



**UNIVERSIDADE FEDERAL DE SANTA
CATARINA – UFSC**

Departamento de Computação – Engenharia de Computação

Disciplina: Redes sem Fios

Professor: Jayor Nesi Teixeira

**Estação Meteorológica Inteligente
com Alerta Remoto baseada em
ESP32 e MQTT**

Abraão Sacaia
João Sebastião
Marcolino de Lemos

Local e ano: Araranguá – SC, 2025

Sumário

1	Resumo	2
2	Objetivos	2
3	Materiais e métodos	2
3.1	Componentes utilizados	2
3.2	Arquitetura do sistema	2
3.3	Software e desenvolvimento	3
4	Passos executados	3
4.1	Montagem física	3
4.2	Configuração e gravação	3
4.3	Testes de comunicação	3
4.4	Testes funcionais dos sensores e alertas	3
5	Resultados	4
6	Discussão	4
7	Conclusão	4

1 Resumo

Este relatório apresenta o desenvolvimento completo de uma estação meteorológica inteligente com alerta remoto utilizando dois módulos ESP32 conectados via Wi-Fi e comunicando-se por MQTT. O nó sensor realiza a leitura de temperatura, umidade, pressão atmosférica, chuva e radiação solar, enquanto o nó receptor subscreve os tópicos de telemetria, exibe os dados e aciona alertas visuais e sonoros conforme limiares configurados. São descritos a arquitetura final, os detalhes de hardware e software, os testes realizados em laboratório/ambiente externo e os resultados obtidos, bem como limitações e possibilidades de melhorias futuras.

2 Objetivos

- Validar na prática uma rede sem fio entre dois ESP32 usando MQTT para envio e recepção de dados de sensores ambientais.
- Integrar completamente sensores físicos (temperatura, umidade, pressão, chuva, radiação) ao nó sensor e atuadores (LEDs e buzzer) ao nó receptor.
- Configurar limiares de alerta para diferentes condições climáticas e verificar o acionamento correto dos atuadores.
- Avaliar robustez de comunicação, reconexão ao broker e comportamento do sistema em diferentes cenários de teste.

3 Materiais e métodos

3.1 Componentes utilizados

- 2 × ESP32 (nó sensor e nó receptor).
- 1 × DHT11 (temperatura e umidade) e 1 × BMP280/BME280 (pressão e, opcionalmente, temperatura/umidade).
- 1 × sensor de chuva (digital/analógico).
- LEDs de alerta (verde, amarelo, vermelho) .
- Protoboard, cabos, resistores e fonte 5 V/USB.
- Broker MQTT (por exemplo, `test.mosquitto.org` ou instância local).

3.2 Arquitetura do sistema

- **Nó sensor:** conecta ao Wi-Fi, lê ciclicamente os sensores e publica uma mensagem JSON em um tópico de telemetria (por exemplo, `estacao/telemetria`).
- **Nó receptor:** conecta ao mesmo Wi-Fi, subscreve o tópico, desserializa o JSON, exibe os dados no monitor serial ou display OLED e compara com limiares para acionar LEDs e buzzer.
- **Broker MQTT:** intermedia a comunicação publish/subscribe entre os nós, podendo ser público ou local.

3.3 Software e desenvolvimento

- **Ambiente:** ESP-IDF ou Arduino IDE/PlatformIO, com bibliotecas para MQTT, sensores (DHT, BMP/BME), JSON (cJSON) e, opcionalmente, display OLED.
- **Código do nó sensor:**
 - Inicialização de Wi-Fi e MQTT.
 - Leitura dos sensores, tratamento de erro e cálculo de médias se necessário.
 - Montagem de payload JSON com campos como `temp`, `umid`, `pres`, `chuva`, `radiacao` e envio periódico ao broker.
- **Código do nó receptor:**
 - Conexão a Wi-Fi e subscrição do tópico.
 - Callback de recepção MQTT para decodificar o JSON e atualizar variáveis internas e o display.
 - Função de verificação de limiares que define estados de alerta (normal, atenção, crítico) e aciona LEDs e buzzer.

4 Passos executados

4.1 Montagem física

- Ligação do DHT22, BMP280/BME280, sensor de chuva e sensor de radiação solar ao ESP32 respeitando alimentação, pinos de dados e I2C.
- Conexão de LEDs com resistores e buzzer aos pinos configurados no nó receptor.

4.2 Configuração e gravação

- Definição de SSID, senha de Wi-Fi, endereço e porta do broker MQTT e tópicos em um arquivo de configuração.
- Compilação e upload do firmware de sensor e receptor para os respectivos ESP32.

4.3 Testes de comunicação

- Verificação da conexão de ambos os nós à rede Wi-Fi e ao broker, monitorando mensagens de conexão e reconexão.
- Verificação da publicação e recepção de payloads MQTT, comparando logs de server e receiver para conferir consistência das leituras.

4.4 Testes funcionais dos sensores e alertas

- Variação proposital de temperatura e umidade (aquecendo ou resfriando o sensor) para testar limiares de alerta.
- Simulação de chuva (umidificação do sensor de chuva) e alteração da incidência de luz no sensor de radiação para observar mudanças nas leituras e acionamento dos LEDs.

- Desconexão manual do Wi-Fi/broker para avaliar o comportamento da lógica de reconexão.

5 Resultados

- A comunicação MQTT entre o nó sensor e o nó receptor funcionou de forma estável, com recepção contínua das mensagens e latência adequada para monitoramento em tempo quase real.
- A montagem física dos sensores permitiu obter leituras coerentes com as condições ambientais observadas, dentro da faixa de precisão esperada para DHT22 e BMP280/BME280.
- Os limiares configurados resultaram em acionamentos corretos dos LEDs e do buzzer em cenários de temperatura elevada, umidade muito baixa ou alta e presença de chuva.
- Em testes de reconexão, o sistema retornou à operação normal após perdas temporárias de Wi-Fi, embora tenham sido observados pequenos atrasos até o restabelecimento da sessão MQTT.

6 Discussão

A integração completa hardware–software demonstrou a viabilidade do uso de ESP32 e MQTT para monitoramento ambiental de baixo custo.

Limitações observadas incluem sensibilidade dos sensores às condições de instalação (ventilação, exposição ao sol direto), necessidade de calibração mais refinada e eventual uso de encapsulamentos adequados para instalação externa.

Melhorias futuras incluem: armazenamento dos dados em um servidor ou banco de dados para histórico, desenvolvimento de uma interface web ou dashboard, uso de TLS no MQTT e refinamento dos limiares com base em séries temporais de dados.

7 Conclusão

O projeto atingiu o objetivo de implementar uma estação meteorológica inteligente funcional, com coleta de múltiplas variáveis ambientais e alertas remotos baseados em MQTT entre dois ESP32. O protótipo mostrou-se adequado como ferramenta didática para a disciplina de Redes sem Fios, permitindo a exploração prática de conceitos de IoT, protocolos de aplicação e integração de sensores e atuadores.