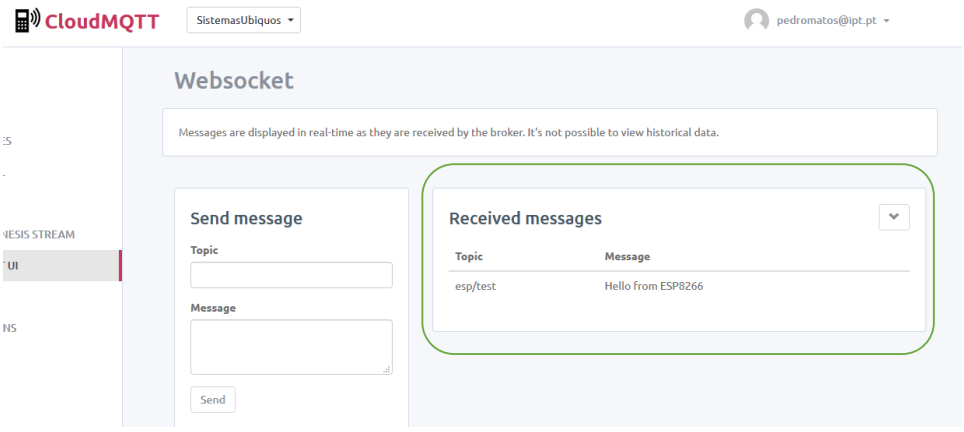
b) *Programar NodeMCU ESP-12E em Arduino IDE*

```
/*
 *  Sistemas Ubiquos
 *  MQTT - ESP-12E NodeCMU
 *  Pedro Matos
 */
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#define LED 2
// WiFi credenciais
const char* ssid = "WiFi_31DF";
const char* password =  "************";

// MQTT server configurações
const char* mqttServer = "farmer.cloudmqtt.com";
const int mqttPort = 13172;
const char* mqttUser = "sidjminy";
const char* mqttPassword = "mdBcJhZuHOgR";

WiFiClient espClient;
PubSubClient client(espClient);
//Inicio – carrega uma vez
void setup() {
  delay(1000);
  pinMode(LED,OUTPUT);
  Serial.begin(115200);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println("Connecting to WiFi..");
  }
  Serial.print("Connected to WiFi :");
  Serial.println(WiFi.SSID());
 //Ligação - CloudMQTT
  client.setServer(mqttServer, mqttPort);
  client.setCallback(MQTTcallback);

  while (!client.connected()) {
    Serial.println("Connecting to MQTT...");
//… Autenticação no MQTT
    if (client.connect("ESP8266", mqttUser, mqttPassword )) {

      Serial.println("connected");

    } else {
      Serial.print("failed with state ");
```

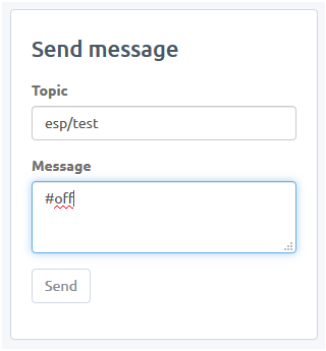
```
      Serial.println(client.state());
      delay(2000);
    }
  }
 //MQTT - enviar só uma vez no arranque do dispositivo (teste)
  client.publish("esp/test", "Hello from ESP8266");
  client.subscribe("esp/test");
} //……..fim setup

 //---------------------------- Receção da mensagem vinda MQTT
void MQTTcallback(char* topic, byte* payload, unsigned int length) {

  Serial.print("Message arrived in topic: ");
  Serial.println(topic);
  Serial.print("Message:");

  String message;
  for (int i = 0; i < length; i++) {
    message = message + (char)payload[i];  //Conver *byte to String
  }
   Serial.print(message);
  if(message == "#on") {digitalWrite(LED,LOW);}   //LED on
  if(message == "#off") {digitalWrite(LED,HIGH);} //LED off
  Serial.println();
  Serial.println("-----------------------");
}
 //----- faz várias vezes, até ser desligado.
void loop() {
  client.loop();
}
```

c) Verificar os envios na CloulMQTT

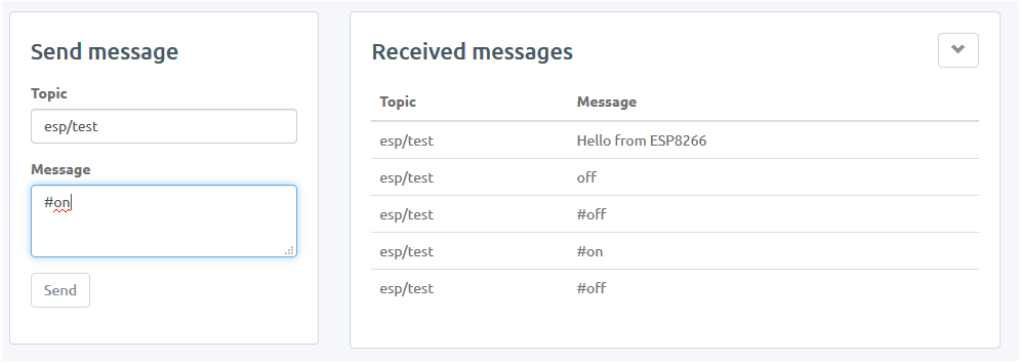
Recepção do ESP-12E para CloudMQTT



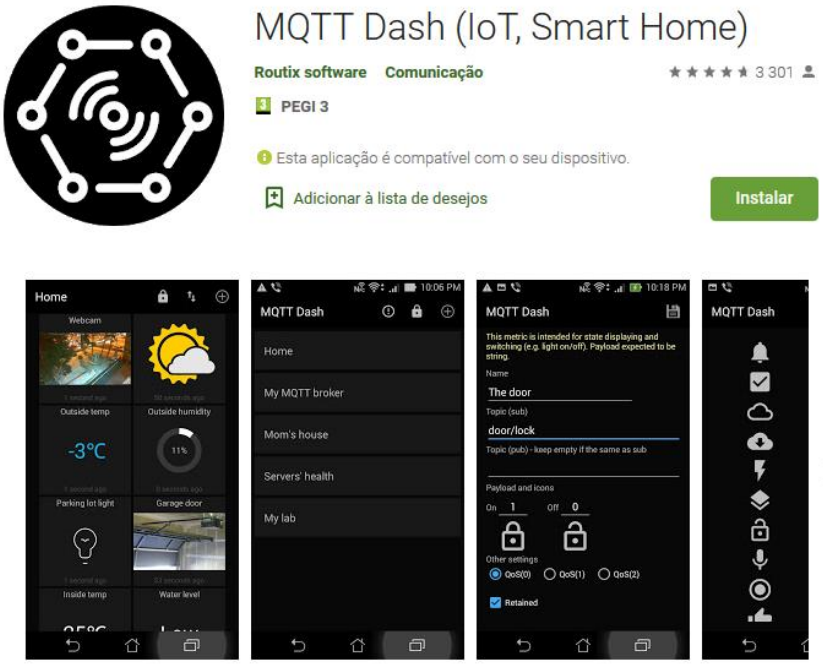
Enviou do MQTT para ESP-12E (desliga o led)



Enviou do MQTT para ESP-12E (liga o Led)

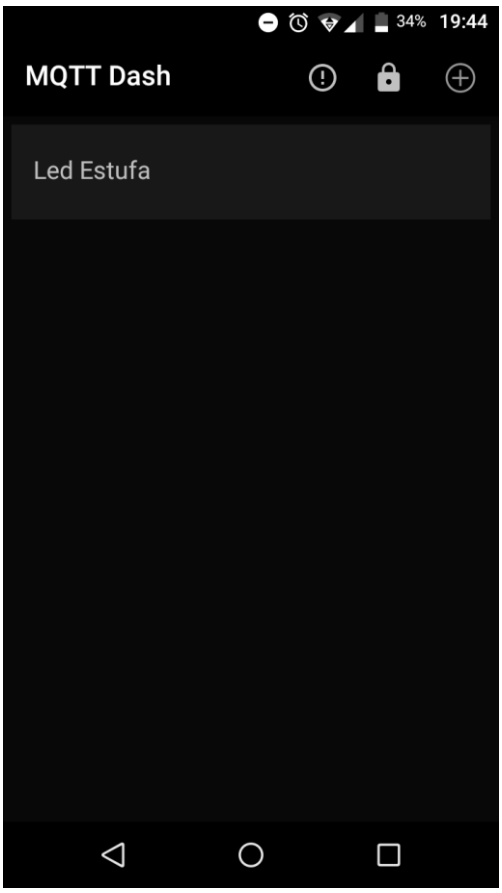


d) Cliente Mobile

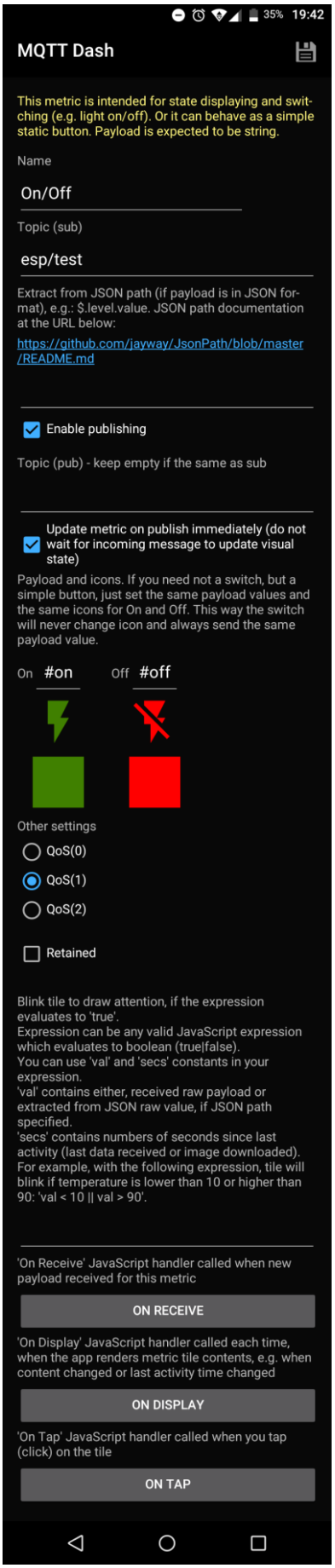


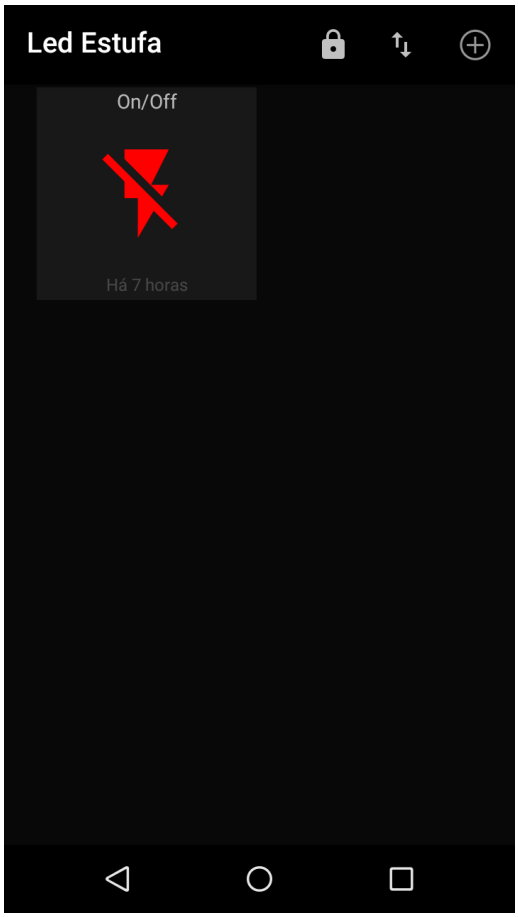
- Configurar com os dados de autenticação CloudMQTT

Criar Ligação com MQTT Cloud



Criar o Botão de comunicação com MQTT





4. Testar o 3º Exemplo – AutoConnect_WiFi + PubSubClient + MQTT Cloud

Se for efetuado o exemplo 1 (AutoConnect) já não é necessário depois de fazer o Update deste código, voltar a configurar a ligação.

```
/*
 *  Sistemas Ubiquos
 *  MQTT - ESP-12E NodeCMU
 *  Pedro Matos
 *  AutoConnect_WiFi + PubSubClient + MQTT Cloud
 */
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <DNSServer.h>
#include <ESP8266WebServer.h>
#include <WiFiManager.h>
//Led Azul da Placa
#define LED 2
//Enter your MQTT server configurations
const char* mqttServer = "farmer.cloudmqtt.com";
const int mqttPort = 13172;
const char* mqttUser = "sidjminy";
const char* mqttPassword = "mdBcJhZuHOgR";

WiFiClient espClient;
PubSubClient client(espClient);

void setup() {
  delay(1000);
  pinMode(LED,OUTPUT);
  Serial.begin(115200);
 //WiFi
  WiFiManager wifiManager;
  wifiManager.autoConnect("AutoConnectAP");
  Serial.println("connected...WiFi.... :)");
  //MQTT - Ligação
  client.setServer(mqttServer, mqttPort);
  client.setCallback(MQTTcallback);
  while (!client.connected()) {
    Serial.println("Connecting to MQTT...");
//MQTT - Autenticação
    if (client.connect("ESP8266", mqttUser, mqttPassword )) {
      Serial.println("connected");
    } else {
      Serial.print("failed with state ");
      Serial.println(client.state());
      delay(2000);
    }
```

```
  }
 //MQTT - enviar uma vez para teste
  client.publish("esp/test", "Hello from ESP8266");
  client.subscribe("esp/test");
} //...:FIM Setup
 //----------Função para receber mensagem ClouMQTT
void MQTTcallback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived in topic: ");
  Serial.println(topic);
  Serial.print("Message:");
  String message;
  for (int i = 0; i < length; i++) {
    message = message + (char)payload[i];  //Conver *byte to String
  }
   Serial.print(message);
  if(message == "#on") {digitalWrite(LED,LOW);}   //LED on
  if(message == "#off") {digitalWrite(LED,HIGH);} //LED off
  Serial.println();
  Serial.println("-----------------------");
}
// Loop - faz sempre até o dispositivo ser desligado
void loop() {
  client.loop();
}
```