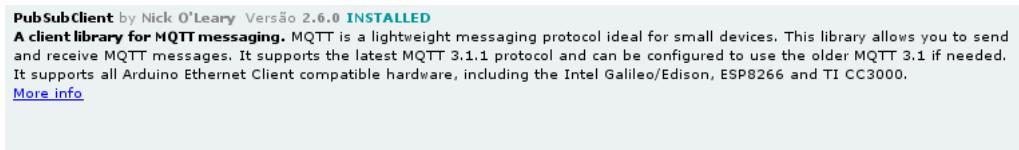


12

## Mão na massa: Finalizando o projeto

Para que o projeto possa publicar mensagens em um *broker*, ele deve primeiramente ter acesso à internet. Para isso, antes de mais nada, inclua a biblioteca ***PubSubClient***, selecionando no menu superior da IDE do Arduino o item **Sketch -> Incluir Biblioteca -> Gerenciar Bibliotecas...** (ou **Sketch -> Include Library -> Manage Libraries**):



Com essa biblioteca, você já pode fazer tanto o acesso Wi-Fi quanto o uso do MQTT. Para isto, siga os passos abaixo, que foram feitos baseados no exemplo **mqtt\_esp8266**, que vem junto com a biblioteca.

1 - Importe as bibliotecas **ESP8266WiFi.h** e **PubSubClient.h**.

2 - Crie constantes com os valores do nome da sua rede, senha da rede, *server* MQTT, nome do tópico MQTT, nome do usuário, senha e porta do tópico.

3 - Crie uma instância de **WiFiClient** e uma de **PubSubClient**, passando o cliente Wi-Fi por argumento.

4 - Crie a função **setup\_wifi()**, que fará a conexão Wi-Fi:

```
void setup_wifi() {
    delay(10);
    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());
}
```

5 - Na função **setup()**, chame a função **setup\_wifi()** e faça a configuração do *server* do MQTT.

6 - Crie a função **reconnect()**, que fará a conexão com o MQTT:

```
void reconnect() {
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        if (client.connect("ESP8266Client", mqtt_user, mqtt_password)) {
```

```
    Serial.println("connected");
} else {
    Serial.print("failed, rc=");
    Serial.print(client.state());
    Serial.println(" try again in 5 seconds");
    delay(2000);
}
}
```

7 - Por fim, no começo da função `loop()`, faça a conexão com o MQTT, caso ela ainda não tenha sido feita. E após a leitura do *trigger*, publique o status no MQTT:

```
void loop() {
    if (!client.connected()) {
        reconnect();
    }

    int trigger = digitalRead(PIR);
    client.publish(mqtt_topic, String(trigger).c_str());
    if (trigger == HIGH) {
        liga_laser();
    }
    else {
        desliga_laser();
    }
    delay(1000);
}
```