

## Mãos 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**):

**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, 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);
}
```