

### Projeto ESP32 com Sensor Ultrassônico e MQTT

Este projeto utiliza um ESP32 com sensor ultrassônico para medir distâncias e enviar os dados via MQTT para um painel de controle no Node-RED.

### Componentes Utilizados

- ESP32 Devkit
- Sensor Ultrassônico HC-SR04
- LEDs (Verde, Amarelo, Vermelho)
- Resistores
- Plataforma Wokwi para simulação
- Broker MQTT (broker.hivemq.com)
- Node-RED

### Lógica de Funcionamento

1. O sensor ultrassônico mede a distância.
2. De acordo com a distância, um LED acende:
  - **Verde**: > 300 cm "Tudo certo"
  - **Amarelo**: entre 100 cm e 300 cm "Ficar atento"
  - **Vermelho**: < 100 cm "Crítico"
3. Os dados são enviados em formato JSON via MQTT:

```
{  
  "distancia": 97,  
  "status": "Crítico",  
  "led": "vermelho"  
}
```

### Configuração do Wi-Fi no Wokwi

## Documentação do Projeto ESP32 - Sensor Ultrassônico

Certifique-se de usar o seguinte trecho no `setup\_wifi()`:

```
WiFi.begin("Wokwi-GUEST", "", 6); // Canal 6 obrigatório
```

## Dashboard Node-RED

### Fluxo

- MQTT IN (topico: `agua/monitoramento`)
- JSON
- Function: separa distância e status
- ui\_text: exibe distância
- ui\_text: exibe status com cores (vermelho/lime/laranja)

### Function para distância

```
msg.payload = "Distância: " + msg.payload.distancia + " cm";  
return msg;
```

### Function para status

```
let status = msg.payload.status;  
msg.payload = "Status do LED: " + status;  
  
if (status === "Crítico") {  
    msg.color = "#ff0000";  
    msg.bold = true;  
} else if (status === "Ficar atento") {  
    msg.color = "#ff9900";  
    msg.bold = true;  
} else {  
    msg.color = "#00cc00";  
    msg.bold = true;  
}  
  
return msg;
```

# **Documentação do Projeto ESP32 - Sensor Ultrassônico**

## **Autor**

Evandro Yuji Kaibara de Oliveira RM: 559274

Mateus da Rocha Mallet RM: 560491