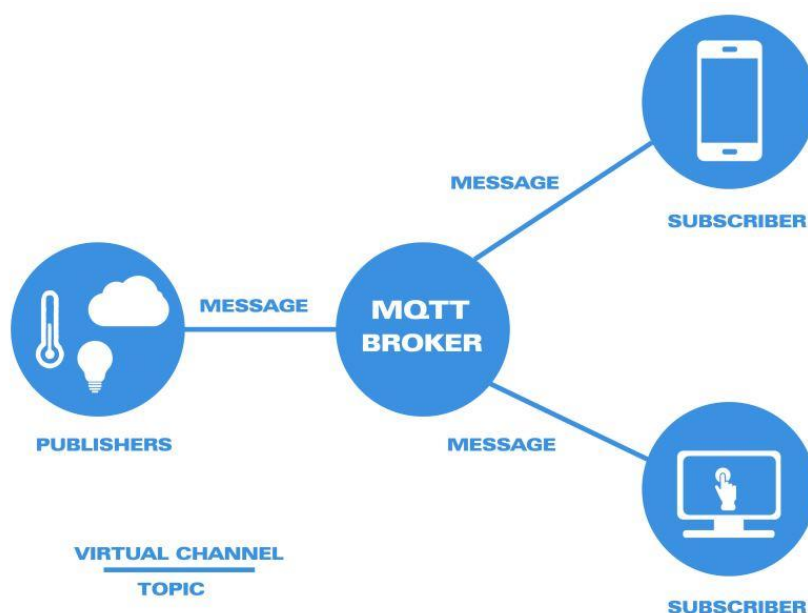




Parte 2: Controle do LED Através do MQTT

Neste tutorial vamos criar uma aplicação Node-RED que irá controlar o funcionamento de um LED através da troca de mensagens em um servidor MQTT (Message Queue Telemetry Transport).

Artigo: Conhecendo o MQTT. Por que o MQTT é um dos melhores protocolos de rede para a Internet das Coisas? Disponível em <https://developer.ibm.com/br/technologies/iot/articles/iot-mqtt-why-good-for-iot/>



Dica: Eclipse Mosquitto é um broker MQTT gratuito. Mais informações: <https://mosquitto.org/>

No Node-RED realize a criação de um fluxo. Em seguida, insira dois nós do tipo **inject** e um nó **mqtt out**. Faça as conexões conforme mostra a Figura 1.

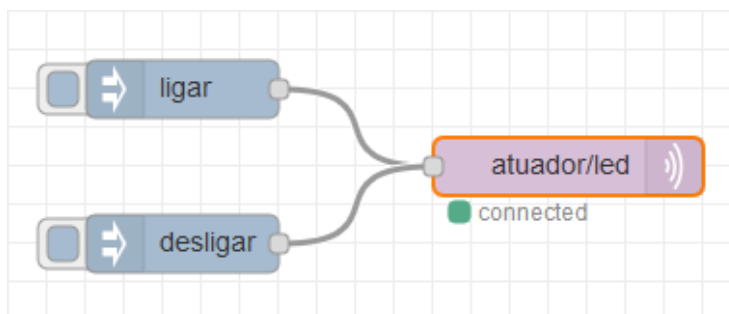


Figura 1: Utilização de um Broker MQTT

Clique sobre um dos nós **inject** e defina a propriedade **Payload** com a string ligar (Figura 2). Repita o mesmo procedimento para o outro nó **inject**, mas defina a propriedade **Payload** com a string desligar.

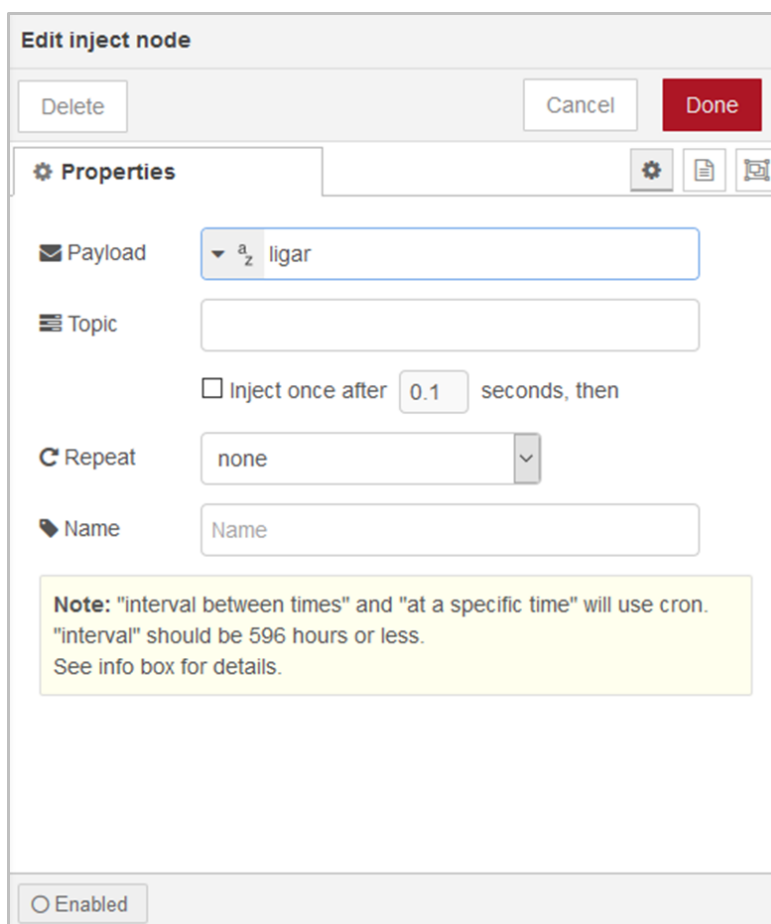


Figura 2: Alteração das Propriedades do Nó

Em seguida, dê um clique duplo sobre o nó **mqtt out** para entrar no modo de edição do nó. Conforme podemos observar na Figura 3, na propriedade **Server**, escolha a opção **Add new mqtt-broker...** e clique no ícone , responsável pela edição dos dados do servidor.

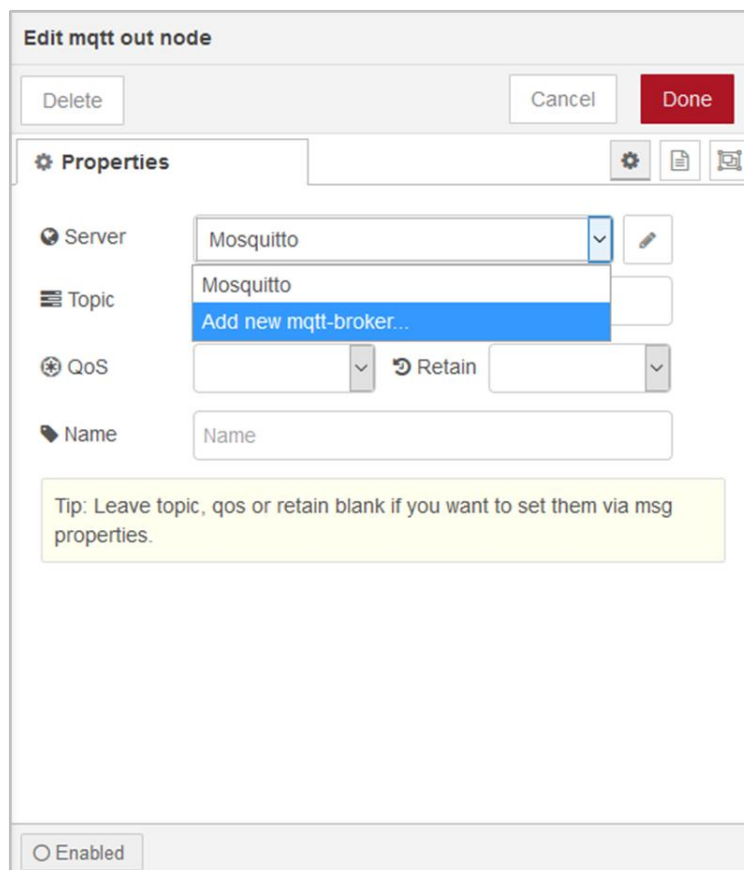


Figura 3: Adicionando um Broker MQTT

No painel que será mostrado (Figura 4), preencha o nome do broker (Propriedade **Name**), por exemplo, Mosquitto e, na propriedade **Server**, coloque o endereço IP local ou **test.mosquitto.org**. Após preencher os dados, clique no botão **Add**.

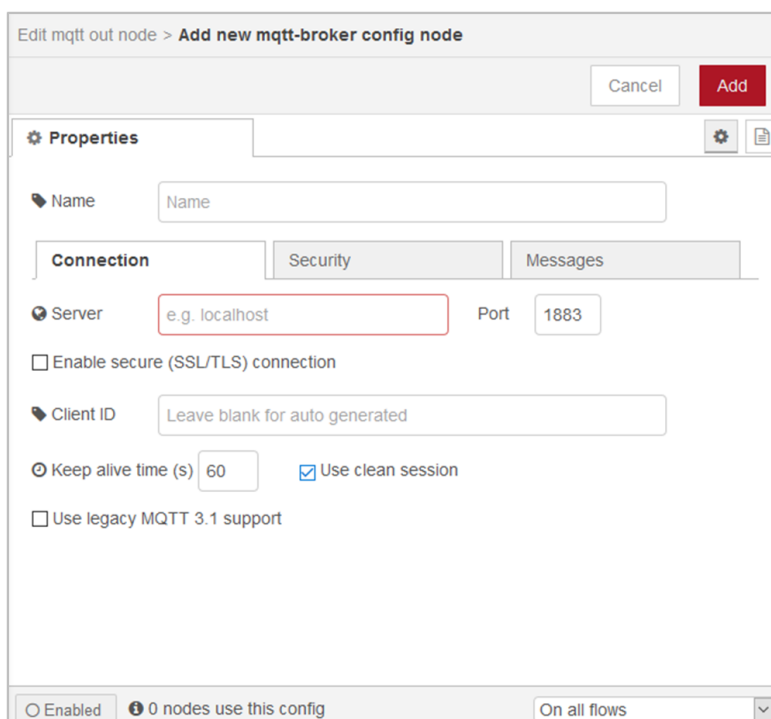


Figura 4: Configuração do Servidor

De volta ao painel de edição do nó (Figura 5), selecione na propriedade **Server** o broker MQTT que foi criado. Na sequência preencha a propriedade **Topic** com atuador/led e clique no botão **Done**.

The image shows the 'Edit mqtt out node' configuration window in Node-RED. At the top, there are 'Delete', 'Cancel', and 'Done' buttons. Below is a 'Properties' section with a gear icon and three utility icons. The 'Server' property is a dropdown menu currently showing 'Mosquitto'. The 'Topic' property is a text input field containing 'atuador/led'. The 'QoS' property is a dropdown menu set to '0', and the 'Retain' property is a checkbox that is checked. The 'Name' property is a text input field containing 'Name'. A yellow tip box at the bottom of the properties section reads: 'Tip: Leave topic, qos or retain blank if you want to set them via msg properties.' At the bottom of the window, there is an 'Enabled' checkbox which is currently unchecked.

Figura 5: Definição do Servidor e do Tópico

Clique no botão **Deploy** para executar o fluxo. Observe que ao clicar no nó **inject ligar** o LED deverá ser aceso, ao clicar no nó **inject desligar** o LED irá apagar.

Dashboard

O Node-RED apresenta um conjunto de nós que possibilitam a construção de interfaces gráficas que irão facilitar para o usuário a interação e visualização das informações. Adicione à paleta o módulo “node-red-dashboard”, para isso clique no menu principal e, em seguida, na opção “Manage palette”, conforme ilustra a Figura 6.

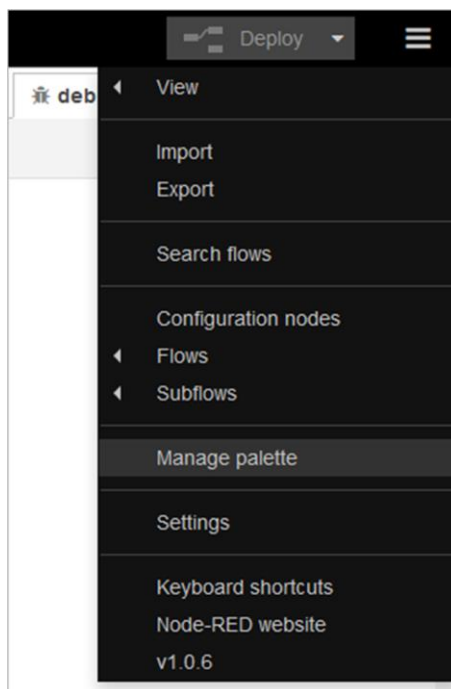


Figura 6: Acesso ao Menu do Node-RED

Em seguida, clique na aba Install (Figura 7), procure pelo módulo “node-red-dashboard” e clique no botão **install**.

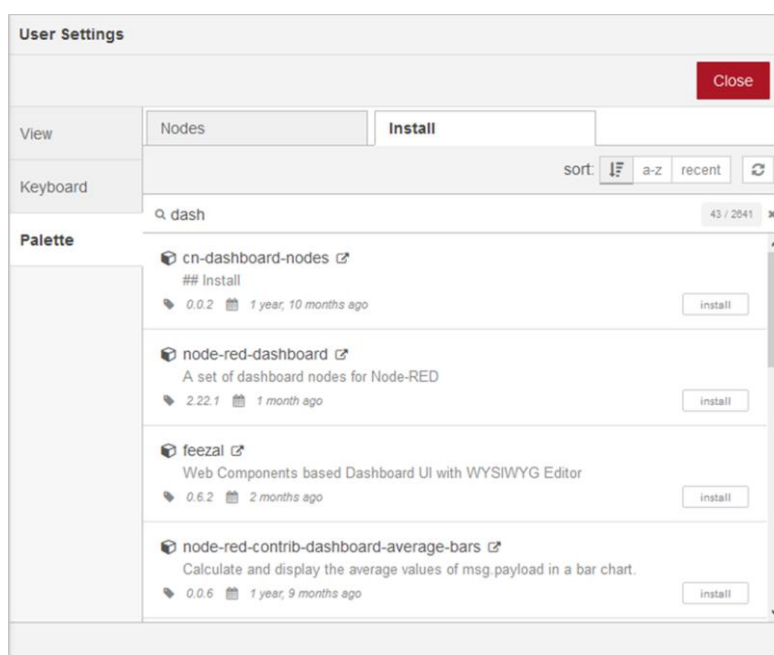


Figura 7: Instalação dos Módulos

No Node-RED realize a criação de um fluxo. Em seguida, insira dois nós do tipo **switch** e um nó **mqtt out**, realizando a conexão entre eles, conforme mostra a Figura 8.

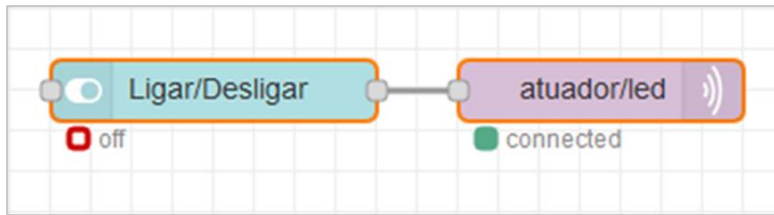


Figura 8: Nós **switch** e **mqtt out**

Entre na edição do nó **switch**, realizando um clique duplo do mouse sobre ele. O primeiro passo consiste em criar um Grupo, para fazer isso, selecione a opção “Add new ui_group...” presente no atributo “Group” e clique no botão de edição, que é identificado pelo ícone do lápis. Agora selecione o atributo Tab e escolha a opção “Add new ui_tab...” e clique no botão de edição (ícone do lápis). Defina a propriedade Name com o valor “Painel de Controle”, conforme mostra a Figura 9 e clique no botão “Update”.

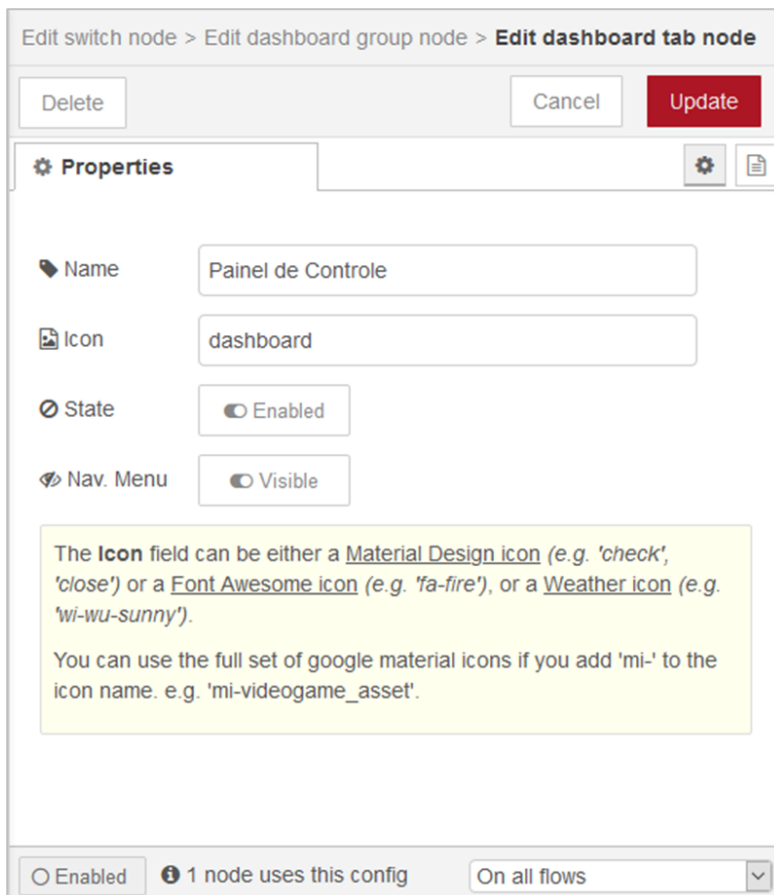


Figura 9: Criação do Nó **dashboard tab**

Agora defina o atributo Nome com o valor “LED” e, para o atributo Tab, selecione a opção “Painel de Controle” (Figura 10). Clique no botão Update para finalizar a edição do nó **dashboard group**.

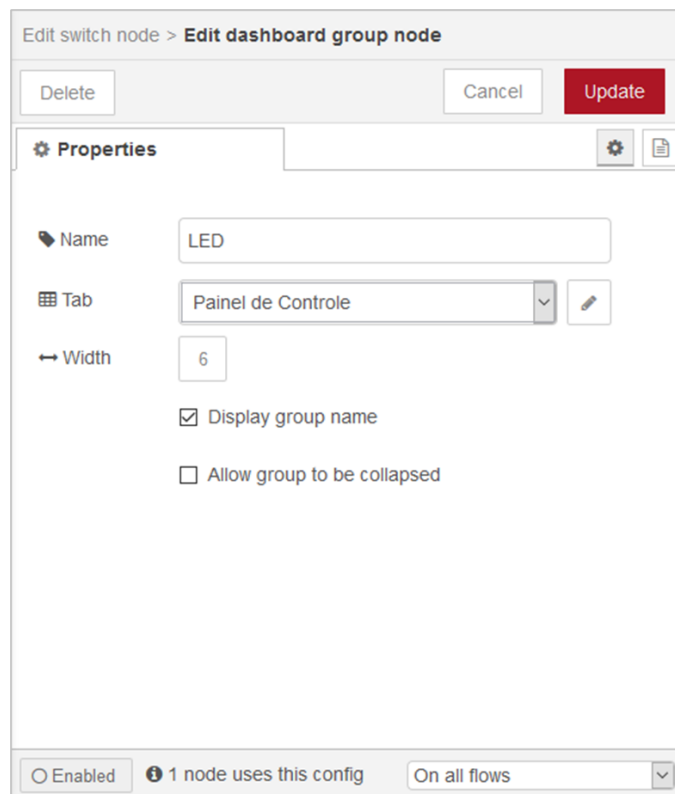


Figura 10: Configuração do Nó **dashboard group**

Ao retornar ao painel de edição do nó switch, na propriedade Group selecione a opção LED. A propriedade Label deverá receber o valor “Ligar/Desligar”. A mensagem a ser enviada no atributo On Payload é “ligar” e no atributo Off Payload é “desligar”, conforme ilustra a Figura 11.

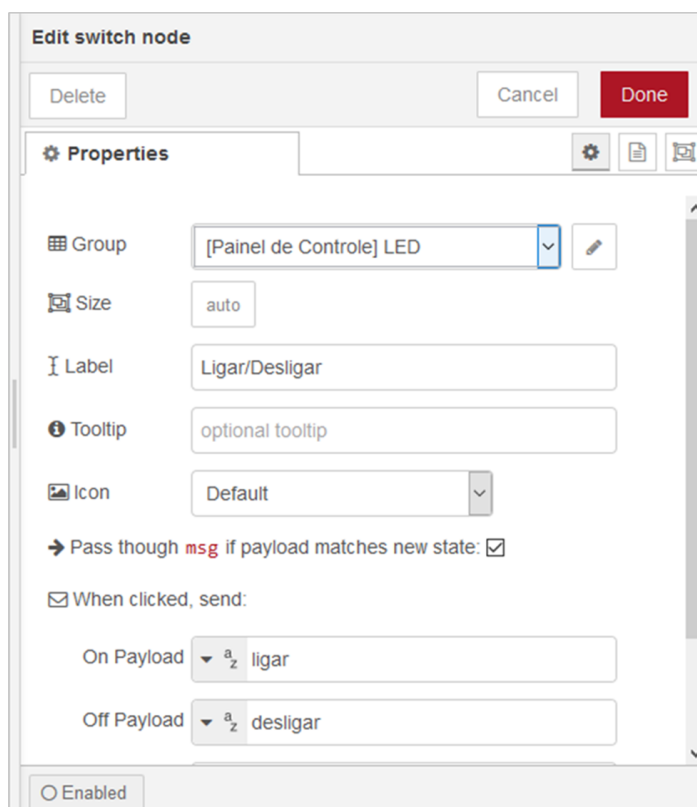


Figura 11: Configuração do Nó **switch**

No NodeMCU use a montagem e o programa (mqtt-led.py) adotados no projeto anterior. No Node-RED clique no botão **Deploy** para executar o fluxo. Abra uma nova aba no navegador e digite a URL <http://localhost:1880/ui>. Na Figura 12 apresentamos a exibição do painel de controle (dashboard) que foi criado.

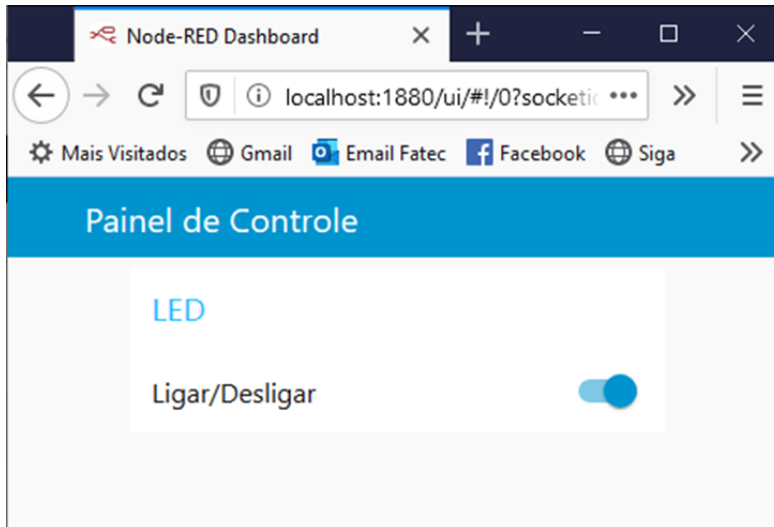


Figura 12: Dashboard Criado

Cliente MQTT com ESP e MicroPython

O Node-RED, como já mencionado anteriormente, possibilita a comunicação de dispositivos de hardware com serviços online. Este projeto irá demonstrar esta possibilidade, através da implementação de um cliente MQTT. Utilize os materiais relacionados a seguir para montar o circuito eletrônico.

Relação de materiais

- 1 NodeMCU (ESP8266 ou ESP32).
 - 1 Resistor de 220 Ohms (vermelho, vermelho, marrom) ou 330 Ohms (laranja, laranja, marrom).
 - 1 LED (qualquer cor).
 - 1 Protoboard.
 - Cabos de ligação.
-

Em seguida, realize a montagem da maneira indicada pela Figura 13, caso esteja utilizando o NodeMCU (ESP8266).

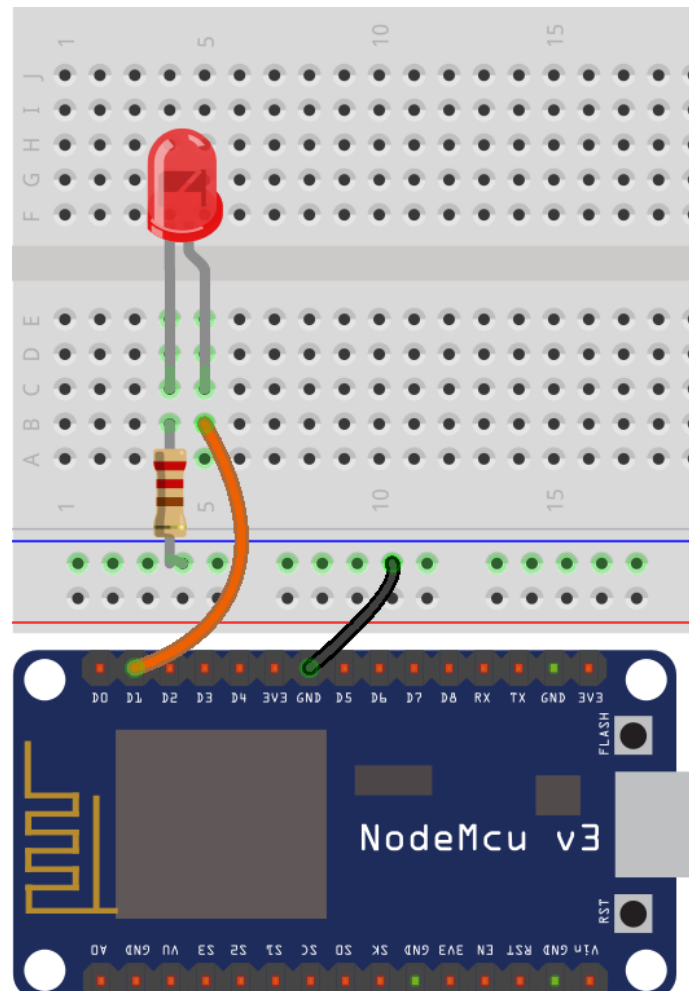


Figura 13: Circuito Eletrônico (NodeMCU-ESP8266)

Por outro lado, se estiver usando o NodeMCU (ESP32) adote a Figura 14 como referência.

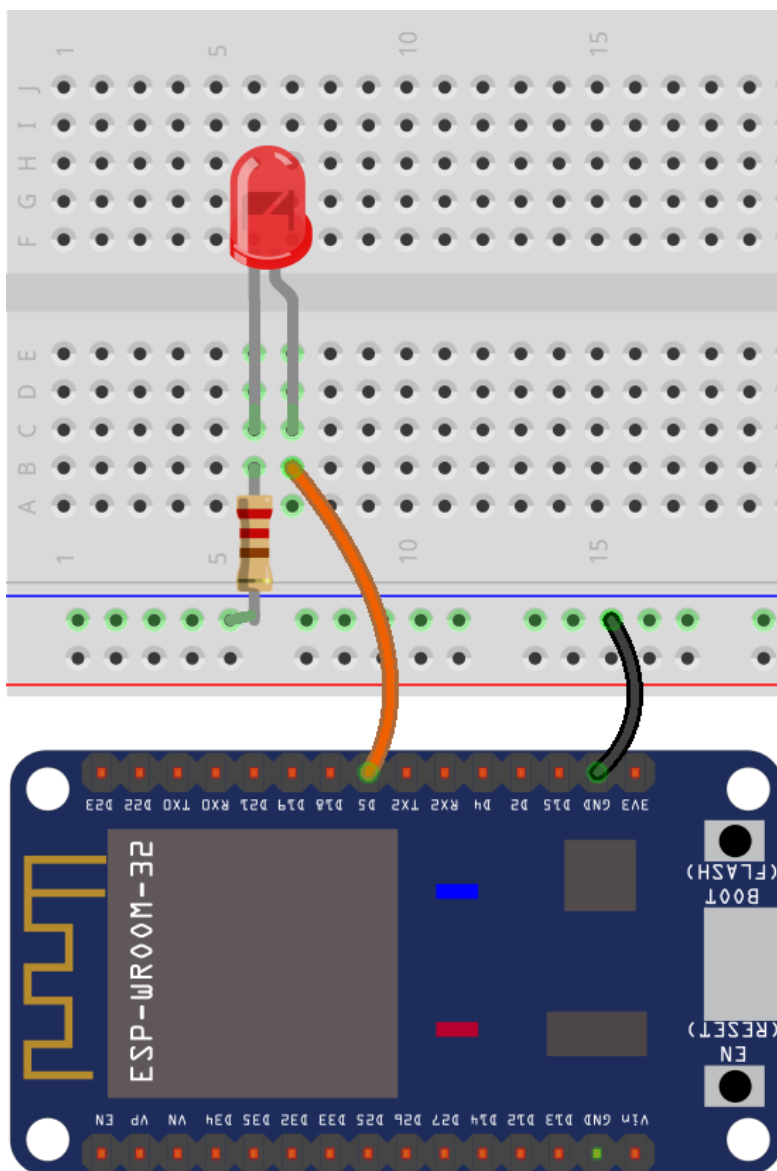


Figura 14: Circuito Eletrônico (NodeMCU-ESP32)

No ambiente de desenvolvimento Thonny, disponível para download gratuito em <https://thonny.org/>, iremos desenvolver o programa **mqtt-led.py**, que consiste na implementação de um cliente MQTT, que aguarda o recebimento das mensagens postadas em determinado tópico no broker. Lembre-se apenas de ajustar o endereço IP do servidor (Linha 23) para acessar o servidor instalado no seu computador ou o endereço do ambiente de testes.



```
1 import network, machine
2 from umqtt.simple import MQTTClient
3 from machine import Pin
4
5 estacao = network.WLAN(network.STA_IF)
6 estacao.active(True)
7 estacao.connect('ap', 'senha')
8 while estacao.isconnected() == False:
9     pass
10 print('Conexao realizada.')
11 print(estacao.ifconfig())
12
13 def obter_mensagem(topico, mensagem):
14     print(topico, mensagem)
15     if mensagem == b'ligar':
16         led.value(1)
17     elif mensagem == b'desligar':
18         led.value(0)
19
20 led = Pin(5, Pin.OUT)
21
22 print ('Conectando ao servidor MQTT')
23 servidor = 'test.mosquitto.org'
24 topico = 'atuador/led'
25 cliente = MQTTClient('NodeMCU', servidor, 1883)
26 cliente.set_callback(obter_mensagem)
27 cliente.connect()
28 cliente.subscribe(topico)
29 try:
30     while True:
31         cliente.wait_msg()
32 finally:
33     cliente.disconnect()
34     estacao.disconnect()
35     estacao.active(False)
36     print("Fim.")
37
```

mqtt-led.py