

## 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:  $\approx 70\text{mA}$  (200mA MAX), Standby:  $< 200\mu\text{A}$

Transfer rate: 110-460800bps

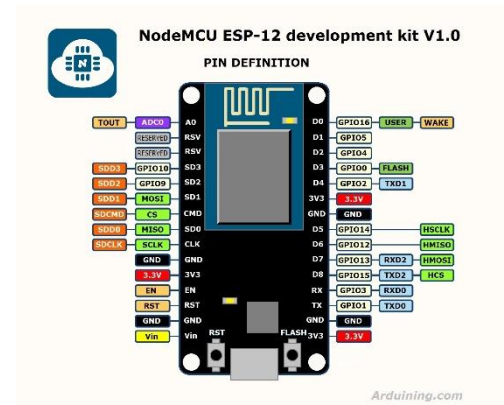
Support UART / **GPIO** data communication interface

Remote firmware upgrade (OTA)

Support Smart Link Smart Networking

Working temperature:  $-40^{\circ}\text{C} \sim +125^{\circ}\text{C}$

Drive Type: Dual high-power H-bridge driver



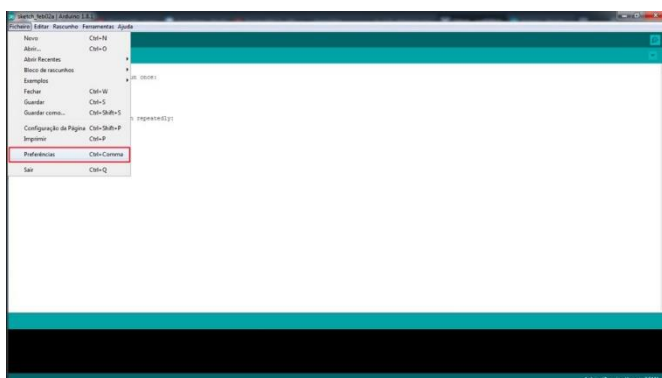
### 1. INSTALAÇÃO Placa ESP-12E NodeNCU no Arduino IDE

#### a. Configuração do lick para a placa ESP

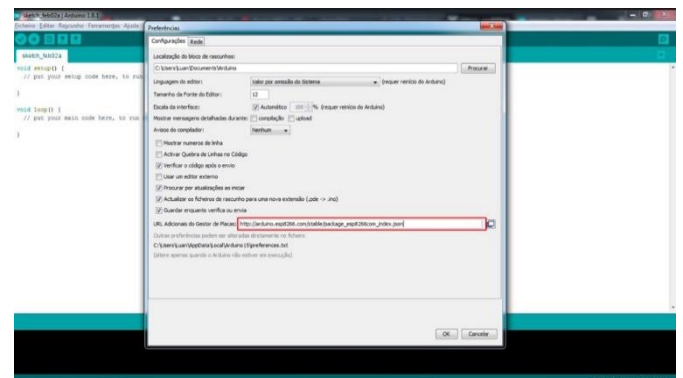
Link a ser colocado no IDE do Arduino (<https://github.com/esp8266/Arduino>):

[https://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](https://arduino.esp8266.com/stable/package_esp8266com_index.json)

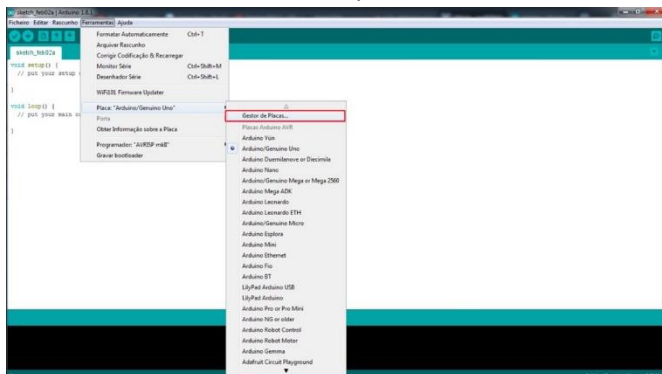
#### 1: Preferências



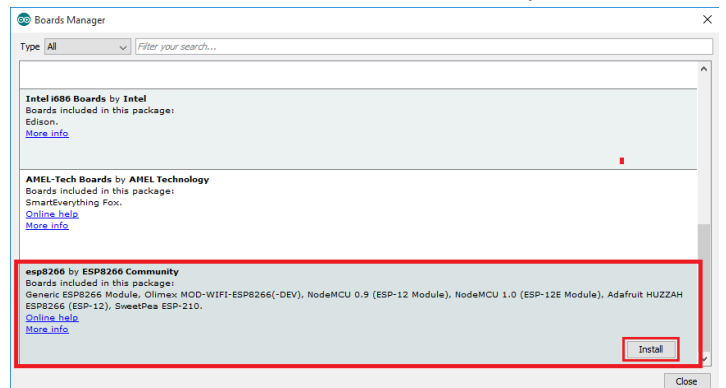
#### 2: Copy e Cole LINK



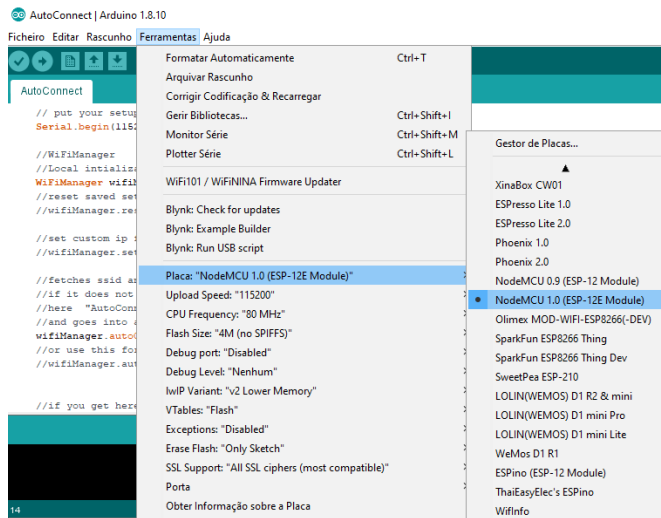
#### 3: Gestão de placas



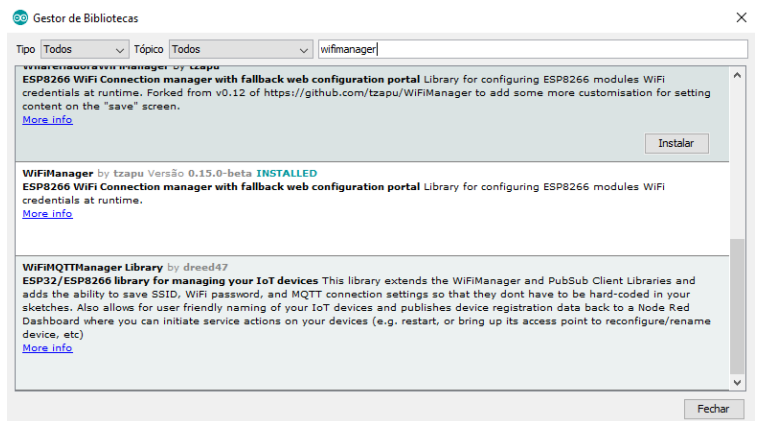
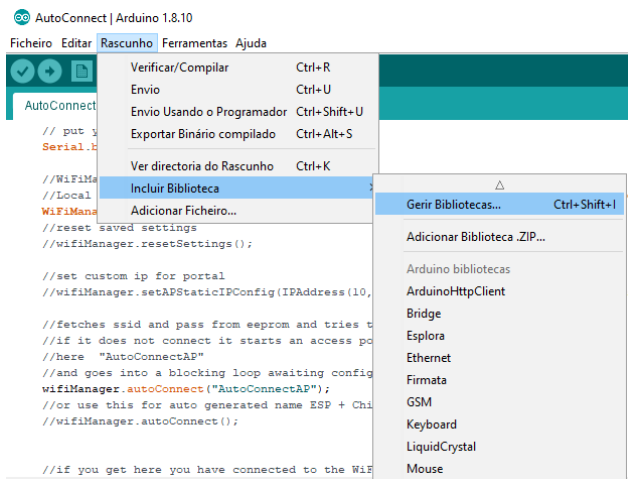
#### 4: instalar ESP8266 community



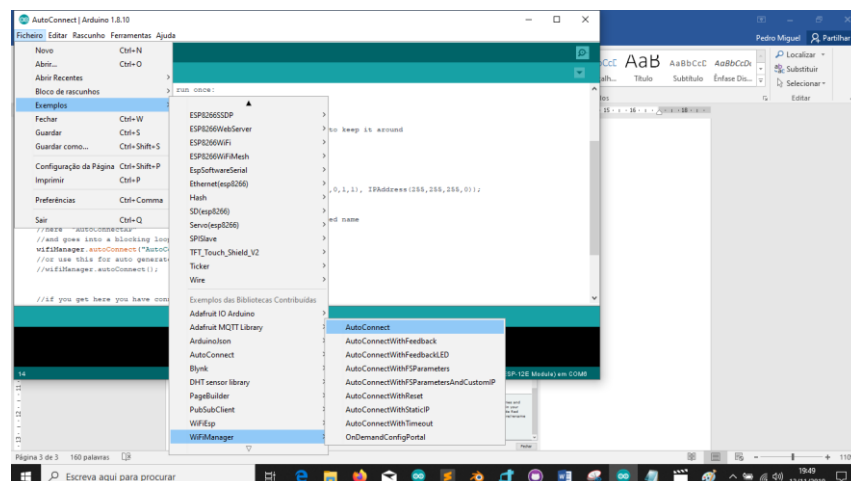
5: Escolher “**NodeMCU 1.0 ESP-12E**”



**b. Biblioteca WiFi ESP-12E**  
Adicionar biblioteca “WiFiManager”



**EXEMPLOS:**



## 2. Testar o 1º Exemplo – AutoConnect

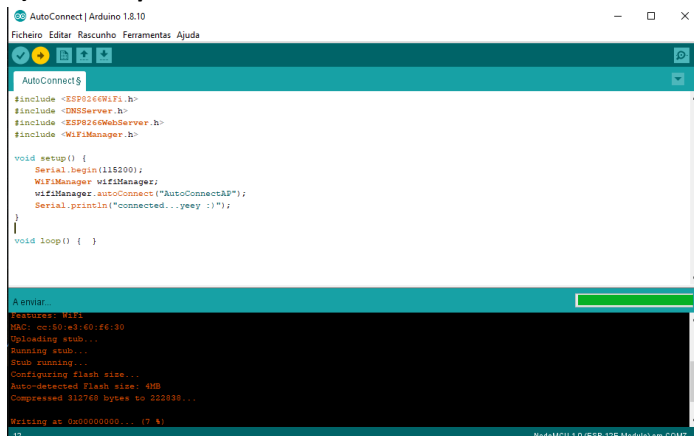
### a) Limpar o código do exemplo

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

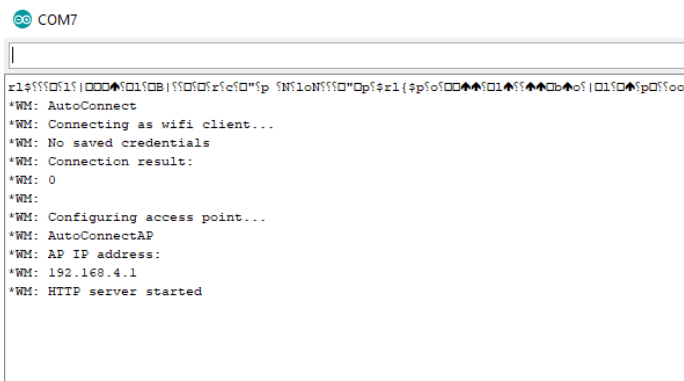
void setup() {
  Serial.begin(115200);
  WiFiManager wifiManager;
  wifiManager.autoConnect("AutoConnectAP");
  Serial.println("connected...yeey :)");
}

void loop() { }
```

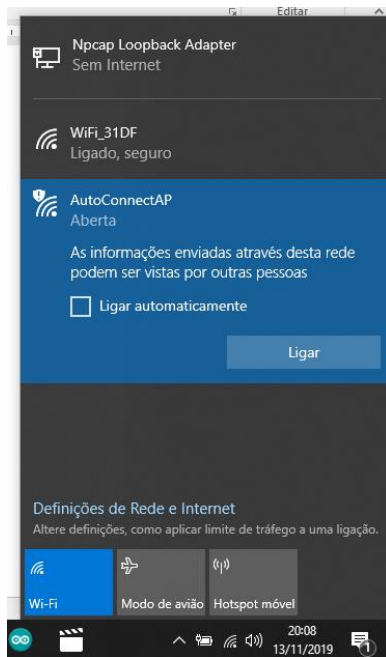
### b) Enviar para NodeMCU ESP-12E



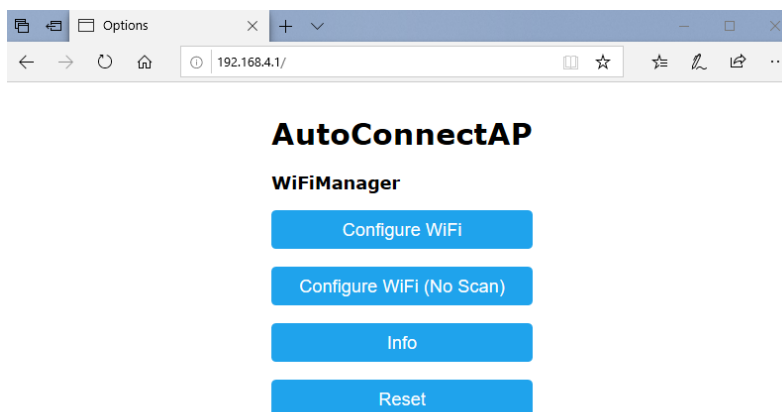
### c) Verificar se gravou e o ip atribuído pelo ESP-12E



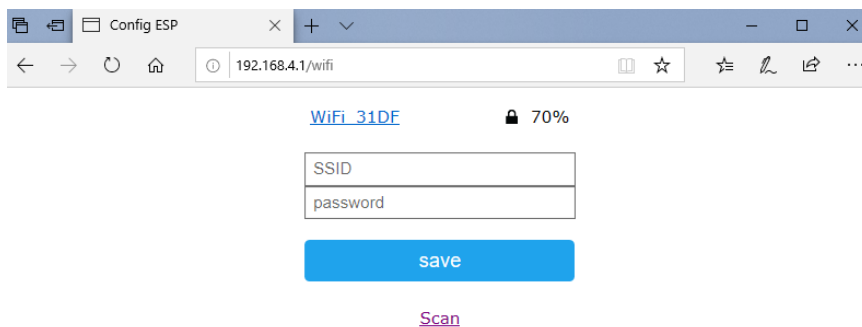
### d) Escolher e ligar a rede SSID dado pelo ESP-12E



**e) Navegador colocar ip ( 192.168.4.1 )**

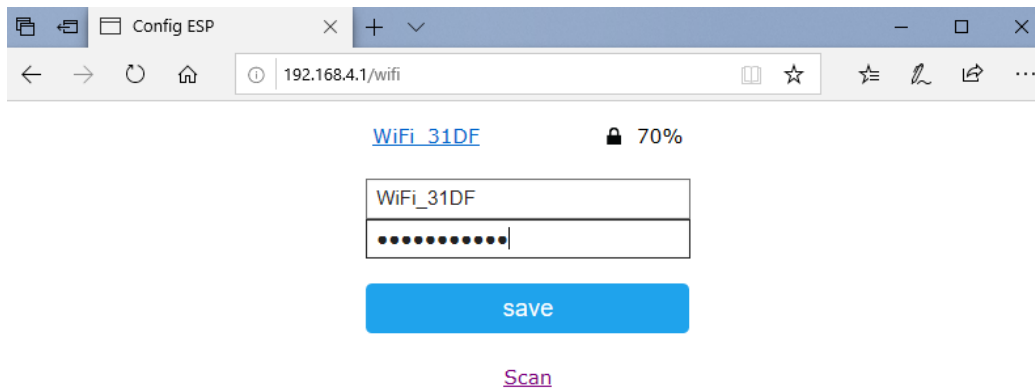


**f) Escolher Configurar WiFi**



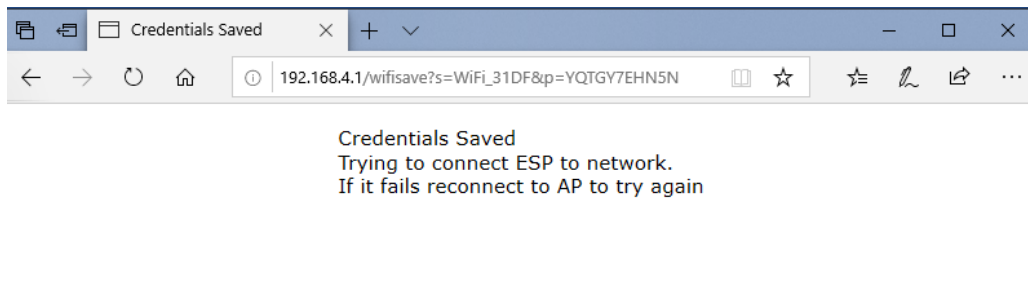
**g) Colocar SSID e a Password da rede pretendida.**

- SSID - Password - Clicar SAVE



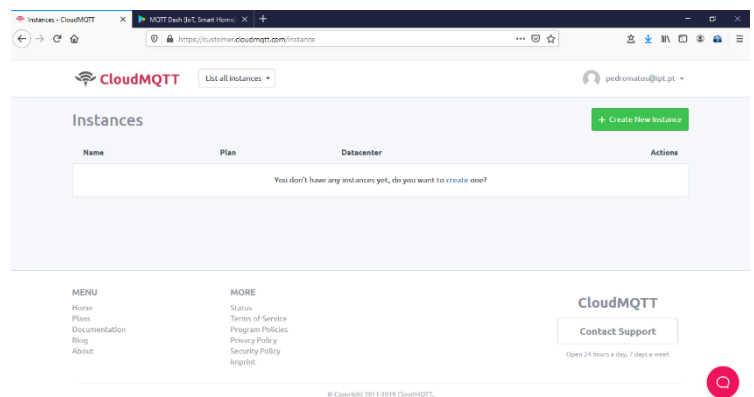
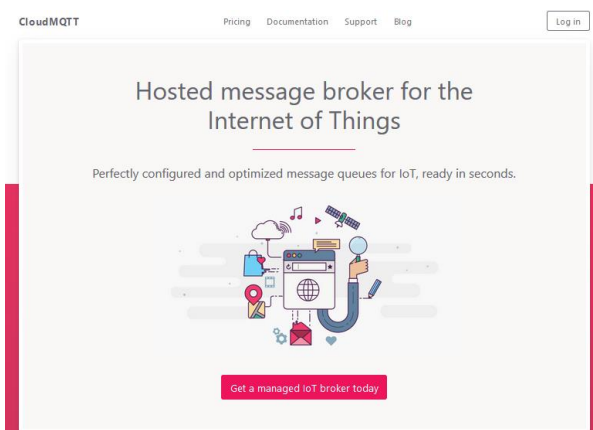
#### h) FIM da Configuração de rede

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

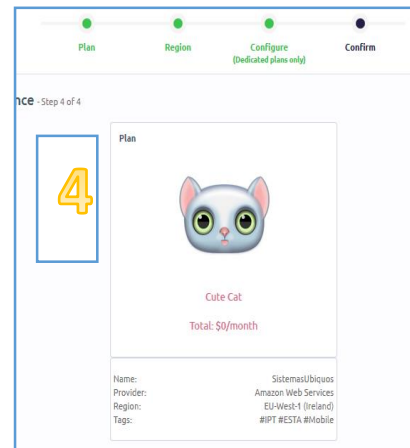
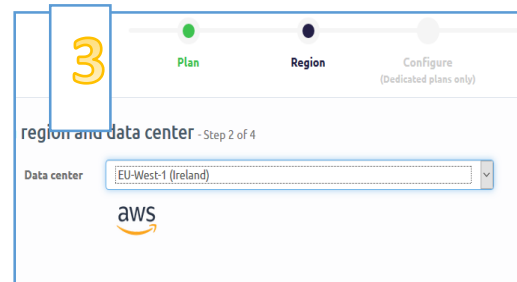
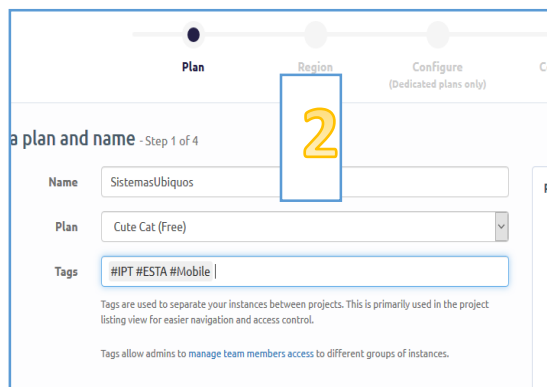
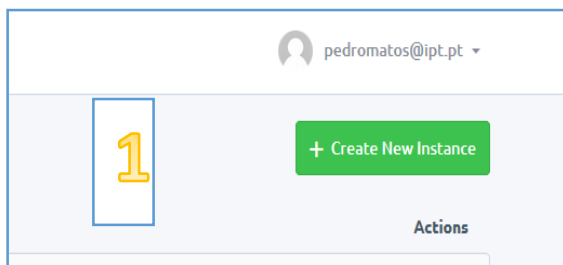


### 3. Testar o 2º Exemplo – MQTT (PubSubClient)

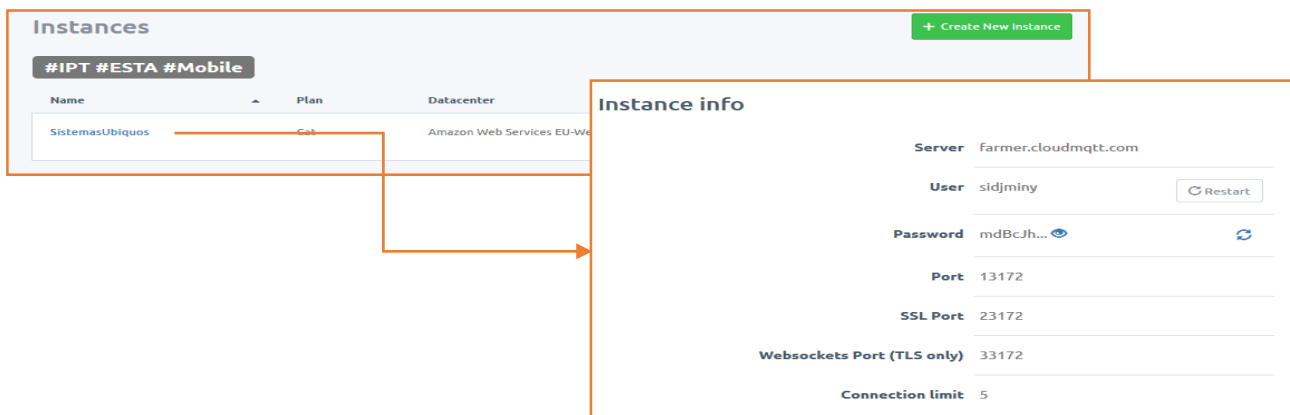
#### a) Criar conta CloudMQTT



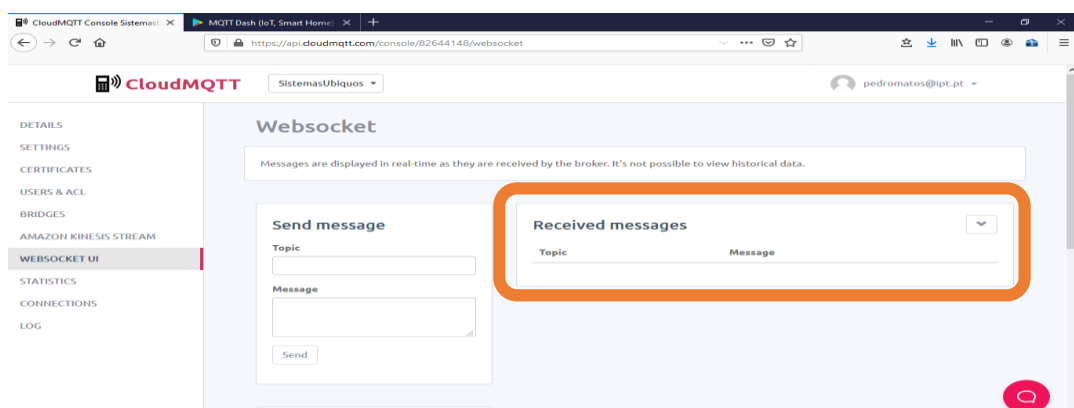
## - Criar um projeto (instance)



## - 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);
//Início – 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) {

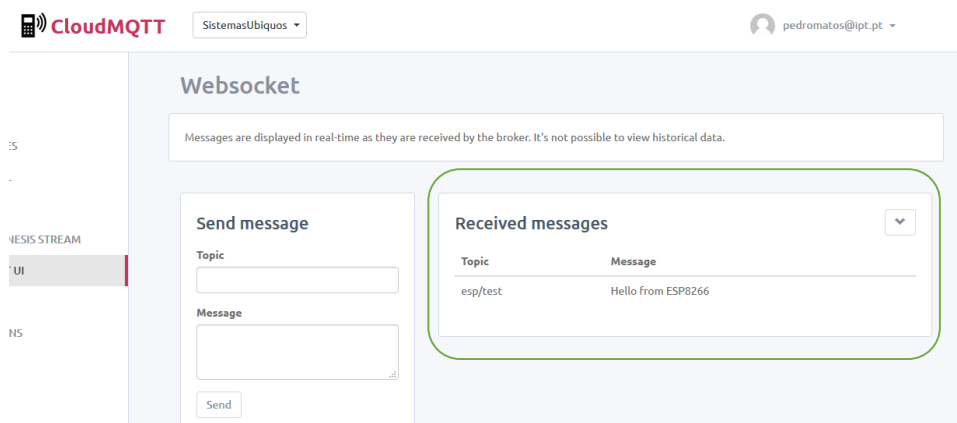
  Serial.print("Message arrived in topic: ");
  Serial.println(topic);
  Serial.print("Message:");

  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 ClouMQTT**

#### **Recepção do ESP-12E para CloudMQTT**





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

#### Send message

Topic

Message

Send

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

#### Send message

Topic

Message

Send

#### Received messages

Topic	Message
esp/test	Hello from ESP8266
esp/test	off
esp/test	#off
esp/test	#on
esp/test	#off

### d) Cliente Mobile



## MQTT Dash (IoT, Smart Home)

Routix software Comunicação

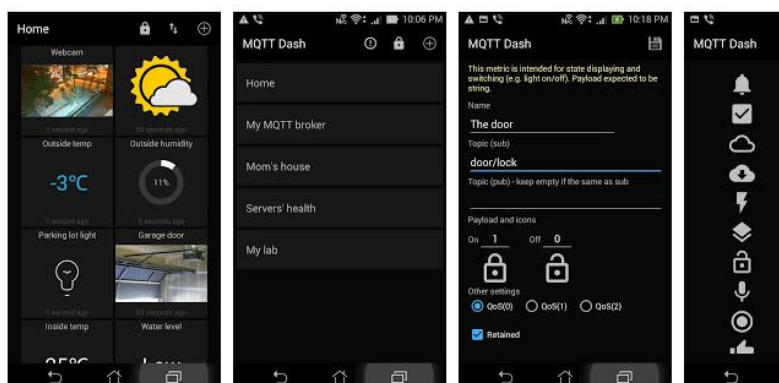
★★★★★ 3 301

PEGI 3

Esta aplicação é compatível com o seu dispositivo.

Adicionar à lista de desejos

Instalar



- Configurar com os dados de autenticação CloudMQTT

## Criar Ligação com MQTT Cloud

**MQTT Dash**

Default (automatically connect on start up).  
Note: this option is useful if you have just one connection configured.

☐ If you have more than one connection, you can create home screen shortcut for every connection.  
To create shortcut long press on any connection in connections list.

☒ Keep screen on when connected to this broker

Allow metrics management. If disabled, you can't add, edit, delete or rearrange metrics. This serves as protection from unintentional metrics changing.

☒

Name

**Led Estufa**

Address

**farmer.cloudmqtt.com**

Port

**13172**

Enable connection encryption (SSL/TLS).  
Note: if server certificate is self-signed, you need to install it to your device or enable option below, otherwise connection will fail. If server certificate issued by a known Certificate Authority (CA), it will work out of box, without installing to you device. Also don't forget, that MQTT servers have different ports for plain and SSL/TLS connections.

☐

☐ This broker uses self-signed SSL/TLS certificate. I trust this certificate at my own risk.

User name

**sidjminy**

User password

.....

Client ID (must be unique)

**mqttdash-c5120832**

Tile size

☐ Small

☒ Medium

☐ Large

Metrics columns count for vertical orientation (0 - auto)

**0**

Metrics columns count for horizontal orientation (0 - auto)

**0**

**MQTT Dash**

**Led Estufa**

## Criar o Botão de comunicação com MQTT

**MQTT Dash**

This metric is intended for state displaying and switching (e.g. light on/off). Or it can behave as a simple static button. Payload is expected to be string.

Name  
**On/Off**

Topic (sub)  
**esp/test**





Extract from JSON path (if payload is in JSON format), e.g.: \$.level.value. JSON path documentation at the URL below:  
<https://github.com/jayway/JsonPath/blob/master/README.md>

☒ Enable publishing

Topic (pub) - keep empty if the same as sub

☒ Update metric on publish immediately (do not wait for incoming message to update visual state)

Payload and icons. If you need not a switch, but a simple button, just set the same payload values and the same icons for On and Off. This way the switch will never change icon and always send the same payload value.

On <b>#on</b>	Off <b>#off</b>
	
	

Other settings

☐ QoS(0)

☒ QoS(1)

☐ QoS(2)

☐ Retained

Blink tile to draw attention, if the expression evaluates to 'true'.  
Expression can be any valid JavaScript expression which evaluates to boolean (true/false).  
You can use 'val' and 'secs' constants in your expression.  
'val' contains either, received raw payload or extracted from JSON raw value, if JSON path specified.  
'secs' contains numbers of seconds since last activity (last data received or image downloaded).  
For example, with the following expression, tile will blink if temperature is lower than 10 or higher than 90: 'val < 10 || val > 90'.

'On Receive' JavaScript handler called when new payload received for this metric

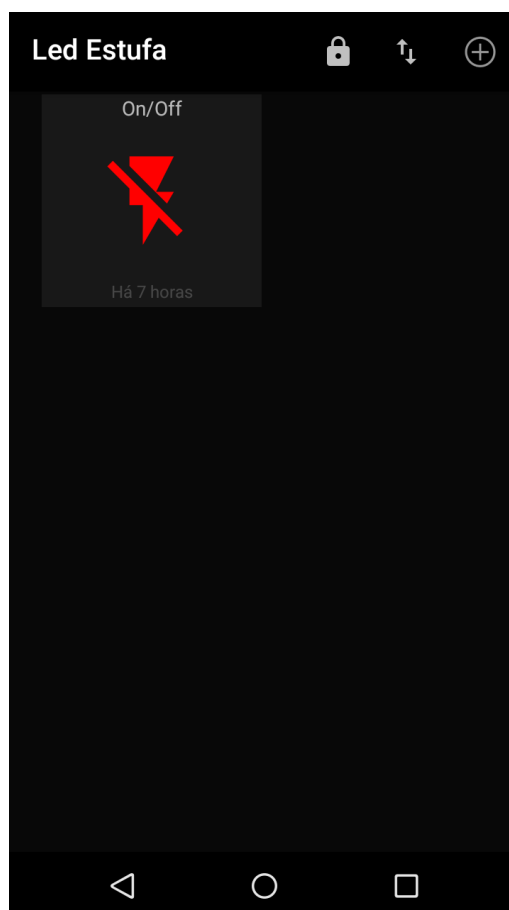
**ON RECEIVE**

'On Display' JavaScript handler called each time, when the app renders metric tile contents, e.g. when content changed or last activity time changed

**ON DISPLAY**

'On Tap' JavaScript handler called when you tap (click) on the tile

**ON TAP**



#### 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();
}

```