

01

## Conectando ao Wifi

### Transcrição

Para que o nosso projeto possa publicar mensagens em um *broker*, ele deve primeiramente ter acesso à internet. Para isso, antes de mais nada, precisamos incluir a biblioteca **PubSubClient**, selecionando **Sketch -> Incluir Biblioteca -> Gerenciar Bibliotecas...** (ou **Sketch -> Include Library -> Manage Libraries**):

**PubSubClient** by Nick O'Leary Versão 2.6.0 **INSTALLED**  
**A client library for MQTT messaging.** MQTT is a lightweight messaging protocol ideal for small devices. This library allows you to send and receive MQTT messages. It supports the latest MQTT 3.1.1 protocol and can be configured to use the older MQTT 3.1 if needed. It supports all Arduino Ethernet Client compatible hardware, including the Intel Galileo/Edison, ESP8266 and TI CC3000.  
[More info](#)

Com essa biblioteca, já podemos fazer tanto o acesso Wi-Fi quanto o uso do MQTT. Utilizaremos como consulta o exemplo **mqtt\_esp8266**, que vem junto com a biblioteca.

### Conectando o NodeMCU ao Wi-Fi

Primeiramente, devemos importar a biblioteca responsável pela conexão Wi-Fi e criar duas constantes, uma representando o nome da rede e outra a senha, além do cliente Wi-Fi:

```
// --- WIFI ---
#include <ESP8266WiFi.h>
const char* ssid = "lab-rmerces";           // troque pelo nome da sua rede
const char* password = "rmerces-iot";        // troque pela senha da sua rede
WiFiClient nodemcuClient;

// restante do código omitido
```

Agora, vamos criar a função **conectarWifi()**, responsável por realizar a conexão com a nossa rede:

```
/* Exibindo somente a função
conectarWifi(), omitindo o
restante do código */

void conectarWifi() {
```

}

E podemos copiar o código responsável pela conexão do exemplo:

```
/* Exibindo somente a função
conectarWifi(), omitindo o
restante do código */

void conectarWifi() {
    delay(10);
```

```

Serial.println();
Serial.print("Connecting to ");
Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
}

```

Mas em vez de exibir as informações na *Serial*, vamos exibi-las no display:

```

/* Exibindo somente a função
conectarWifi(), omitindo o
restante do código */

void conectarWifi() {
    delay(10);

    display.setTextSize(2);
    display.setCursor(0, 0);
    display.print("Conectando ");
    display.display();

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        display.print(".");
        display.display();
    }
}

```

A função que efetivamente realiza a conexão com a rede é a função `begin`, e enquanto não estivermos conectados (`WiFi.status() != WL_CONNECTED`), vamos exibindo pontos no display.

Finalizada a função, devemos chamá-la dentro a função `setup()`. Além disso, podemos remover o código de inicialização da *Serial*, já que não iremos mais utilizá-la:

```

/* Exibindo somente a função
setup(), omitindo o restante do código */

void setup() {
    configurarDisplay();
    conectarWifi();
}

```

Com o Wi-Fi configurado, vamos partir para o MQTT a partir do próximo vídeo.

