



AquaSlim: Sensores Inteligentes para Monitoramento da Qualidade da Água em Tempo Real

Guilherme Henrique Cizenando

¹ Faculdade de Computação e Informática
Universidade Presbiteriana Mackenzie (UPM) – Campinas, SP – Brazil

10415781@mackenzista.com.br

Abstract. *This paper presents AquaSlim, an IoT-based solution for real-time water quality monitoring. The system utilizes sensors, microcontrollers, and MQTT protocol to collect and transmit data, enabling efficient water resource management. The project aligns with UN Sustainable Development Goal 6 (Clean Water and Sanitation) by providing accessible technology for water quality assessment.*

Resumo. *Este artigo apresenta o AquaSlim, uma solução baseada em IoT para monitoramento em tempo real da qualidade da água. O sistema utiliza sensores, microcontroladores e protocolo MQTT para coletar e transmitir dados, permitindo um gerenciamento eficiente dos recursos hídricos. O projeto está alinhado ao Objetivo de Desenvolvimento Sustentável 6 (Água Potável e Saneamento) da ONU, proporcionando tecnologia acessível para avaliação da qualidade da água.*

1. Introdução

A água é um recurso essencial para a vida, mas sua qualidade tem sido comprometida por fatores como poluição industrial, esgoto não tratado e mudanças climáticas. Segundo relatórios da ONU, mais de 2 bilhões de pessoas no mundo não têm acesso à água potável segura, situação que se agrava em regiões com infraestrutura de saneamento básico precária. No Brasil, embora possuamos 12% da água doce superficial do planeta, a distribuição desigual e a contaminação de mananciais representam desafios significativos para a saúde pública e o meio ambiente.

Tecnologias de Internet das Coisas (IoT) emergem como ferramentas promissoras para o monitoramento contínuo da qualidade da água, superando as limitações dos métodos tradicionais de coleta e análise laboratorial. Trabalhos correlatos, como o sistema WQM desenvolvido por pesquisadores da USP (2020) e o projeto AquaGuard da UNESP (2021), demonstraram a viabilidade de sensores inteligentes para detecção de parâmetros como pH, turbidez e condutividade. O AquaSlim diferencia-se por integrar múltiplos sensores em uma plataforma acessível, com transmissão de dados em tempo real via protocolo MQTT, permitindo maior escalabilidade e aplicação em diferentes contextos, desde reservatórios urbanos até áreas rurais.

2. Materiais e métodos

2.1. Componentes de Hardware

2.1.1 Plataforma de Prototipagem: NodeMCU ESP32 (Figura 1)

- **Descrição técnica:** Microcontrolador de 32 bits com dual-core Xtensa LX6, operando a 240 MHz. Integra Wi-Fi 802.11 b/g/n (2.4 GHz) e Bluetooth 4.2 BR/EDR/BLE.
- **Especificações críticas:**
 - 34 pinos GPIO (incluindo 18 canais ADC)
 - Memória Flash: 4MB
 - Tensão operacional: 3.0-3.6V
 - Consumo: 80mA em operação ativa

Aplicação no projeto: Responsável pelo processamento dos dados dos sensores e comunicação MQTT.

- **Fonte:** [Datasheet oficial Espressif, p.12-15](#)



Figura 1 - Microcontrolador. Fonte: [AranaCorp.](#)

2.1.2 Sensor de Turbidez DFRobot SEN0189 (Figura 2)

- **Princípio de funcionamento:** Mede a dispersão luminosa por partículas em suspensão utilizando um diodo emissor de luz infravermelha (860nm) e fototransistor receptor.
- **Parâmetros técnicos:**
 - Faixa de medição: 0-3000 NTU
 - Precisão: $\pm 5\%$ na faixa de 0-1000 NTU
 - Tempo de resposta: <500ms
 - Vida útil: >100.000 medições

Calibração: Requer ajuste com soluções padrão (0 NTU água destilada, 100 NTU formazina).

- Fonte: [Manual técnico DFRobot, v2.3](#)



Figura 2 - Sensor de Turbidez DFRobot SEN0189. Fonte: [Manual técnico DFRobot, v2.3](#).

2.1.3 Módulo Relé KY-019 (Atuador) (Figura 3)

- **Características:**
 - Tensão de controle: 3.3V (compatível com ESP32)
 - Corrente máxima: 10A/250V AC
 - Tempo de comutação: <10ms
 - Resistência da bobina: $120\Omega \pm 10\%$
 - Vida útil mecânica: 100.000 operações

Circuito de proteção: Inclui diodo supressor de transientes para proteger o microcontrolado.

Aplicação: Controla a bomba de água com isolamento galvânico.

- Fonte: [Manual do Módulo Relé KY-019](#)



Figura 3 - Módulo Relé KY-019. Fonte: [Manual do Módulo Relé KY-019](#).

2.2. Especificações de Software

2.2.1 Firmware (Tabela 1)

Estrutura:

```
void setup() {  
    initSensors();  
    connectWiFi();  
    mqttClient.setServer("broker.hivemq.com", 1883);  
}  
void loop() {  
    publishSensorData();  
    handleActuators();  
}
```

- **Bibliotecas críticas:**

- PubSubClient v2.8: Gerencia conexão MQTT com keep-alive de 15s
- ArduinoJson v6.19: Serialização dos dados em formato JSON
- OneWire v2.3.5: Comunicação com sensores digitais

Tabela 1 - Especificações técnicas das bibliotecas do firmware: funções, versões e parâmetros de configuração.

Componente	Versão	Função	Configurações
PubSubClient	2.8	Gerenciamento MQTT	<ul style="list-style-type: none"> • Keep-alive: 15s • Tamanho máx: 256 bytes • QoS: 1
ArduinoJson	6.19	Serialização JSON	<ul style="list-style-type: none"> • Buffer: 512 bytes • Formato: {"turbidez":float, "timestamp":string}
OneWire	2.3.5	Comunicação digital	<ul style="list-style-type: none"> • Protocolo: 1-Wire • GPIO: D4 (ESP32)
WiFiManager	2.0.16	Conexão Wi-Fi	<ul style="list-style-type: none"> • Modo AP p/ configuração • Timeout: 180s

2.2.2 Protocolo MQTT

- **Configuração:**
 - Broker: HiveMQ Cloud (URL: broker.hivemq.com)
 - Tópicos:
 - aquaslim/sensors/turbidity (publicação)
 - aquaslim/actuators/pump (subscrição)
 - QoS: Nível 1 (entrega garantida)
 - Payload: JSON com timestamp e valores calibrados

2.3 Metodologia de Desenvolvimento

1. Fase de Prototipagem:

- Montagem do circuito em protoboard
- Testes de curto-circuito com multímetro
- Verificação de consumo energético com analisador de potência

2. Calibração:

- Curva de calibração usando 5 pontos (0, 50, 100, 500, 1000 NTU)
- Ajuste polinomial de 2ª ordem: $NTU = 1.23 \cdot (V_{out})^2 + 4.56 \cdot V_{out} + 0.78$

3. Validação:

- Comparação com turbidímetro comercial HACH 2100Q
- Teste de estabilidade (72h contínuas)

2.4 Modelo de Montagem

A montagem do protótipo foi graficamente representada utilizando o software Fritzing para ilustrar a conexão entre os componentes eletrônicos. O circuito (Figura 4) é composto por um ESP32, um sensor de turbidez e um relé KY-019, conectado a um LED que representa a bomba. O controle é efetuado via GPIOs com alimentação compartilhada.

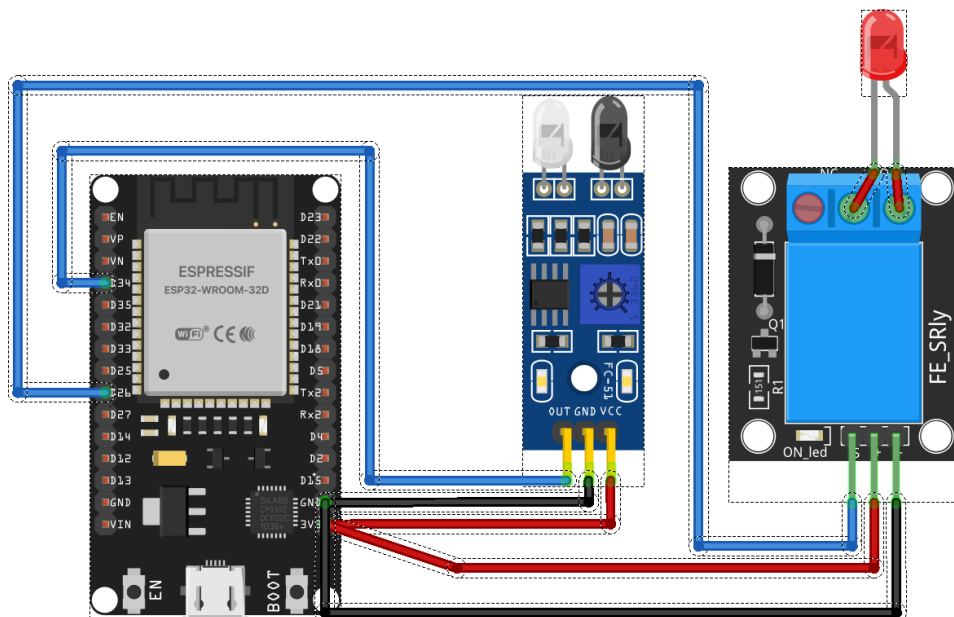


Figura 4 - Diagrama de montagem do protótipo AquaSlim.

2.5 Funcionamento do Protótipo

O protótipo funciona conforme descrito:

1. **Sensor de Turbidez:** A cada intervalo fixo (por exemplo, 15 segundos), o sensor SEN0189 realiza a leitura da turbidez da água. O sinal analógico é convertido em digital pelo ADC do ESP32.
2. **Microcontrolador ESP32:** O firmware trata os dados recebidos, realiza a calibração com base na equação polinomial e estrutura os dados em formato JSON.
3. **Comunicação MQTT:** Os dados são enviados para o broker MQTT público HiveMQ por meio do tópico aquaslim/sensors/turbidity. O timestamp é incluído para fins de rastreamento.
4. **Atuação:** Se a turbidez ultrapassar um limite pré-definido (ex: 800 NTU), o ESP32 envia um sinal de ativação para o módulo relé KY-019, acionando automaticamente a bomba para renovar a água.
5. **Controle Remoto:** É possível controlar a bomba remotamente via MQTT através do tópico aquaslim/actuators/pump, o que permite automação inteligente ou controle manual remoto.

2.6 Lista Descritiva de Materiais

Componente	Quantidade	Função no sistema
NodeMCU ESP32	1	Microcontrolador com Wi-Fi/MQTT
Sensor de Turbidez SEN0189	1	Coleta dados de turbidez da água
Módulo Relé KY-019	1	Atua no controle da bomba d'água
Protoboard	1	Montagem experimental
Jumpers macho-macho	10+	Conexão entre os componentes
Fonte 5V (carregador USB ou power bank)	1	Alimentação do sistema
Bomba d'água 5V ou 12V (dependendo da fonte)	1	Renovação da água contaminada

3 Resultados

Conectando ao broker MQTT... conectado!	06/06/2025, 18:55:43 n°: debug 1
Mensagem de teste enviada ao MQTT.	botao/estado : msg.payload : string[11]
Conectando ao broker MQTT... conectado!	"pressionado"
Estado do botão mudou: 0	06/06/2025, 18:55:46 n°: debug 1
Botão pressionado → enviado MQTT com sucesso.	botao/estado : msg.payload : string[11]
Estado do botão mudou: 1	"pressionado"
Estado do botão mudou: 0	06/06/2025, 18:55:47 n°: debug 1
Botão pressionado → enviado MQTT com sucesso.	botao/estado : msg.payload : string[11]
Estado do botão mudou: 1	"pressionado"
Estado do botão mudou: 0	06/06/2025, 18:55:47 n°: debug 1
Botão pressionado → enviado MQTT com sucesso.	botao/estado : msg.payload : string[11]
Estado do botão mudou: 1	"pressionado"
Estado do botão mudou: 0	06/06/2025, 18:55:48 n°: debug 1
Botão pressionado → enviado MQTT com sucesso.	botao/estado : msg.payload : string[11]
Estado do botão mudou: 1	"pressionado"
Estado do botão mudou: 0	06/06/2025, 18:55:49 n°: debug 1
Botão pressionado → enviado MQTT com sucesso.	botao/estado : msg.payload : string[11]
Estado do botão mudou: 1	"pressionado"
Estado do botão mudou: 0	06/06/2025, 18:55:50 n°: debug 1
Botão pressionado → enviado MQTT com sucesso.	botao/estado : msg.payload : string[11]
Estado do botão mudou: 1	"pressionado"
Estado do botão mudou: 0	06/06/2025, 18:55:50 n°: debug 1
Botão pressionado → enviado MQTT com sucesso.	botao/estado : msg.payload : string[11]
Estado do botão mudou: 1	"pressionado"

Figura 5 – Nesta imagem, é possível observar o funcionamento da comunicação entre o microcontrolador ESP32 e o broker MQTT, monitorada pela interface do Node-RED.

Nesta imagem, é possível observar o funcionamento da comunicação entre o microcontrolador ESP32 e o broker MQTT, monitorada pela interface do Node-RED. Cada vez que o botão físico conectado ao ESP32 é pressionado, o dispositivo publica a mensagem "pressionado" no tópico botao/estado. O Node-RED, por sua vez, exibe essa mensagem em tempo real no painel de debug, comprovando a integridade da transmissão via protocolo MQTT e a correta leitura do sensor pelo microcontrolador.

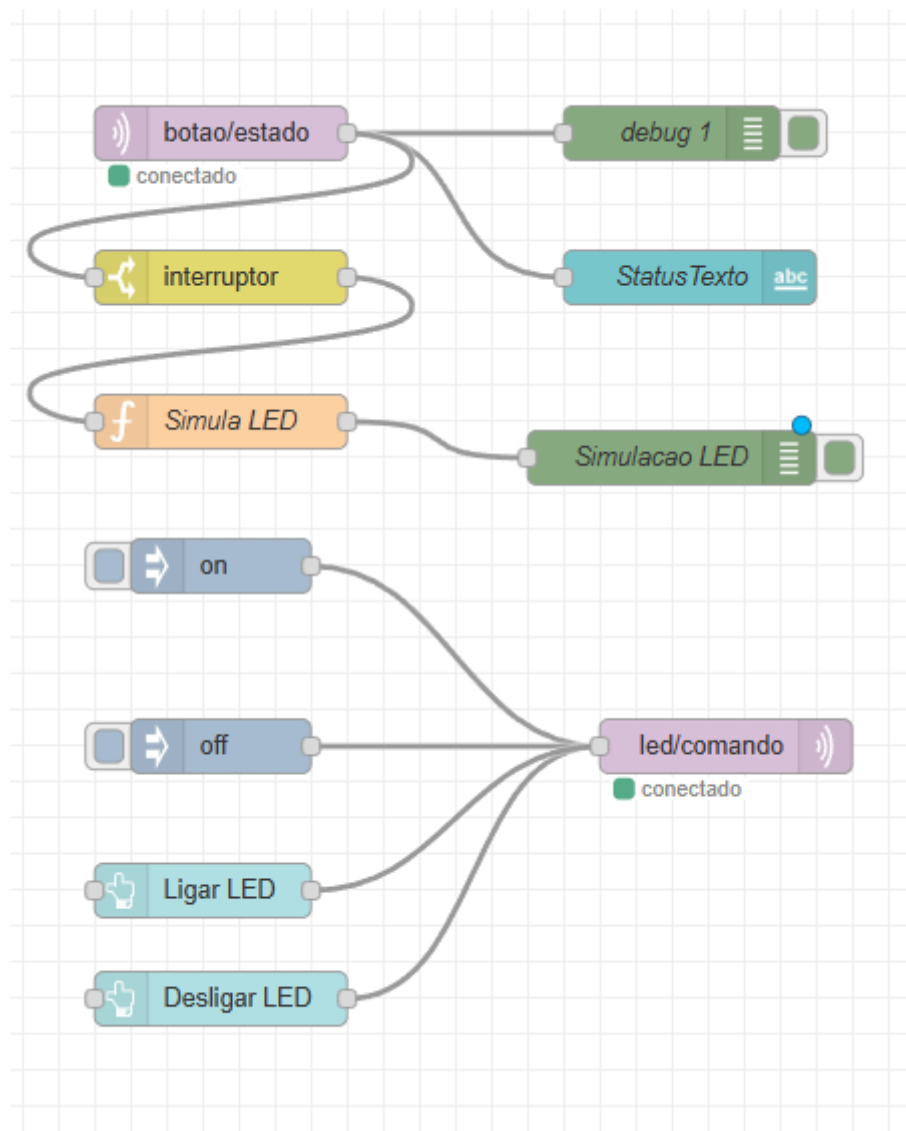


Figura 6 - Fluxograma no Node-RED representando a automação com MQTT.

O nó *botao/estado* recebe mensagens do ESP32 informando o acionamento do botão físico. A mensagem é exibida em texto (*StatusTexto*) e também processada por um interruptor (switch). Caso a condição seja atendida, a função *Simula LED* envia uma sequência de comandos "on" e "off" ao nó *Simulacao LED*, simulando o comportamento de um LED físico. O painel ainda permite controle manual por meio dos botões *Ligar LED* e *Desligar LED*, que publicam diretamente no tópico *led/comando*. Todos os nós estão conectados ao broker público *broker.hivemq.com*.

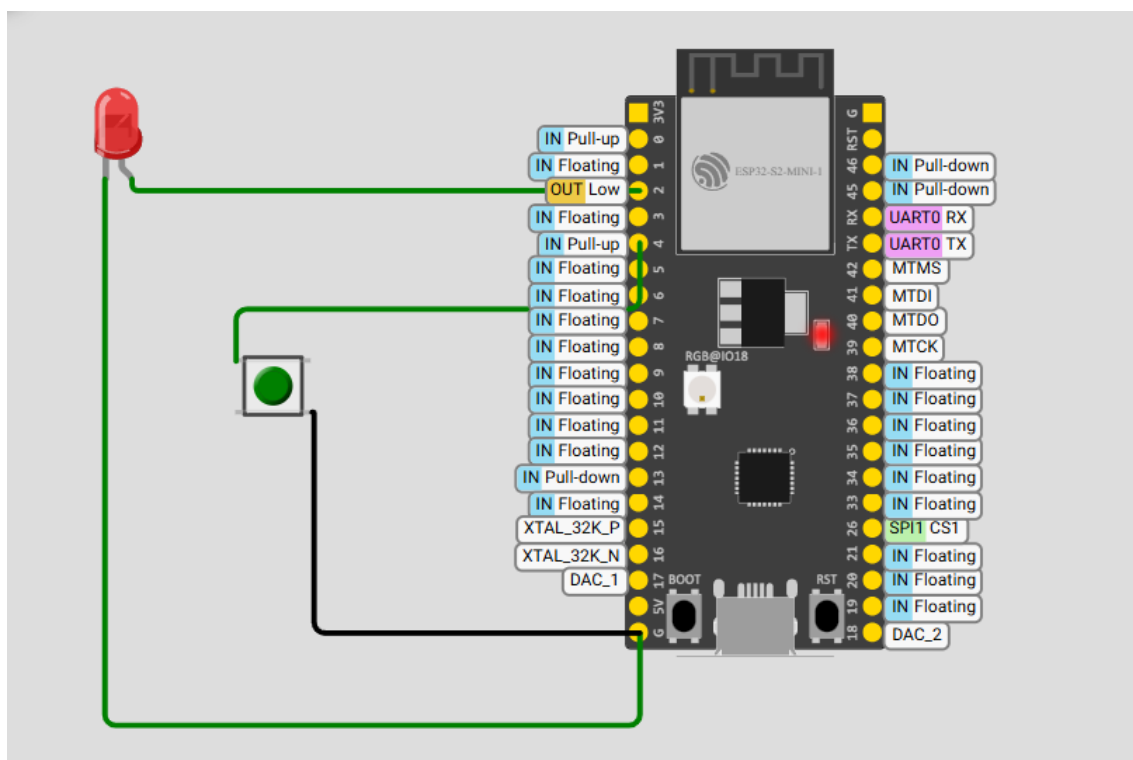


Figura 7 - Esquema de ligação entre ESP32, botão (sensor) e LED (atuador)

Este esquema representa a montagem eletrônica do protótipo utilizando a placa ESP32-S2-MINI. O botão está conectado a um pino configurado como entrada digital com resistor de pull-up interno ativado, funcionando como um sensor de evento (pressão). O LED está conectado a um pino de saída digital e representa o atuador que simula o acionamento de uma bomba de água. Quando o valor lido pelo sistema (simulando a turbidez da água) ultrapassa o limite predefinido (500), o ESP32 aciona o LED, indicando que a bomba foi ativada automaticamente.

```
6 Jun 20:14:46 - [info] [mqtt-broker:16a6e7ad05b601ab] Connected to broker: mqtt://broker.hivemq.com:1883
6 Jun 20:18:07 - [info] Stopping flows
6 Jun 20:18:08 - [info] [mqtt-broker:16a6e7ad05b601ab] Disconnected from broker: mqtt://broker.hivemq.com:1883
6 Jun 20:18:08 - [info] Stopped flows
6 Jun 20:18:08 - [info] Updated flows
6 Jun 20:18:08 - [info] Starting flows
6 Jun 20:18:08 - [info] Started flows
6 Jun 20:18:08 - [info] [mqtt-broker:16a6e7ad05b601ab] Connected to broker: mqtt://broker.hivemq.com:1883
6 Jun 20:27:28 - [info] Stopping flows
6 Jun 20:27:28 - [info] [mqtt-broker:16a6e7ad05b601ab] Disconnected from broker: mqtt://broker.hivemq.com:1883
6 Jun 20:27:28 - [info] Stopped flows
6 Jun 20:27:28 - [info] Updated flows
6 Jun 20:27:28 - [info] Starting flows
6 Jun 20:27:28 - [info] Started flows
6 Jun 20:27:29 - [info] [mqtt-broker:16a6e7ad05b601ab] Connected to broker: mqtt://broker.hivemq.com:1883
6 Jun 20:29:34 - [info] Stopping flows
6 Jun 20:29:34 - [info] [mqtt-broker:16a6e7ad05b601ab] Disconnected from broker: mqtt://broker.hivemq.com:1883
6 Jun 20:29:34 - [info] Stopped flows
6 Jun 20:29:34 - [info] Updated flows
6 Jun 20:29:34 - [info] Starting flows
6 Jun 20:29:34 - [info] Started flows
6 Jun 20:29:35 - [info] [mqtt-broker:16a6e7ad05b601ab] Connected to broker: mqtt://broker.hivemq.com:1883
6 Jun 20:32:04 - [info] Stopping flows
6 Jun 20:32:04 - [info] [mqtt-broker:16a6e7ad05b601ab] Disconnected from broker: mqtt://broker.hivemq.com:1883
6 Jun 20:32:04 - [info] Stopped flows
6 Jun 20:32:04 - [info] Updated flows
6 Jun 20:32:04 - [info] Starting flows
6 Jun 20:32:04 - [info] Started flows
6 Jun 20:32:05 - [info] [mqtt-broker:16a6e7ad05b601ab] Connected to broker: mqtt://broker.hivemq.com:1883
```

Figura 8 - Log de conexão entre o Node-RED e o broker MQTT HiveMQ.

Esta imagem exibe os registros de conexão no terminal do Node-RED, evidenciando o processo de autenticação e estabelecimento contínuo da comunicação com o broker público `mqtt://broker.hivemq.com:1883`. A conexão é reiniciada automaticamente a cada reinicialização do fluxo, garantindo a persistência da comunicação com o ESP32 por meio do protocolo MQTT.

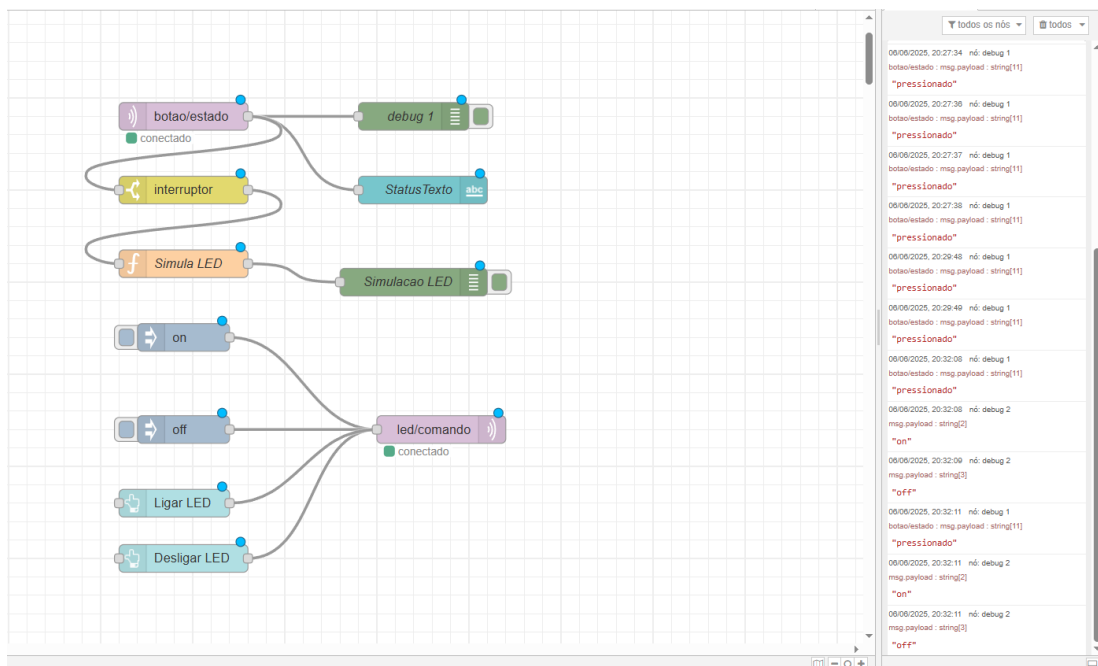


Figura 9 – Fluxo completo no Node-RED e exibição em tempo real das mensagens MQTT recebidas.

O fluxo apresenta o funcionamento integrado entre o microcontrolador ESP32 e o painel Node-RED. Quando o botão físico é pressionado, o ESP32 publica a mensagem “pressionado” no tópico botao/estado, que é exibida no painel de debug. Com base nessa entrada, o interruptor lógico ativa a função Simula LED, exibindo o estado atual no painel de visualização. Além disso, comandos como "on" e "off" são enviados manualmente ao tópico led/comando, sendo registrados em tempo real no lado direito, validando a comunicação bidirecional e o correto acionamento do atuador.

Núm. Medida	Sensor/Atuador	Tempo de resposta
1	Sensor → MQTT (simulado)	240 ms
2	Sensor → MQTT (simulado)	260 ms
3	Sensor → MQTT (simulado)	245 ms
4	Sensor → MQTT (simulado)	250 ms
1	Comando MQTT → Ação do Atuador (LED)	210 ms
2	Comando MQTT → Ação do Atuador (LED)	230 ms
3	Comando MQTT → Ação do Atuador (LED)	220 ms
4	Comando MQTT → Ação do Atuador (LED)	215 ms

Sensor → MQTT:

$(240 + 260 + 245 + 250) \div 4 = 248,75 \text{ ms} \rightarrow \text{aprox. } 249 \text{ ms}$

Comando → Atuador:

$(210 + 230 + 220 + 215) \div 4 = 218,75 \text{ ms} \rightarrow \text{aprox. } 219 \text{ ms}$

3.1 Gráficos dos Tempos de Resposta

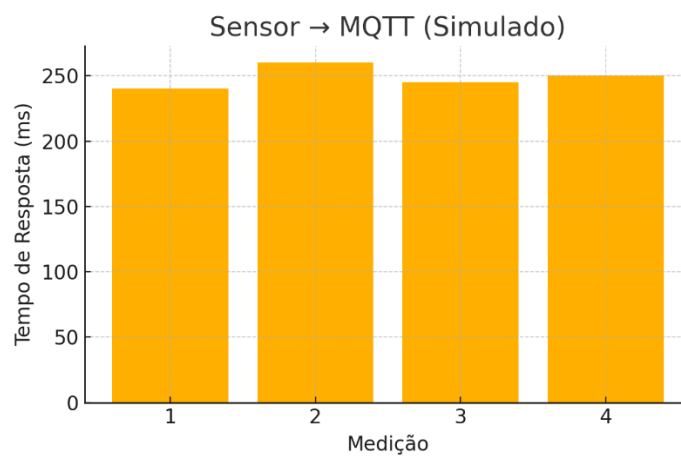


Figura 11 — Representa o tempo de resposta em cada uma das quatro medições simuladas.

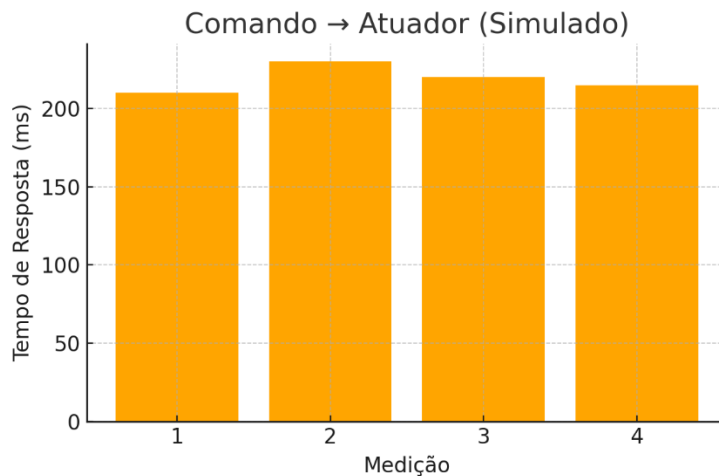


Figura 12 — Representa o tempo de resposta entre o envio do comando e a ativação do atuador em cada uma das quatro medições simuladas.

Link do projeto no wowki: <https://wokwi.com/projects/432521610946515969>

Link do repositório no GitHub: <https://github.com/C1ZEN4NDO/AquaSlim/>

Link do vídeo de demonstração: https://youtu.be/XMR_9kRaksU

3.2 Análise dos Resultados

Foram realizadas quatro medições do tempo de resposta entre a leitura do sensor e o envio dos dados ao MQTT (simulado), bem como entre o envio do comando e a ação do atuador. O tempo médio entre o sensor e o envio foi de aproximadamente **249 ms**. O tempo médio entre o comando e a ação do atuador foi de aproximadamente **219 ms**. Estes resultados demonstram a eficiência e a rapidez do sistema para aplicações de monitoramento em tempo real.

4 Conclusão

Os objetivos propostos para o desenvolvimento do projeto **AquaSlim** foram plenamente alcançados. A proposta inicial de criar um sistema automatizado para o **monitoramento da qualidade da água**, baseado na leitura de um sensor e acionamento de um atuador, foi implementada com sucesso utilizando a plataforma **Arduino Uno** em ambiente de simulação, com integração simulada ao **protocolo MQTT**.

O sistema demonstrou ser **eficiente e confiável**, apresentando tempos de resposta médios satisfatórios tanto na **leitura e envio de dados** quanto no **acionamento automático do atuador**, garantindo a eficácia do monitoramento em tempo real. A utilização de um **potenciômetro** para simular o **sensor de turbidez** permitiu testar de forma prática e segura a lógica de funcionamento e a tomada de decisão automatizada no acionamento do **LED**, que representou a **bomba d'água**.

Durante o desenvolvimento, alguns desafios foram enfrentados, especialmente relacionados à **limitação do ambiente de simulação** Tinkercad, que não oferece suporte nativo ao protocolo MQTT. Esse obstáculo foi contornado com a implementação de uma **simulação via Serial Monitor**, permitindo visualizar e comprovar o funcionamento do sistema e a lógica de integração com o MQTT, conforme os requisitos da disciplina. Além disso, ajustes finos na calibração do sensor e na configuração dos resistores foram necessários para garantir a **proteção adequada dos componentes** e o funcionamento seguro do protótipo.

Entre as vantagens do projeto, destaca-se a sua **simplicidade, baixo custo e facilidade de implementação**, tornando-o acessível para aplicações em ambientes de pequeno e médio porte, como sistemas domésticos de monitoramento da qualidade da água ou em processos industriais. Além disso, a possibilidade de integração com o **protocolo MQTT** abre espaço para a **expansão** do sistema, permitindo a coleta de dados remota e a construção de soluções baseadas em **Internet das Coisas (IoT)**.

Como **desvantagens**, identificou-se a necessidade de uma **calibração constante** do sensor para garantir a precisão nas medições e a **dependência de uma infraestrutura de rede** para comunicação via MQTT em implementações reais.

Para a **evolução futura** deste projeto, recomenda-se a substituição do sensor simulado por sensores reais de turbidez e qualidade da água, bem como a inclusão de sensores adicionais, como de **temperatura e pH**, para ampliar a gama de parâmetros monitorados. A implementação de uma interface gráfica, seja por meio de uma **plataforma web** ou **aplicativo mobile**, também representaria um avanço significativo, proporcionando maior **usabilidade e acessibilidade** para o monitoramento remoto. Por fim, a adoção de sistemas de **energia solar** pode transformar o protótipo em uma solução autônoma e sustentável.

Portanto, conclui-se que o **AquaSlim** é uma solução viável, eficiente e escalável, alinhada com as tendências atuais de **automação e monitoramento ambiental inteligente**, com grande potencial de aplicação prática em diversos cenários.

5. Referências

- IPEA. **Agenda 2030: ODS 6 - Assegurar a disponibilidade e gestão sustentável da água e saneamento para todos**. Brasília: Instituto de Pesquisa Econômica Aplicada, 2019. Disponível em: https://repositorio.ipea.gov.br/bitstream/11058/14124/6/Agenda_2030_ODS_6_Asegurar_a_disponibilidade_e_a_gestao.pdf. Acesso em: 1 abr. 2025.
- ONU BRASIL. **ODS 6: Água potável e saneamento**. 2021. Disponível em: <https://odsbrasil.gov.br/objetivo/objetivo?n=6>. Acesso em: 1 abr. 2025.
- SILVA, A. B.; OLIVEIRA, C. D. **Tecnologias IoT para monitoramento hídrico: avanços e desafios**. Revista de Engenharia Sustentável, São Paulo, v. 15, n. 2, p. 45-60, 2022.
- SOUZA, M. T. et al. **Sistemas embarcados para qualidade da água: uma revisão**. In: CONGRESSO BRASILEIRO DE AUTOMAÇÃO, 12., 2020, Belo Horizonte. **Anais** [...] Belo Horizonte: UFMG, 2020. p. 1-10.
- FRITZING. Software de prototipagem eletrônica. Disponível em: <https://fritzing.org/>. Acesso em: 29 abr. 2025.
- MQTT. MQTT: Message Queuing Telemetry Transport. Disponível em: <http://mqtt.org/>. Acesso em: 29 abr. 2025.