

## Programando o Wi-Fi

### Transcrição

Com as conexões físicas prontas entre o Módulo Relé e o NodeMCU, vamos dar inteligência ao nosso sensor, lembrando que essa inteligência vai funcionar a partir de um comando do *dashboard*, ou seja, o *dashboard* mandará um comando para ativar o nosso sensor, e esse comando funcionará via protocolo **MQTT**.

Para programar o NodeMCU, vamos utilizar o Arduino IDE, que precisa de algumas configurações antes de iniciarmos a programação.

### Preparando a IDE do Arduino

Antes de utilizar a IDE do Arduino, precisamos incluir nela as bibliotecas do módulos ESP8266. Para isso, no [GitHub do módulo](#) (<https://github.com/esp8266/Arduino>), nos é disponibilizado o seguinte link:

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

Nós copiamos esse link, e na IDE do Arduino, acessando o primeiro item do menu superior e clicando em **Preferências** (ou **Preferences**), nós colamos esse link no campo **URLs Adicionais para Gerenciadores de Placas** (ou **Additional Boards Manager URLs**), e clicamos em **OK**.

Após isso, acessamos o item **Ferramentas** no menu superior, depois **Placa** e clicamos em **Gerenciador de Placas...** (ou **Tools -> Board -> Boards Manager...**). Na barra de pesquisa, pesquisamos por **esp8266**, clicamos na placa que de mesmo nome que será exibida e clicamos em **Instalar** (ou **Install**).

Além disso, precisamos instalar a biblioteca para trabalharmos com o MQTT. Para isso, no menu superior, selecionamos **Sketch -> Incluir Biblioteca -> Gerenciar Bibliotecas...** (ou **Sketch -> Include Library -> Manage Libraries**), pesquisamos por **Adafruit MQTT** e instalamos a biblioteca **Adafruit MQTT Library** na sua versão mais recente.

**Adafruit MQTT Library** by Adafruit  
MQTT library that supports the CC3000, FONA, ESP8266, Yun, and generic Arduino Client hardware. Simple MQTT library that supports the bare minimum to publish and subscribe to topics.  
[More info](#)

Terminando a instalação, ao selecionar a placa na IDE, temos as opções do módulo ESP8266. O NodeMCU, em específico, corresponde à placa **NodeMCU 1.0 (ESP-12E Module)**, então basta selecioná-la, e em seguida selecionar sua porta, onde o NodeMCU está conectado.

Ao selecionar a placa, a IDE nos disponibiliza alguns exemplos **relativos a ela**, ou seja, os exemplos variam de acordo com a placa. Acessando o primeiro item do menu superior, e selecionando **Exemplos -> Adafruit MQTT Library -> mqtt\_2subs\_esp8266** (ou **Examples -> Adafruit MQTT Library -> mqtt\_2subs\_esp8266**), temos um dos exemplos disponibilizados, no qual iremos nos basear para a nossa programação.

### Iniciando a programação

Primeiramente, devemos importar as bibliotecas para podermos utilizar o MQTT e o Wi-Fi:

```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
```

Depois, criamos constantes para o pino de controle do NodeMCU conectado o Módulo Relé (**D2**) e para o nome da rede e sua senha:

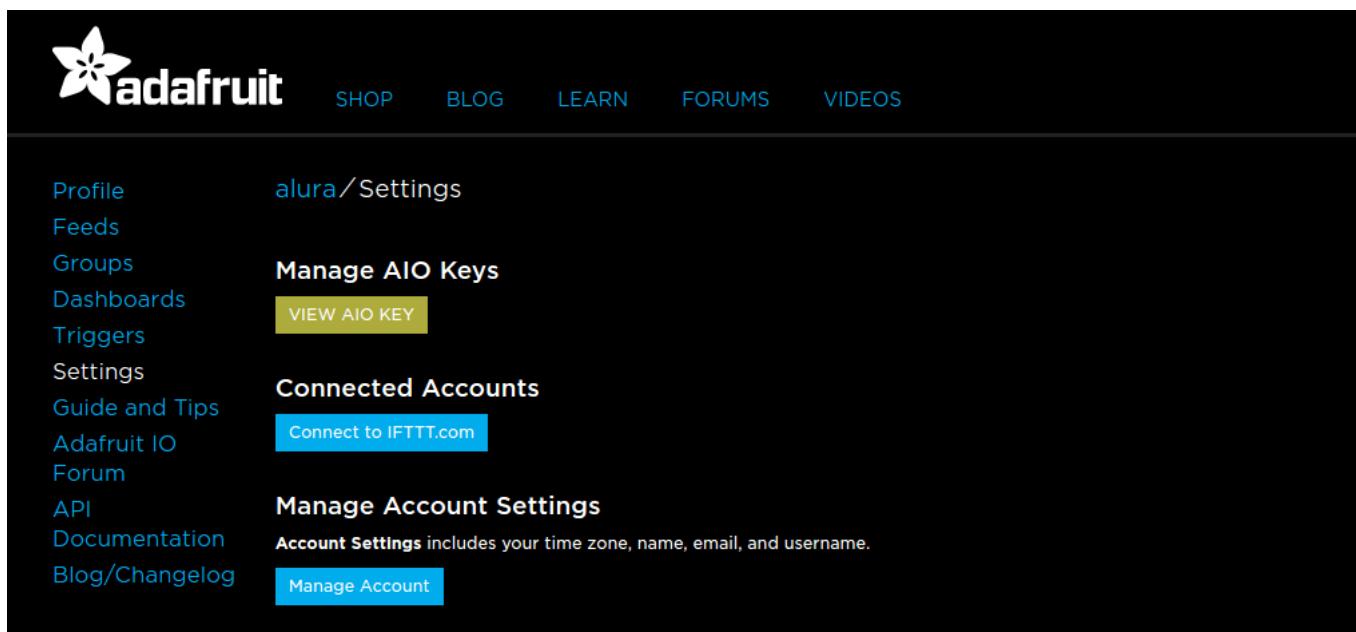
```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

#define RELAY D2

/***************** WiFi Access Point *******/

#define WLAN_SSID      "alura-rede"          // TROQUE PELO NOME DA SUA REDE
#define WLAN_PASS      "senhaDaRede"         // TROQUE PELA SENHA DA SUA REDE
```

Agora, precisamos fazer o link do NodeMCU com o nosso *dashboard*. Para isso, precisamos do nome do nosso usuário e da chave do Adafruit IO. Então, acessamos o [Adafruit IO \(https://io.adafruit.com\)](https://io.adafruit.com), fazemos o login e acessamos o item **Settings**, no menu à esquerda:



Em **Manage AIO Keys**, clicamos em **VIEW AIO KEY** para ver o nosso usuário e a nossa chave. Copiamos esses dois dados e criamos mais duas constantes no código:

```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

#define RELAY D2

/***************** WiFi Access Point *******/

#define WLAN_SSID      "alura-rede"          // TROQUE PELO NOME DA SUA REDE
```

```
#define WLAN_PASS      "senhaDaRede"          // TROQUE PELA SENHA DA SUA REDE

/***** Adafruit.io Setup *****/

#define AIO_USERNAME   "nomeDoUsuario"        // TROQUE PELO SEU USUARIO
#define AIO_KEY         "chaveDoUsuario"       // TROQUE PELO SUA CHAVE
```

Além disso, precisamos definir qual o servidor MQTT, que será o próprio Adafruit IO, e a sua porta, que por padrão é 1883, mas como vamos utilizar SSL, a porta será 8883:

```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

#define RELAY D2

/***** WiFi Access Point *****

#define WLAN_SSID      "alura-rede"          // TROQUE PELO NOME DA SUA REDE
#define WLAN_PASS       "senhaDaRede"         // TROQUE PELA SENHA DA SUA REDE

/***** Adafruit.io Setup *****

#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  8883
#define AIO_USERNAME    "nomeDoUsuario"        // TROQUE PELO SEU USUARIO
#define AIO_KEY         "chaveDoUsuario"       // TROQUE PELO SUA CHAVE
```

Para encerrar a configuração, com o `WiFiClientSecure` criamos um cliente Wi-Fi para se conectar ao servidor MQTT com SSL , e além disso, vamos configurar a classe do cliente MQTT, passando para ele o cliente Wi-Fi, o servidor MQTT e os detalhes de login:

```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

#define RELAY D2

/***** WiFi Access Point *****

#define WLAN_SSID      "alura-rede"          // TROQUE PELO NOME DA SUA REDE
#define WLAN_PASS       "senhaDaRede"         // TROQUE PELA SENHA DA SUA REDE

/***** Adafruit.io Setup *****

#define AIO_SERVER      "io.adafruit.com"
#define AIO_SERVERPORT  8883
#define AIO_USERNAME    "nomeDoUsuario"        // TROQUE PELO SEU USUARIO
#define AIO_KEY         "chaveDoUsuario"       // TROQUE PELO SUA CHAVE

/***** Global State *****

WiFiClientSecure client;
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_USERNAME, AIO_KEY');
```



Com isso, concluímos o *setup* inicial. Na próxima aula, mexeremos com os *feeds*, os tópicos do MQTT que queremos configurar.