



## Monitoramento e Controle de Temperatura Aquática via IoT: Contribuições para a ODS 6

Silvania G. Correia<sup>1</sup>

<sup>1</sup> Faculdade de Computação e Informática  
Universidade Presbiteriana Mackenzie (UPM) – São Paulo, SP – Brazil

[silvania.correia@mackenzista.com.br](mailto:silvania.correia@mackenzista.com.br)

**Abstract.** *This paper presents the development of a water temperature monitoring and control system using Internet of Things (IoT) technologies. The system integrates a DHT22 sensor and SG90 servo motor connected to an ESP32 microcontroller, with communication via the MQTT protocol. The project was developed and tested using the Wokwi simulator, and data were visualized on a Node-RED dashboard. The temperature threshold is remotely adjustable. Response times and MQTT latencies were measured, confirming the system's viability for environmental monitoring aligned with the United Nations' Sustainable Development Goal 6 (SDG 6).*

**Resumo.** *Este trabalho apresenta o desenvolvimento de um sistema de monitoramento e controle de temperatura da água, utilizando tecnologias de Internet das Coisas (IoT). O sistema integra o sensor DHT22 e o atuador servo motor SG90, ambos conectados a um microcontrolador ESP32, com comunicação via protocolo MQTT. A montagem e os testes foram realizados no simulador Wokwi, enquanto a visualização dos dados ocorreu por meio de dashboard no Node-RED. O limite de temperatura é ajustável remotamente. Foram medidas as latências da comunicação e o tempo de resposta do atuador, demonstrando a viabilidade da solução para aplicações ambientais, em conformidade com a ODS 6 da ONU.*

### 1. Introdução

A crescente preocupação com a disponibilidade e a qualidade da água doce no mundo tem estimulado a adoção de soluções tecnológicas capazes de monitorar, em tempo real, parâmetros críticos, como a temperatura. A ODS 6 da ONU, que aborda “Água Potável e Saneamento” (NAÇÕES UNIDAS, 2015), enfatiza a importância de assegurar a disponibilidade e gestão sustentável da água para todos. Nesse cenário, sistemas de monitoramento de temperatura aquática são valiosos na detecção rápida de variações climáticas ou ambientais que possam afetar a qualidade e a disponibilidade de recursos hídricos (POFF; BRINSON; DAY, 2002).

Diversos estudos indicam que a variação de temperatura da água exerce impacto significativo sobre a saúde de ecossistemas aquáticos, influenciando parâmetros como o metabolismo e a reprodução de espécies, a disponibilidade de oxigênio dissolvido e a proliferação de algas nocivas (KARVONEN et al., 2010; MADDEN et al., 2013). Em

ambientes de piscicultura e no setor industrial, variações excessivas de temperatura podem acarretar prejuízos econômicos e riscos para a segurança hídrica (EISSA; ZAKI, 2011).

O avanço das Tecnologias de Informação e Comunicação (TICs), em particular a Internet das Coisas (IoT), possibilitou a criação de plataformas de monitoramento baseadas em microcontroladores (por exemplo, NodeMCU e ESP32), sensores de temperatura submersíveis (como o DHT22) e protocolos de comunicação em nuvem (MQTT). Essas tecnologias permitem não apenas a coleta de dados de forma contínua, mas também a adoção de ações corretivas por meio de atuadores (por exemplo, bombas de recirculação, válvulas automatizadas, sistemas de resfriamento) (MACKNICK et al., 2012).

Aliado a isso, o uso de simuladores (por exemplo, Wokwi, Tinkercad) para prototipagem acelera o processo de desenvolvimento e reduz custos iniciais, antes mesmo de se construir ou instalar um protótipo físico no local de monitoramento (MEYER et al., 1999). Dessa forma, é possível testar algoritmos de controle, calibrar sensores e integrar ferramentas de análise de dados e alerta em tempo real.

Portanto, este artigo apresenta o desenvolvimento de um sistema que realiza o monitoramento e o controle de temperatura aquática em tempo real, integrado à nuvem via protocolo MQTT, contemplando um sensor de temperatura e um atuador. O objetivo é contribuir para a preservação e gestão sustentável dos recursos hídricos, alinhando-se às metas da ODS 6 e oferecendo um modelo de aplicação que pode ser adaptado tanto para o setor público quanto para o privado.

## 2. Materiais e Métodos

Esta seção apresenta os componentes de hardware e software, a metodologia de desenvolvimento em ambiente simulado, e o funcionamento da solução de monitoramento de temperatura aquática proposta.

### 2.1. Lista de Materiais

A Tabela 1 apresenta a lista de materiais utilizados no projeto. Todos os componentes foram utilizados em ambiente simulado.

**Tabela 1 – Lista de materiais do projeto de monitoramento e controle de temperatura aquática.**

Material	Descrição	Fonte
ESP32 DevKit v1	Microcontrolador com Wi-Fi integrado	(DOIT, 2024)
Sensor DHT22	Sensor de temperatura e umidade	(AOSONG, 2020)
Servo motor SG90	Atuador para controle de ângulo	(TOWER PRO, 2019)
Ambiente Wokwi	Simulador online de hardware	(WOKWI, 2025)

Node-RED	Plataforma de desenvolvimento visual para integrar dispositivos, APIs e serviços via fluxos lógicos	(NODE-RED, 2025)
----------	---	------------------

## 2.2. Plataforma de Prototipagem

Optou-se pela utilização da placa ESP32 DevKit v1 (Figura 1), que possui conectividade Wi-Fi integrada, facilitando a comunicação via protocolo MQTT. A ESP32 DevKit v1 conta com arquitetura de 32 bits, memória RAM adequada para aplicações de IoT e múltiplos pinos de entrada/saída (GPIO). Além disso, suporta bibliotecas específicas para gerenciamento de sensores de temperatura (como o DHT22) e para implementação do MQTT (SANTOS et al., 2021).

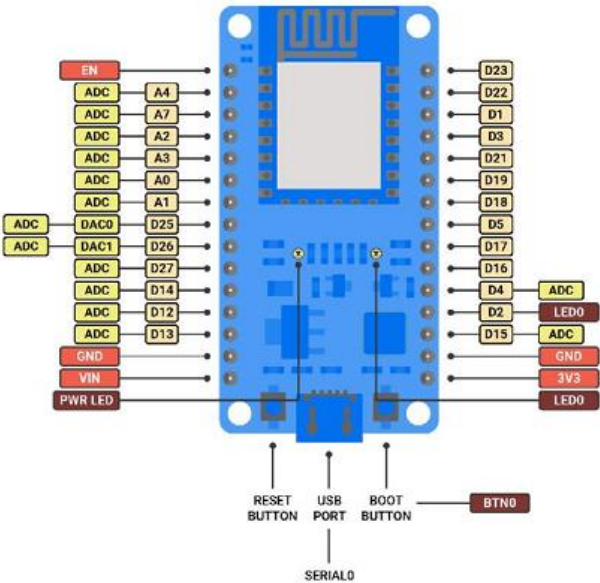


Figura 1 – Pinagem do ESP32 DevKit v1 (DOIT). Fonte: DOIT (2024, p. 1).

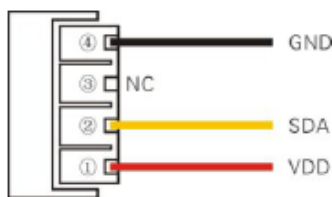
### Principais características da ESP32 DevKit v1 (HEATH, 2021):

- Processador dual-core de 32 bits;
- Conexões Wi-Fi (2,4 GHz) e Bluetooth integradas;
- Grande número de GPIOs configuráveis;
- Compatibilidade com diversas bibliotecas de IoT em C/C++.

## 2.3. Sensor de Temperatura

Para aferir a temperatura aquática, utilizou-se o sensor DHT22 (também conhecido como AM2302), representado na Figura 2. Esse sensor oferece boa precisão ( $\pm 0,5$  °C), faixa

operacional entre  $-40^{\circ}\text{C}$  a  $+80^{\circ}\text{C}$  e possibilidade de monitoramento tanto de temperatura quanto de umidade, utilizando um único pino de dados digital (AOSONG, 2020).



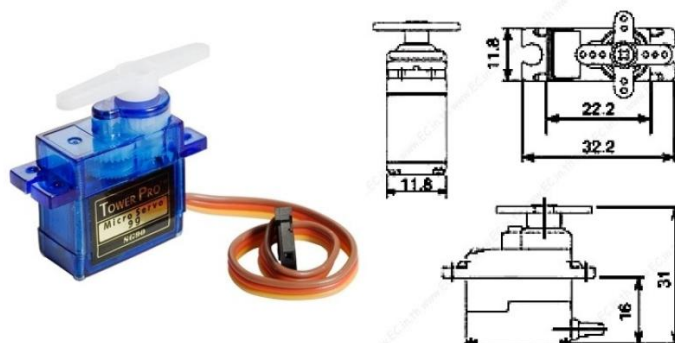
**Figura 2 - Diagrama de pinagem do sensor de temperatura e umidade DHT22 (AM2302). Fonte: AOSONG (2020, p. 3).**

#### **Principais características do DHT22:**

- Medição de temperatura e umidade;
- Tensão de operação de 3V a 5.5V;
- Comunicação via protocolo de um fio (single-wire).

#### **2.4. Atuador**

Utilizou-se o servo motor SG90 (Figura 3) como atuador, acionado para simular a abertura ou fechamento de uma válvula de resfriamento. O controle é realizado via sinal PWM, permitindo posicionar o eixo em diferentes ângulos conforme a necessidade.



**Figura 3 - Servomotor SG90 e dimensões. Fonte: TOWER PRO (2019).**

#### **Principais características do servo SG90:**

- Tensão de operação na faixa de 4,8 a 6,0 V;
- Ângulo de rotação de aproximadamente 180 graus;
- Controle via sinal PWM (frequência típica de 50 Hz).

#### **2.5. Ambiente Simulado e Ferramentas de Desenvolvimento**

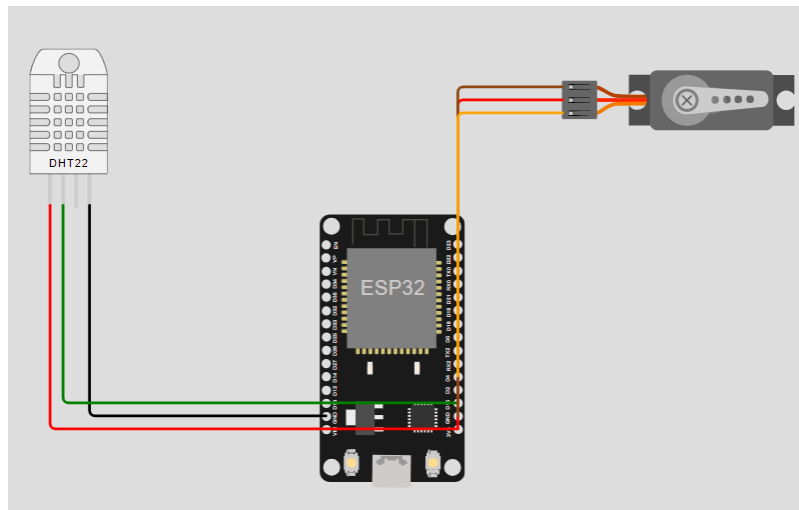
O projeto foi desenvolvido na plataforma Wokwi (WOKWI, 2025), que permite a montagem virtual do circuito, a programação do ESP32 em linguagem C/C++, e a simulação de leituras em tempo real do sensor DHT22 e do comportamento do servo motor. A ferramenta possibilita também a integração com brokers MQTT para publicação e subscrição de dados.

Além do Wokwi, utilizou-se o Node-RED para construir um painel de monitoramento. O fluxo MQTT é recebido, tratado e visualizado através de widgets como gauge, texto e

LED virtual. O projeto também permite reconfiguração do limite de temperatura via campo de entrada conectado ao tópico MQTT correspondente.

## 2.6. Diagrama de Montagem

A Figura 4 apresenta o circuito de montagem no Wokwi, evidenciando a conexão do DHT22 ao pino D15, o servo ao pino D4 e os terminais de alimentação e terra compartilhados com a ESP32.

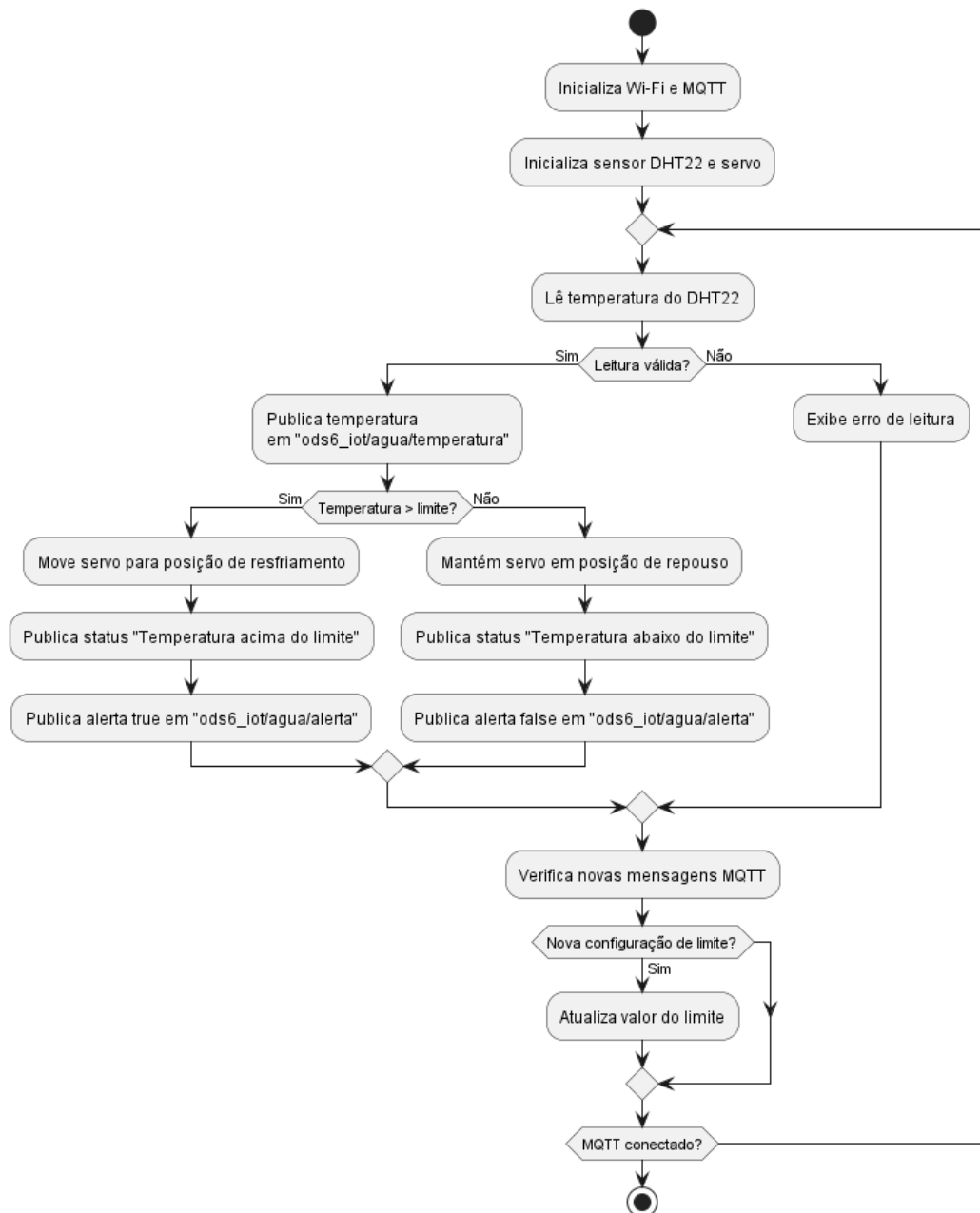


**Figura 4 - Montagem do circuito no Wokwi. Fonte: Elaborado pela autora (2025).**

### 2.6.1. Fluxograma de Funcionamento

A Figura 5 apresenta o fluxograma do sistema, mostrando a leitura do sensor, publicação MQTT, verificação do limite e atuação do servo.

**Fluxograma do Sistema IoT - Controle de Temperatura da Água**



**Figura 5 – Fluxograma de funcionamento. Fonte: Elaborado pela autora (2025).**

## 2.7. Funcionamento do Sistema

O sistema realiza o monitoramento contínuo da temperatura aquática com o sensor DHT22. A cada 3 segundos, a ESP32 realiza uma nova leitura e publica os dados no tópico ``ods6_iot/agua/temperatura``. Se a temperatura ultrapassar o limite pré-definido (ajustável dinamicamente), o servo motor SG90 é acionado, simulando abertura de uma válvula de resfriamento. Um alerta é publicado em ``ods6_iot/agua/status``, e um LED virtual no Node-RED é aceso via ``ods6_iot/agua/alerta``. Quando a temperatura retorna a níveis seguros, o servo volta à posição de repouso e o LED é desligado. O valor do limite de temperatura é ajustável por publicação no tópico ``ods6_iot/agua/config/limite``.

## 2.8. Comunicação MQTT

A comunicação dos dados entre o microcontrolador e a interface de monitoramento foi implementada utilizando o protocolo MQTT (Message Queuing Telemetry Transport - <http://mqtt.org>), amplamente adotado em aplicações de Internet das Coisas devido à sua leveza e eficiência em conexões TCP/IP (IBM, 2021).

O broker MQTT utilizado no projeto foi o broker público HiveMQ, disponível em [broker.hivemq.com](http://broker.hivemq.com), na porta 1883, com comunicação via protocolo TCP/IP sem criptografia.

A ESP32 DevKit v1 realiza tanto publicações quanto assinaturas de tópicos MQTT. As publicações são feitas por meio da biblioteca PubSubClient.h, e os dados são organizados nos seguintes tópicos:

- `ods6_iot/agua/temperatura` – Publicação contínua dos valores de temperatura lidos pelo sensor DHT22.
- `ods6_iot/agua/status` – Mensagem textual de status sobre a situação do sistema (ex: “Atuador acionado”).
- `ods6_iot/agua/alerta` – Valor booleano (“true” ou “false”) usado para controle visual no LED do Node-RED.

Além disso, a ESP32 realiza assinatura do tópico:

- `ods6_iot/agua/config/limite` – Permite que o usuário envie remotamente um novo valor de limite de temperatura para acionar o atuador, modificando o comportamento do sistema sem reprogramação manual.

As mensagens enviadas e recebidas podem ser visualizadas e processadas em tempo real por meio da plataforma Node-RED, que foi integrada ao projeto para criação de um dashboard interativo com elementos como gauge de temperatura, LED virtual e campo de entrada para nova configuração.

## 2.9. Procedimentos Experimentais

1. Montagem do circuito no Wokwi com ESP32, DHT22 (D15) e servo (D4)
2. Implementação do código com bibliotecas DHTesp.h, PubSubClient.h e ESP32Servo.h;
3. Configuração do broker público [broker.hivemq.com](http://broker.hivemq.com);
4. Definição do limite inicial de 28 °C para acionamento do servo;
5. Testes com simulação de temperatura crescente e decrescente;
6. Integração e visualização no painel Node-RED.

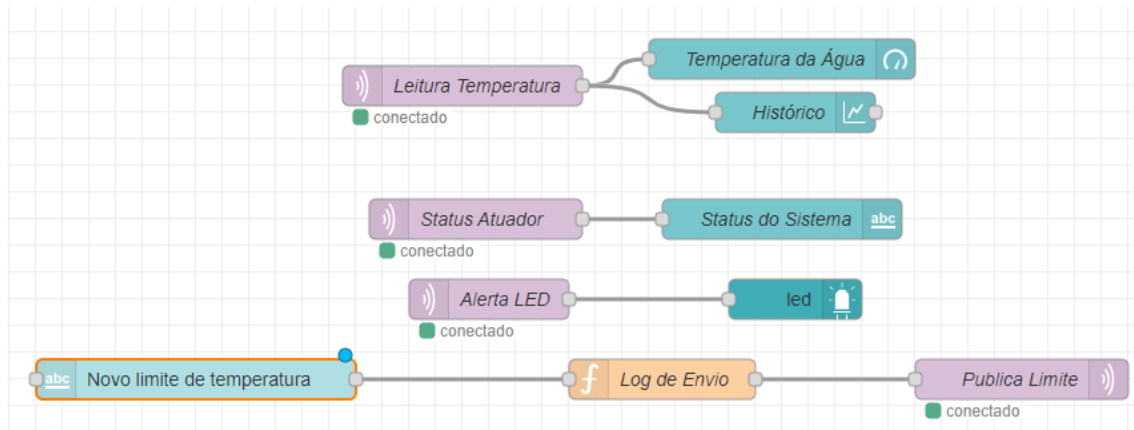
## 3. Resultados

Esta seção apresenta registros visuais da execução do projeto, com imagens capturadas durante o funcionamento real da simulação no Wokwi e da interface Node-RED. As imagens destacam a comunicação MQTT, a leitura dos sensores, a resposta do atuador e a atualização visual do painel.

### 3.1. Fluxo de Mensagens no Node-RED

A Figura 6 apresenta o fluxo de mensagens construído na plataforma Node-RED. Os blocos em lilás representam nós de entrada e saída MQTT que se comunicam com o

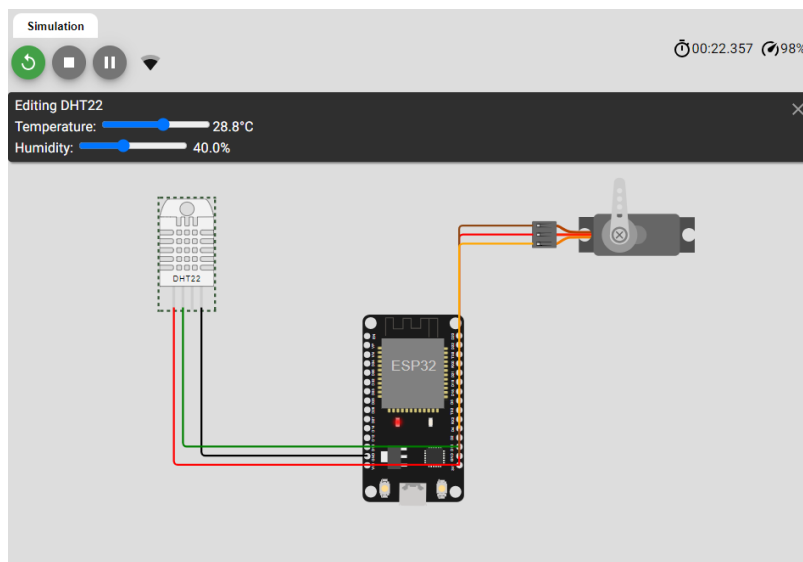
broker público HiveMQ. A temperatura medida é exibida em tempo real em um gauge e um gráfico histórico. O status do sistema e o alerta visual são atualizados com base nas condições definidas no microcontrolador. O fluxo também inclui um campo para reconfiguração dinâmica do limite de temperatura, com log de envio para medição de tempo de resposta.



**Figura 6 – Fluxo de execução no Node-RED. Fonte: Elaborado pela autora (2025).**

### 3.2. Funcionamento do Circuito Simulado no Wokwi

A Figura 7 mostra o circuito com ESP32, sensor DHT22 e servo motor SG90 em funcionamento. Durante a simulação, a temperatura ultrapassou o limite definido, acionando o servo para simular a abertura de uma válvula de resfriamento. O valor de temperatura é publicado via MQTT.



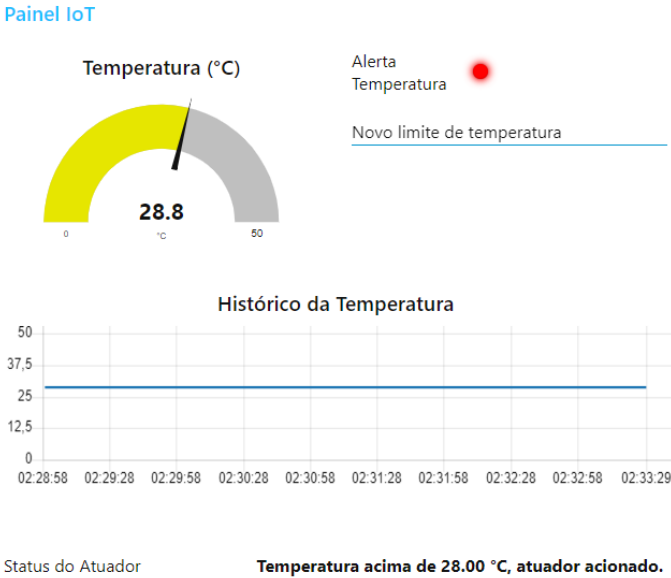
**Figura 7 – Circuito no Wokwi em execução com temperatura elevada. Fonte: Elaborado pela autora (2025).**

### 3.3. Visualização em Tempo Real no Node-RED

No painel desenvolvido no Node-RED (Figura 8), a temperatura acima do limite aciona visualmente o LED de alerta (vermelho), atualiza o valor no gauge e exibe a mensagem



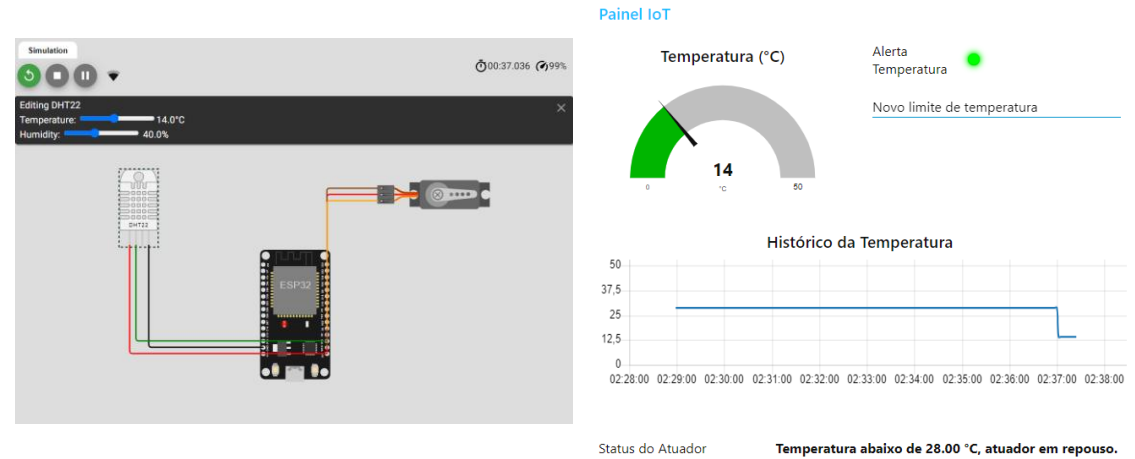
“Temperatura acima de LIMITE”, sendo o LIMITE a constante contendo o limite de temperatura definido no momento. Esta imagem comprova a comunicação MQTT e a resposta visual imediata à leitura do sensor.



**Figura 8 – Dashboard Node-RED com alerta ativo. Fonte: Elaborado pela autora (2025).**

**3.4. Retorno ao Estado Normal**

Após a redução da temperatura simulada no Wokwi, o servo retorna à posição de repouso e o painel Node-RED exibe “Temperatura abaixo de LIMITE”, com o LED de alerta apagado (Figura 9). Isso demonstra o comportamento dinâmico e o controle automático do sistema.

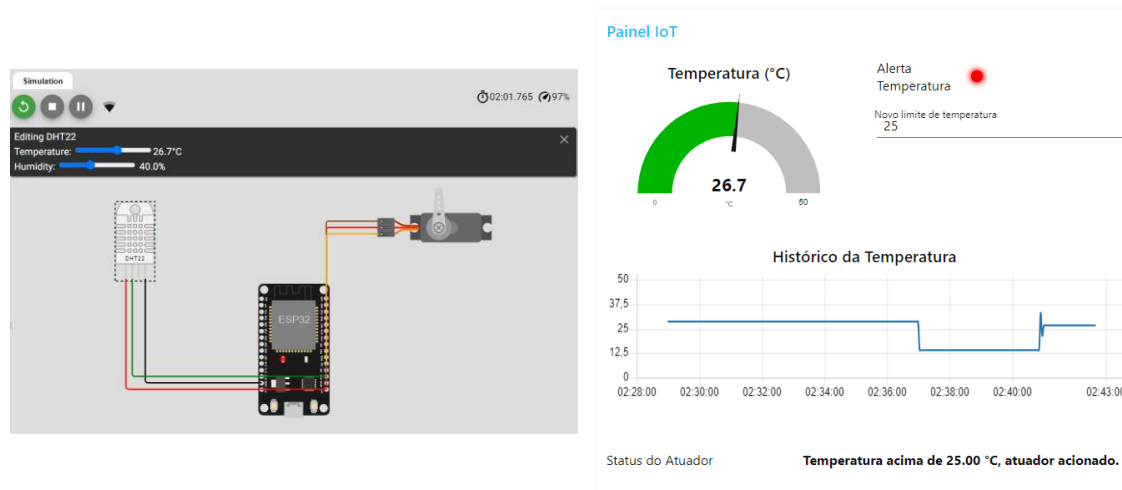


**Figura 9 – Retorno do sistema ao estado normal no circuito em execução no Wokwi e no dashboard Node-RED. Fonte: Elaborado pela autora (2025).**

**3.5. Reconfiguração do Limite de Temperatura via MQTT**

Na interface do Node-RED, o usuário inseriu um novo valor de limite no campo de texto, que foi publicado no tópico `ods6_iot/agua/config/limite`. A ESP32 recebeu este comando

e passou a adotar o novo valor para decisão de acionamento do servo. Na simulação, o novo limite foi definido como 25 °C, e o sistema reagiu de acordo com a nova configuração (Figura 10).



**Figura 10 – Alteração remota do limite de temperatura pelo usuário.**

### 3.6. Medições de Tempo de Resposta

Para validar o desempenho do sistema, foram realizadas medições de tempo de resposta em dois momentos distintos do fluxo:

1. O tempo entre a leitura da temperatura pelo sensor DHT22 e a publicação do dado via protocolo MQTT.
2. O tempo entre o envio remoto de um novo valor de limite de temperatura (via Node-RED) e a resposta do atuador (servo motor).

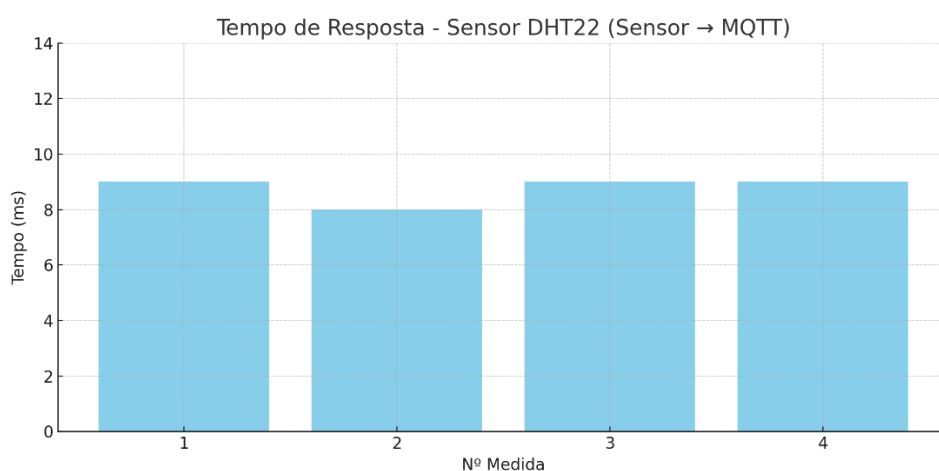
#### 3.6.1. Medição do Tempo de Resposta Sensor → MQTT

As leituras foram obtidas a partir dos `Serial.println(millis())` programados no sketch da ESP32. Cada leitura indica o tempo exato (em milissegundos desde o boot) em que a leitura do sensor ocorreu e quando a publicação MQTT foi realizada. A Tabela 2 apresenta os dados coletados.

**Tabela 2 - Tempo de resposta entre leitura e publicação MQTT**

Nº Medida	Sensor/Atuador	Tempo de resposta
1	DHT22	$8766 - 8757 = 9 \text{ ms}$
2	DHT22	$11782 - 11774 = 8 \text{ ms}$
3	DHT22	$14797 - 14788 = 9 \text{ ms}$
4	DHT22	$17811 - 17802 = 9 \text{ ms}$
Tempo médio (sensor → MQTT): 8,75 ms		

A Figura 11 apresenta o gráfico contendo os tempos de resposta do sensor.



**Figura 11 - Gráfico do tempo de resposta sensor → MQTT. Fonte: Elaborado pela autora (2025).**

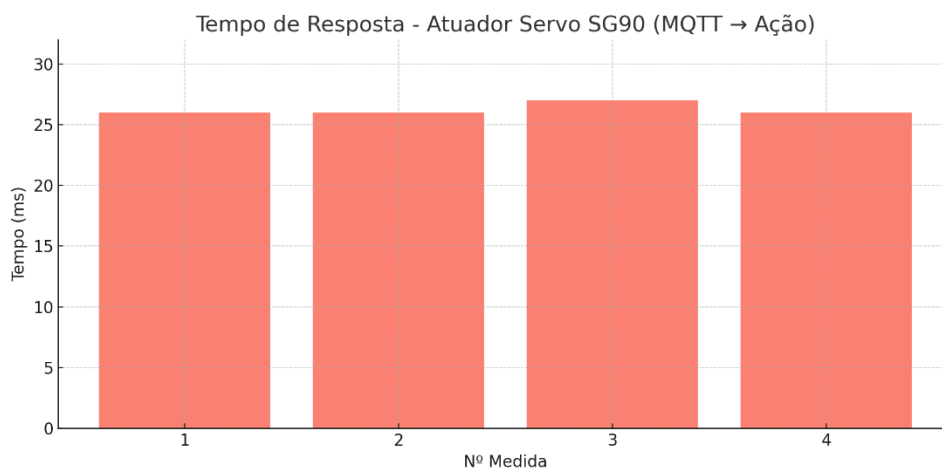
### 3.6.2. Medição do Tempo de Resposta MQTT → Atuador

Neste caso, foram anotados os horários de publicação do novo limite no Node-RED e comparados com o tempo da reação do servo motor registrada no console da Wokwi. A Tabela 3 mostra os dados obtidos.

**Tabela 3 - Tempo de resposta entre comando remoto e ação do atuador**

Nº Medida	Sensor/Atuador	Tempo de resposta
1	SG90	$79899 - 79873 = 26 \text{ ms}$
2	SG90	$90167 - 90141 = 26 \text{ ms}$
3	SG90	$99390 - 99363 = 27 \text{ ms}$
4	SG90	$109658 - 109632 = 26 \text{ ms}$
Tempo médio (MQTT → atuador): 26,25 ms		

A Figura 12 apresenta o gráfico com os valores obtidos para o tempo de resposta do atuador.



**Figura 12 - Gráfico do tempo de resposta MQTT → Atuador. Fonte: Elaborado pela autora (2025).**

### 3.6.3. Análise do Tempo de Resposta

Os resultados obtidos indicam excelente desempenho para o propósito do projeto. A comunicação MQTT foi eficiente, com latência inferior a 10 ms para a publicação dos dados de temperatura e cerca de 26 ms para acionamento do atuador, garantindo resposta quase em tempo real. Segundo Thangavel et al. (2014), aplicações de IoT toleram tempos de resposta de até 100 ms em cenários típicos de monitoramento e automação residencial, o que confirma a adequação do desempenho observado neste experimento.

### 3.7. Acesso ao Código-Fonte

O repositório a seguir contém o código-fonte, imagens do circuito e do painel, arquivos do Node-RED, descrição dos tópicos MQTT e instruções de execução local.

Acesso em: [https://github.com/silvianiacorreia/ods6\\_iot\\_projeto\\_controle\\_temperatura](https://github.com/silvianiacorreia/ods6_iot_projeto_controle_temperatura) - Atualizado em 24/05/2025

## 4. Conclusões

O projeto apresentado demonstrou a viabilidade da aplicação de tecnologias de Internet das Coisas (IoT) para o monitoramento e controle de temperatura aquática em tempo real, utilizando sensores e atuadores conectados via protocolo MQTT, o qual é amplamente utilizado por sua leveza e confiabilidade em aplicações IoT (IBM, 2021; SANTOS et al., 2021).

### 4.1. Alcance de Objetivos Propostos

O sistema foi integralmente desenvolvido em ambiente simulado, realizando com sucesso a leitura contínua da temperatura da água por meio do sensor DHT22, a comunicação dos dados via MQTT para um dashboard Node-RED e o acionamento automático de um servo motor SG90 ao ultrapassar um limite configurável. O projeto também atendeu ao requisito de possibilitar o controle remoto do limite de temperatura, mostrando-se funcional e adaptável às metas da ODS 6 (NAÇÕES UNIDAS, 2015).

### 4.2. Principais Desafios

O principal desafio foi garantir a sincronização dos tempos de leitura e reação do sistema, especialmente no ambiente simulado. Para contornar essa dificuldade, foi utilizada a função `millis()` no código do ESP32 para registrar os tempos internos com precisão, e o Node-RED foi configurado com funções de log para capturar os momentos exatos da publicação de comandos. A adoção dessas práticas está em conformidade com abordagens sugeridas por Heath (2021) para análise de desempenho em sistemas embarcados.

### 4.3. Vantagens e Desvantagens do Projeto

Entre as vantagens destacam-se:

- A baixa latência nas comunicações MQTT (média inferior a 10 ms para sensores e ~26 ms para atuadores), o que atende aos requisitos de aplicações responsivas em IoT (SANTOS et al., 2021);
- A facilidade de reconfiguração dinâmica sem necessidade de reprogramação;
- A visualização amigável e interativa via Node-RED, ferramenta bastante recomendada para integração de sensores com interfaces visuais (NODE-RED, 2025).

As principais limitações do projeto estão relacionadas ao uso de ambiente simulado (WOKWI, 2025), o que restringe a análise de fatores físicos do mundo real, como ruídos elétricos, falhas de hardware e variações ambientais inesperadas.

### 4.4. Melhorias Futuras

A principal evolução possível seria a migração para testes em ambiente físico, com validação em cenários reais de monitoramento (lagos, reservatórios, tanques de piscicultura). Também seria recomendada a integração com sensores adicionais (como pH ou turbidez), armazenamento em banco de dados online e envio de alertas por e-mail ou mensagem, ampliando o impacto da solução para contextos ambientais mais complexos (KARVONEN et al., 2010; MEYER et al., 1999).

## 5. Referências

- AOSONG (Guangzhou) Electronics Co., Ltd. *Temperature and humidity module AM2302 Product Manual*. [S.l.], 2020. Disponível em: <https://www.makerhero.com/img/files/download/DHT22-AM2302-Datasheet.pdf>. Acesso em: 27 abr. 2025.
- DOIT. *ESP32 Module: DOIT ESP32 DevKit v1* (datasheet). [S.l.], 2024. Disponível em: <https://roboeq.ir/files/id/4034/name/ESP32%20MODULE.pdf>. Acesso em: 27 abr. 2025.
- EISSA, A. E.; ZAKI, M. M. *The impact of global climatic changes on the aquatic environment*. Procedia Environmental Sciences, v. 4, p. 251–259, 2011.
- HEATH, S. *Embedded systems design: the art of hardware, software, and systems engineering*. 3. ed. Oxford: Newnes, 2021.
- IBM. *Por que o MQTT é bom para a Internet das Coisas?* IBM DeveloperWorks, 2021. Disponível em: <https://www.ibm.com/developerworks/br/library/iot-mqtt-why-good-for-iot/index.html>. Acesso em: 25 mar. 2025.

- KARVONEN, A.; RINTAMÄKI, P.; JOKELA, J.; VALTONEN, E. T. Increasing water temperature and disease risks in aquatic systems: climate change increases the risk of some, but not all, diseases. *International Journal for Parasitology*, v. 40, n. 12, p. 1483–1488, 2010.
- MACKNICK, J. et al. Operational water consumption and withdrawal factors for electricity generating technologies: a review of existing literature. *Environmental Research Letters*, v. 7, n. 4, p. 045802, 2012.
- MADDEN, N.; LEWIS, A.; DAVIS, M. Thermal effluent from the power sector: an analysis of once-through cooling system impacts on surface water temperature. *Environmental Research Letters*, v. 8, n. 3, p. 035006, 2013.
- MAXIM INTEGRATED. *DS18B20: Programmable resolution 1-Wire digital thermometer (datasheet)*. Rev. 6, 7/19. Califórnia, 2019. Disponível em: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>. Acesso em: 24 mar. 2025.
- MEYER, J. L.; SALE, M. J.; MULHOLLAND, P. J.; POFF, N. L. Impacts of climate change on aquatic ecosystem functioning and health. *Journal of the American Water Resources Association*, v. 35, n. 6, p. 1373–1386, 1999.
- NAÇÕES UNIDAS. *Transformando nosso mundo: a Agenda 2030 para o Desenvolvimento Sustentável*. [S.l.], 2015. Disponível em: <https://brasil.un.org/pt-br/sdgs>. Acesso em: 06 mar. 2025.
- NODE-RED. Node-RED: Low-code programming for event-driven applications. [S.l.], 2025. Disponível em: <https://nodered.org>. Acesso em: 20 mai. 2025.
- POFF, N. L.; BRINSON, M. M.; DAY, J. W. *Aquatic ecosystems & global climate change: potential impacts on inland freshwater and coastal wetland ecosystems in the United States*. Arlington, VA: Pew Center on Global Climate Change, 2002.
- SANTOS, B. et al. *Internet das Coisas: da teoria à prática*. Belo Horizonte: DCC/UFGM, 2021. Disponível em: <https://homepages.dcc.ufmg.br/~mmvieira/cc/papers/internet-das-coisas.pdf>. Acesso em: 25 mar. 2025.
- THANGAVEL, D. et al. *Performance evaluation of MQTT and CoAP via a common middleware*. In: IEEE. IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014, Singapore. Anais [...]. Piscataway: IEEE, 2014. p. 1–6. DOI: 10.1109/ISSNIP.2014.6827678.
- TOWER PRO. *SG90 9 g Micro Servo (datasheet)*. [S.l.]: [s.n.], 2019. Disponível em: <https://www.alldatasheet.com/datasheet-pdf/download/1572383/ETC/SG90.html>. Acesso em: 24 mar. 2025.
- WOKWI. Wokwi - Simulador Online de Arduino, ESP32 e outros dispositivos de IoT. [S.l.], 2025. Disponível em: <https://wokwi.com/>. Acesso em: 26 abr. 2025.