

CESAR SCHOOL

João Ricardo L. Barros

Patrick Edward Costa Catchpole

Matheus Vellez Bione Botelho

MONITORAMENTO DE TEMPERATURA UTILIZANDO ESP32S2 E  
COMUNICAÇÃO VIA MQTT PARA UM SERVIDOR BROKER

Recife – PE

Junho de 2025

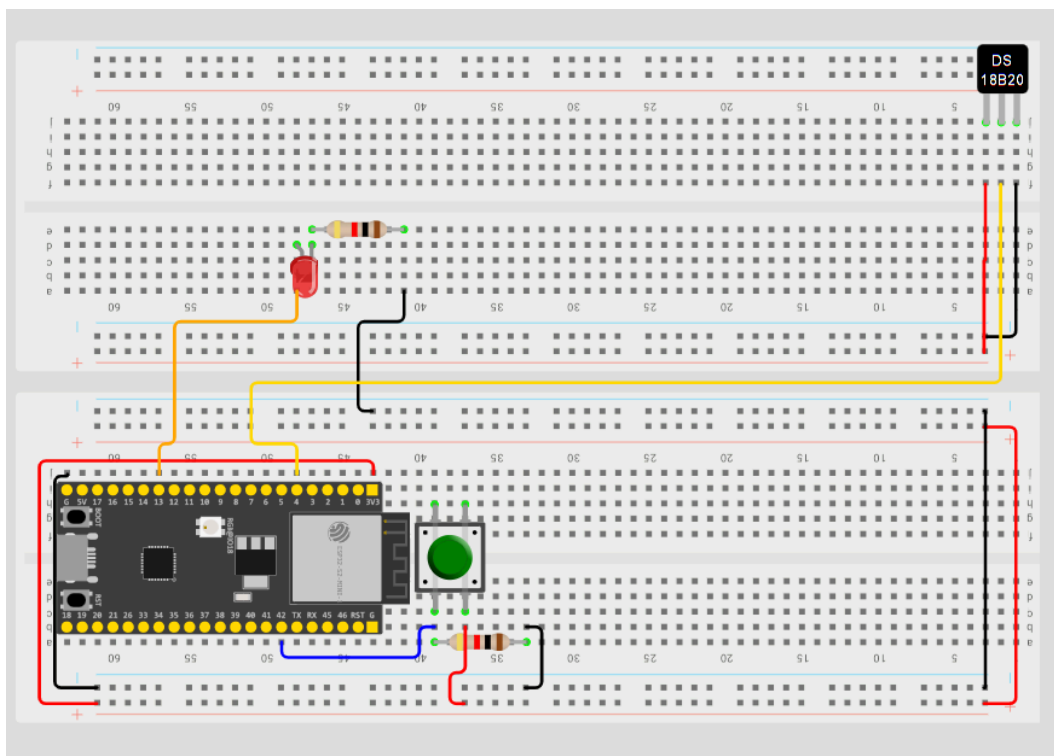
## 1. Introdução

Este projeto propõe a implementação de um sistema de monitoramento de temperatura utilizando uma placa ESP32S2, com um sensor de temperatura acoplado. A ESP32S2, equipada com conectividade Wi-Fi, é responsável por coletar periodicamente os dados do sensor e enviá-los para um servidor broker através do protocolo de comunicação MQTT (Message Queuing Telemetry Transport). O broker que é encontrado em outra máquina, pode acionar alertas ou registrar os dados conforme limites pré-definidos de temperatura.

A motivação para este trabalho surge da necessidade de soluções acessíveis e eficientes para monitoramento de ambientes, especialmente em locais que demandam controle térmico preciso, como laboratórios, estufas, data centers ou ambientes industriais. A escolha da ESP32S2 se deve ao seu baixo custo e versatilidade, enquanto o uso do MQTT garante uma comunicação leve e eficiente entre dispositivos, característica essencial para sistemas IoT.

## 2. Metodologia

O projeto foi desenvolvido com a intenção de utilizar um microcontrolador (ESP32-S2) para poder medir a temperatura de qualquer ambiente utilizando um sensor de temperatura (LM35DZ) onde as informações coletadas por esse sensor são enviadas para um servidor broker via MQTT onde elas são tratadas de forma que possam alertar o usuário de possíveis variações da temperatura que possam trazer riscos ao contexto inserido. Abaixo encontra-se o esquema elétrico da ESP32-S2 conectada ao sensor :



## Hardware utilizado

- Placa ESP32-S2.
- Computador pessoal (ou Raspberry pi com WiFi).
- Sensor de temperatura LM35DZ.
- Led.
- Botão.

## 3. Resultados

O ESP32S2 consegue captar as informações coletadas pelo sensor de temperatura, e processá-las para que sejam legíveis para os usuários, é possível imprimir os valores em um terminal serial, para garantir o funcionamento correto da lógica.

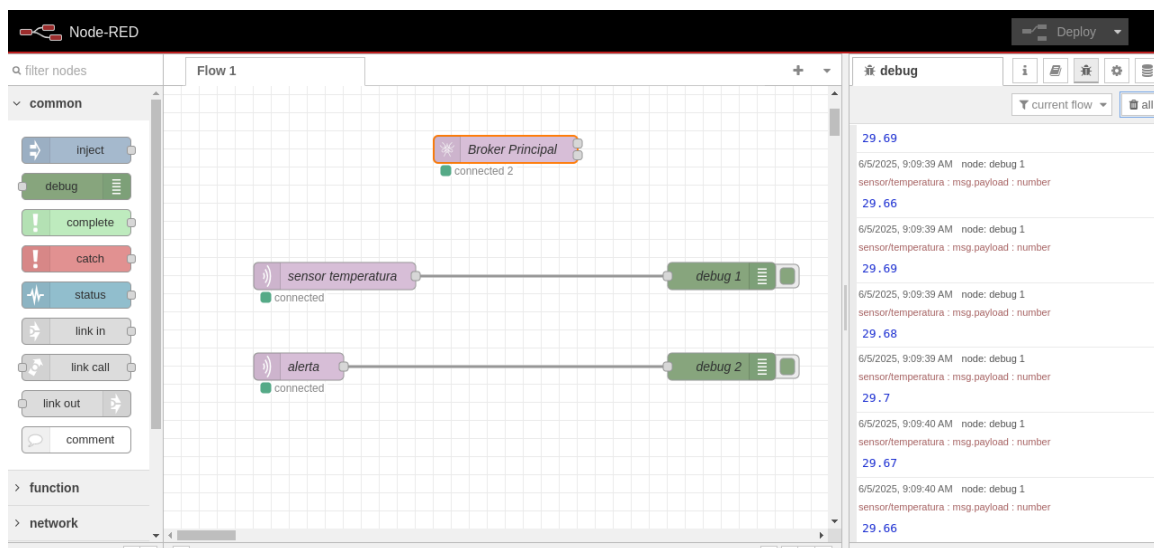
```
Iniciando o Setup!
SERVIDOR BROKER EM: 192.168.88.146
[+] Pino de botao iniciados.
[+] Pino do LED inicializado.

Conectando-se a tron
.....
WiFi conectado.
IP: 192.168.88.78
Precione o botão para continuar!
```

Antes de iniciar a leitura o ESP32S2 faz as configurações iniciais para o funcionamento correto do equipamento, como conectar a uma rede WiFi determinada, e cadastrar o servidor Broker MQTT que será utilizado para receber as mensagens. Após esse setup inicial, o equipamento espera o usuário apertar o botão para iniciar o processo de leitura dos valores recebidos pelo sensor, e realizar o tratamento devido para então enviar para o servidor MQTT.

```
Precione o botão para continuar!  
Finalizando o Setup!  
Tentando conexão MQTT...conectado!  
Temperature: 30.06 °C  
Temperature: 30.06 °C  
Temperature: 30.14 °C  
Temperature: 30.22 °C  
Temperature: 30.14 °C  
Temperature: 29.90 °C  
Temperature: 30.30 °C  
Temperature: 30.14 °C  
Temperature: 29.90 °C
```

Após pressionado o botão, o ESP32S2 tenta se conectar ao servidor MQTT, que é um broker aedes que se encontra no node-red, sendo acessado pela porta 1883, onde uma vez que a conexão é estabelecida, o ESP32S2 envia as mensagens contendo a média das temperaturas obtidas em um certo intervalo de tempo, onde as informações demoram cerca de 1s a 3s para chegarem ao servidor.



O ESP32S2 também está programado para enviar um alerta ao servidor caso a temperatura obtida ultrapasse um determinado valor, assim é possível notificar o usuário de que algum problema está ou pode vir a acontecer se não for tomada nenhuma providência.

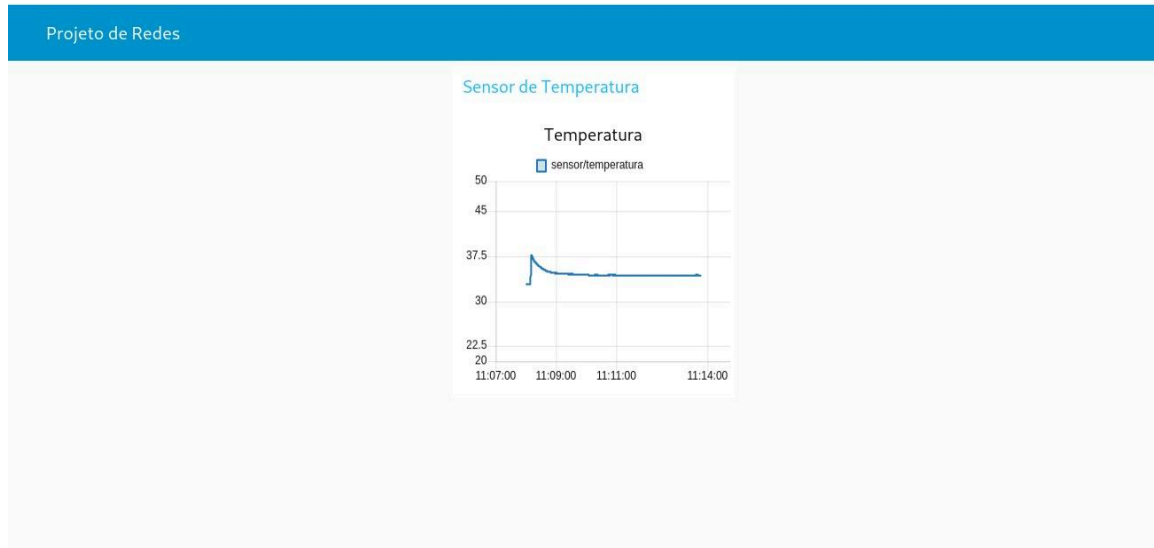
```
Temperature: 33.68 °C
Temperature: 33.68 °C
Temperature: 33.77 °C
Temperature: 33.77 °C
Temperature: 34.49 °C
Temperature: 33.93 °C
      ATENCAO!
TEMPERATURA ACIMA DOS NIVEIS NORMAIS!
Temperature: 37.88 °C
      ATENCAO!
TEMPERATURA ACIMA DOS NIVEIS NORMAIS!
Temperature: 38.36 °C
      ATENCAO!
TEMPERATURA ACIMA DOS NIVEIS NORMAIS!
Temperature: 38.68 °C
```

The screenshot shows a Node-RED debug console window titled "debug". The interface includes a toolbar with icons for information, clipboard, delete, settings, and a dropdown menu. Below the toolbar, there is a filter dropdown set to "current flow" and a "trash all" button. The console displays a list of messages with timestamps and node identifiers. The messages show temperature readings and alert messages in Portuguese.

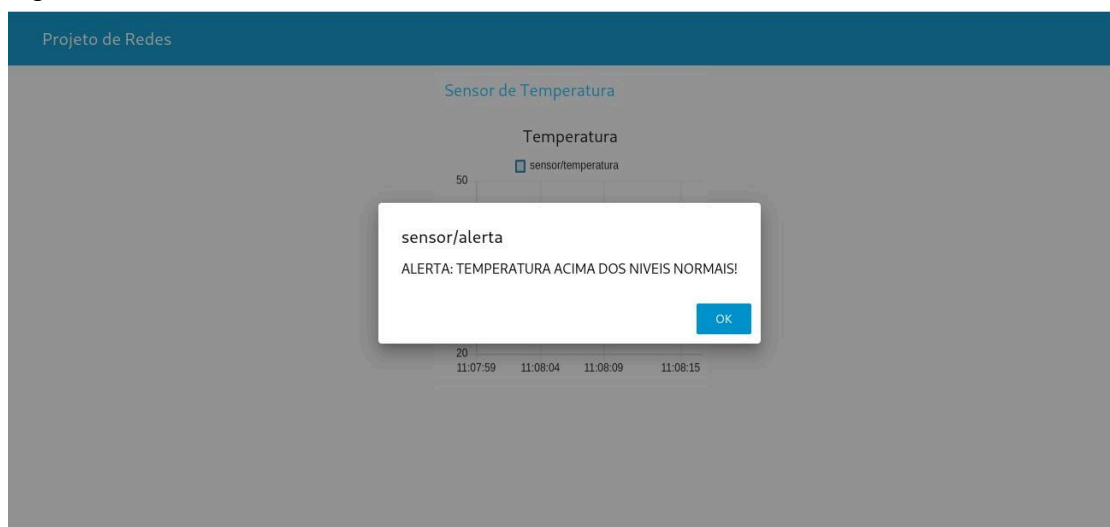
Timestamp	Node	Message
6/5/2025, 9:27:32 AM	node: debug 1	sensor/temperatura : msg.payload : number 33.77
6/5/2025, 9:27:32 AM	node: debug 1	sensor/temperatura : msg.payload : number 33.92
6/5/2025, 9:27:32 AM	node: debug 2	sensor/alerta : msg.payload : string[45] "ALERTA: TEMPERATURA ACIMA DOS NIVEIS NORMAIS!"
6/5/2025, 9:27:33 AM	node: debug 1	sensor/temperatura : msg.payload : number 36.05
6/5/2025, 9:27:33 AM	node: debug 2	sensor/alerta : msg.payload : string[45] "ALERTA: TEMPERATURA ACIMA DOS NIVEIS NORMAIS!"
6/5/2025, 9:27:34 AM	node: debug 1	sensor/temperatura : msg.payload : number

## 4. Interface Gráfica

Embora inicialmente não tenhamos conseguido implementar uma interface gráfica própria para exibição dos dados, optamos por utilizar o **Node-RED Dashboard**, que se mostrou uma solução eficiente e prática para **visualização em tempo real**. Através dessa plataforma, foi possível criar uma interface amigável, que exibe graficamente a temperatura medida e apresenta alertas visuais quando o valor ultrapassa o limite definido.



Para evitar alertas excessivos (spamming), implementamos no código da ESP32-S2 uma lógica com **uso de flag**, de modo que o alerta seja enviado apenas **uma vez** quando a temperatura ultrapassa o limite, e um novo alerta apenas quando a temperatura retornar ao normal. Isso reduziu o ruído nas notificações e melhorou significativamente a experiência do usuário.



```

if(temperatureMean > 34 && !flag)
{
    Serial.println("          ATENCAO!");
    Serial.println("TEMPERATURA ACIMA DOS NIVEIS NORMAIS!");
    digitalWrite(ledOutput, HIGH);
    client.publish("sensor/alerta", "ALERTA: TEMPERATURA ACIMA DOS NIVEIS NORMAIS!");
    flag = true;
    delay(1000);
    resetFlag = millis();
}
else if(temperatureMean <= 34)
    digitalWrite(ledOutput, LOW);
temperatureMean = 0.0;
cycles = 0;
}
else
{
    temperatureMean = temperatureMean + temperatureC;
    cycles++;
    resetFlag++;
}

if(resetFlag - millis() >= 60000 && flag)
{
    resetFlag = millis();
    flag = false;
}
}

```

## 5. Conclusão

Durante a execução desse trabalho foram encontrados diversos problemas para a configuração da comunicação entre a ESP32S2 e o servidor broker, pois realizar as configurações do servidor foram mais difíceis do que o esperado. Entretanto o desenvolvimento do software para a ESP32S2 foi relativamente simples já que o sensor possui boa compatibilidade com a ESP32S2, permitindo uma simples montagem do circuito elétrico, além de bom monitoramento dos valores recebidos, porém é necessário calibrar os valores recebidos do sensor.

A comunicação MQTT é muito boa quando utilizado o ESP32S2, sofrendo poucas instabilidades, garantindo uma comunicação consistente.

## 6. Apêndice

GitHub: [Código da ESP32-S2](#)