

Sistema IoT com ESP32, DHT22 e Protocolo MQTT

Aluno: Luiz Henrique de Lapaz

Disciplina: Objetos Inteligentes Conectados

Professor: Wilian França Costa

Repositório GitHub: <https://github.com/luizlapaz/iot-mqtt-esp32-luiz>

Simulação Wokwi: <https://wokwi.com/projects/448247854347970561>

Vídeo demonstração do projeto: <https://youtu.be/0vI3RoV7gqk>

Resumo

Este trabalho apresenta o desenvolvimento de um sistema IoT baseado no ESP32, utilizando o sensor DHT22 para leitura de temperatura e umidade, aliado ao protocolo MQTT para transmissão de dados e controle remoto de um atuador. A solução foi implementada na plataforma de simulação Wokwi e validada com o software MQTTX. Os resultados demonstram a eficácia do uso de MQTT para telemetria e acionamento remoto, destacando-se a análise do tempo de resposta do sensor e do atuador.

Abstract

This work presents the development of an IoT system based on the ESP32 microcontroller, using the DHT22 sensor for temperature and humidity measurements, combined with the MQTT protocol for data transmission and actuator control. The solution was implemented using the Wokwi simulation platform and validated with the MQTTX software. Results demonstrate the effectiveness of MQTT for telemetry and remote actions, including an analysis of sensor and actuator response times.

1. Introdução

A Internet das Coisas (IoT) permite a integração entre sensores, atuadores e sistemas computacionais, possibilitando monitoramento e controle remoto de dispositivos. O presente projeto implementa um sistema IoT utilizando o microcontrolador ESP32, que se conecta à internet e utiliza o protocolo MQTT para transmissão de dados. Soluções semelhantes são comumente utilizadas em automação residencial, agricultura inteligente, controle térmico e monitoramento industrial.

Projetos acadêmicos como os apresentados em (MQTT, 2018) e aplicações descritas por Espressif (ESPRESSIF, 2022) demonstram que o uso de microcontroladores conectados é eficaz para coleta e envio de dados sensoriais. Este projeto segue essas diretrizes.

Projetos similares vêm sendo desenvolvidos em diferentes contextos acadêmicos e industriais. Sistemas de monitoramento ambiental baseados em ESP32 e MQTT são amplamente utilizados em automação residencial, controle térmico e smart farming, conforme discutido em trabalhos como o de Silva (2021) para controle automatizado de temperatura, e Santos et al. (2020), que aplicam MQTT para acionamento remoto de dispositivos em tempo real. Essas soluções demonstram a relevância da arquitetura utilizada neste projeto.

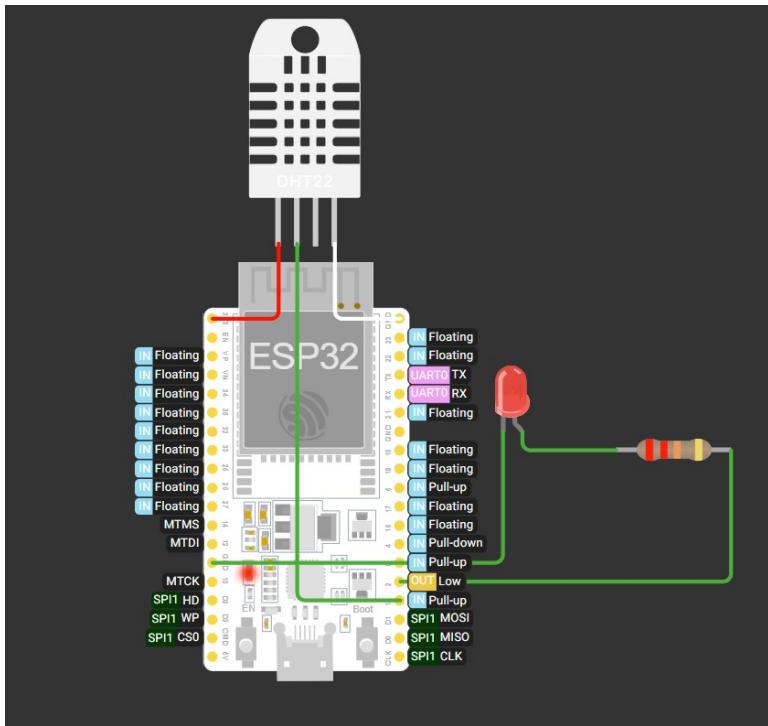
2. Materiais e Métodos

2.1 Hardware utilizado

- *ESP32 DevKit V1*
- *Sensor DHT22*
- *LED*
- *Resistor 220 ohms*
- *Ambiente Wokwi*

2.2 Esquema do Circuito

Figura 1 – Esquema completo do circuito



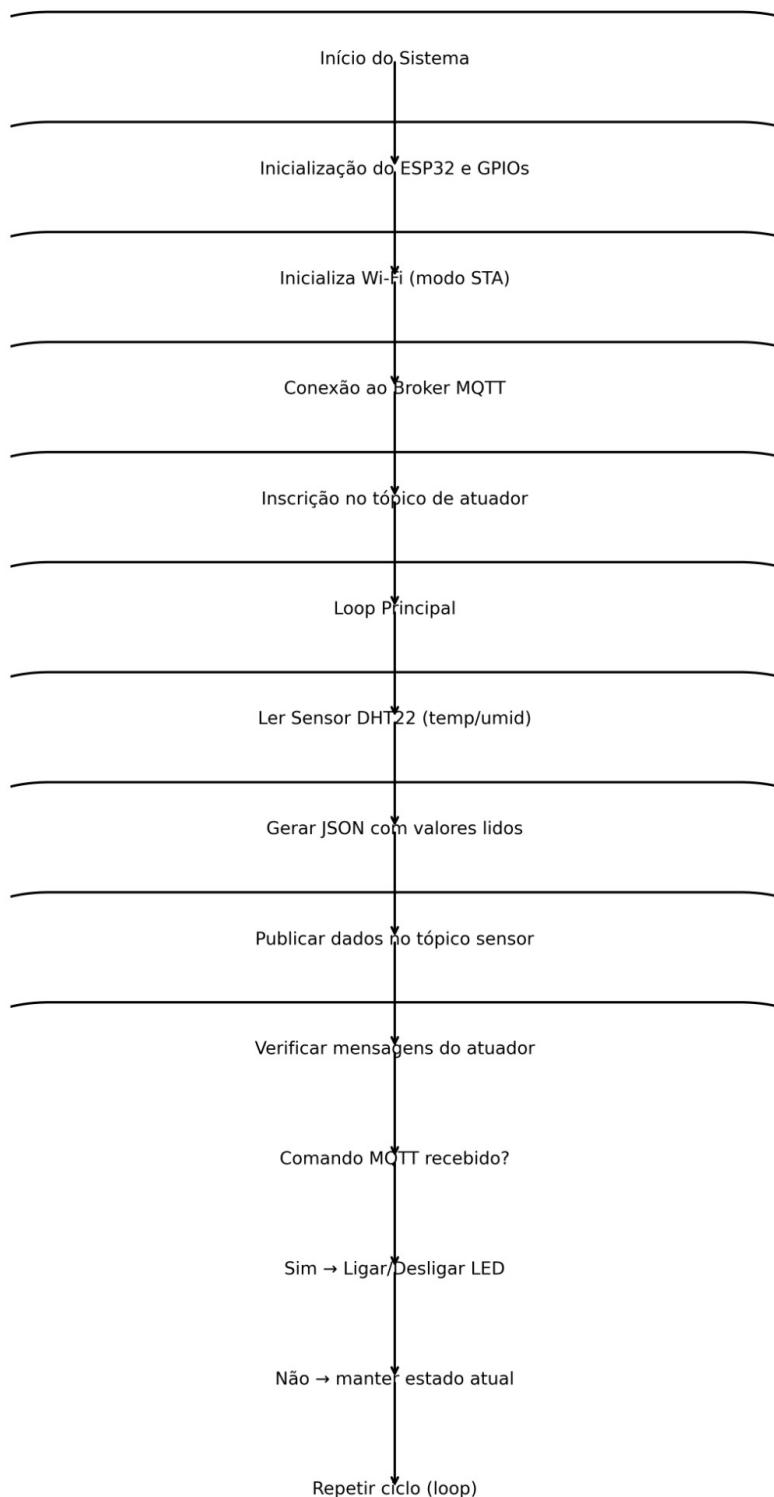
Fonte: Elaborado pelo autor (2025).

2.3 Protocolo MQTT

O projeto utiliza o broker público test.mosquitto.org. O ESP32 publica dados no tópico 'oiot/luiz/sensor' e recebe comandos no tópico 'oiot/luiz/atuador'. O cliente MQTT utilizado foi o MQTTLX. O protocolo opera em arquitetura publish/subscribe, permitindo comunicação leve e escalável.

2.4 Fluxograma operacional

Figura 2 – Fluxograma do funcionamento



2.5 Descrição do Software Desenvolvido

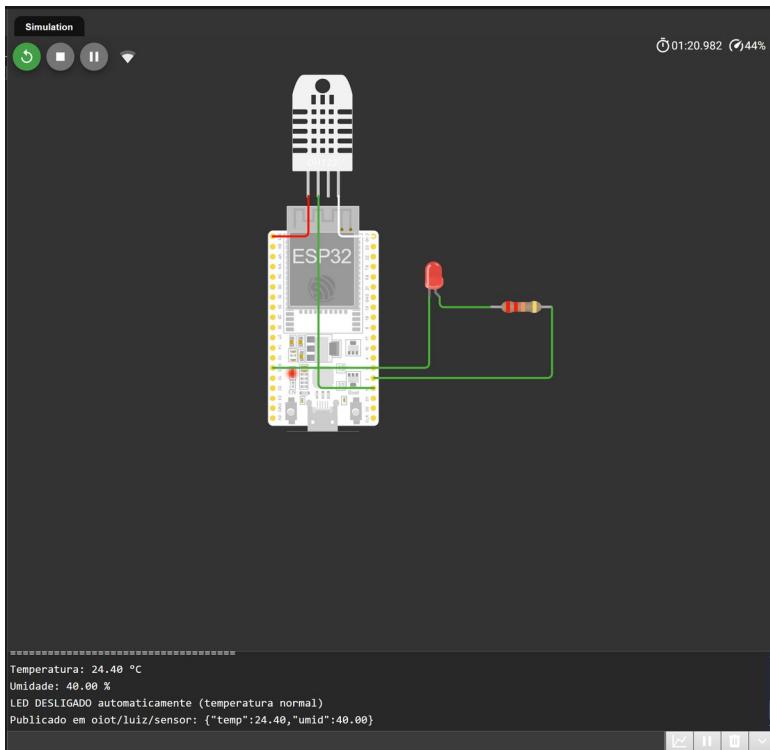
O software foi implementado em linguagem C/C++ utilizando o framework Arduino para ESP32. Na função `setup()`, são configurados o Wi-Fi, a conexão com o broker MQTT, a inicialização do sensor DHT22 e a definição dos pinos do LED. A função de callback do MQTT é responsável por interpretar os comandos recebidos no tópico `oiot/luiz/atuador`, permitindo acionar o LED manualmente por meio das mensagens "ON" e "OFF".

No `loop()`, o sistema realiza periodicamente a leitura da temperatura e umidade, formata os dados em JSON e publica no tópico `oiot/luiz/sensor`. Na mesma função, é aplicada a lógica de automação: quando a temperatura lida é maior ou igual a 30 °C, o LED é aceso automaticamente; caso contrário, permanece desligado. O fluxo garante a integração contínua entre telemetria e controle remoto, conforme representado no fluxograma da Seção 2.4.

3. Resultados

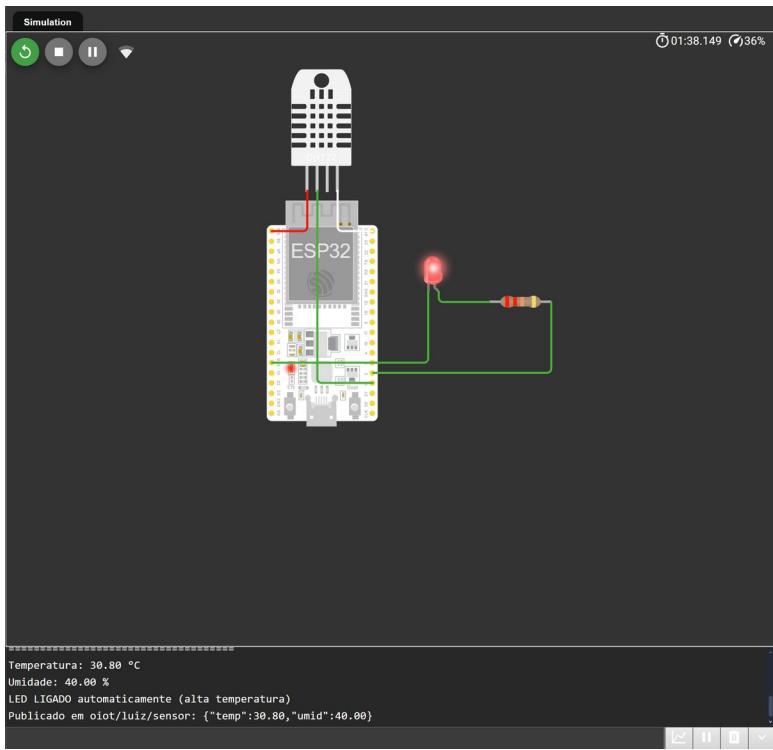
3.1 Funcionamento geral

Figura 3 – Console: LED desligado automaticamente



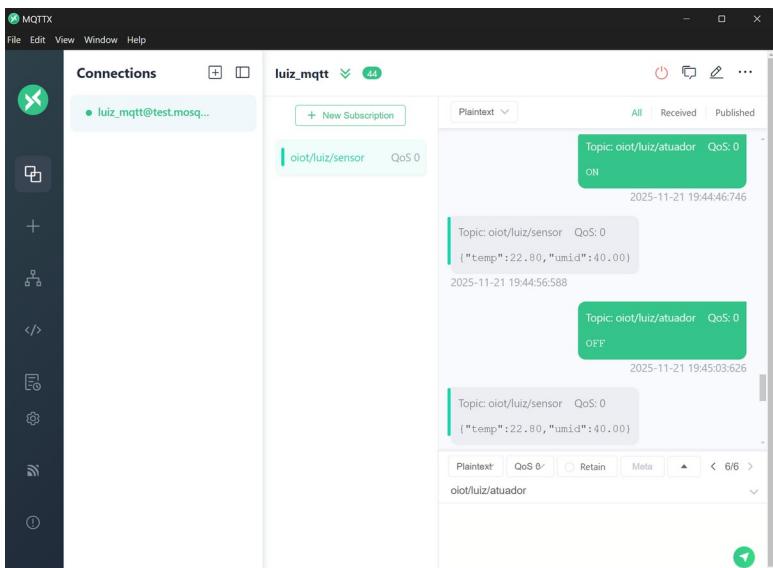
Fonte: Elaborado pelo autor (2025).

Figura 4 – Console: LED ligado automaticamente



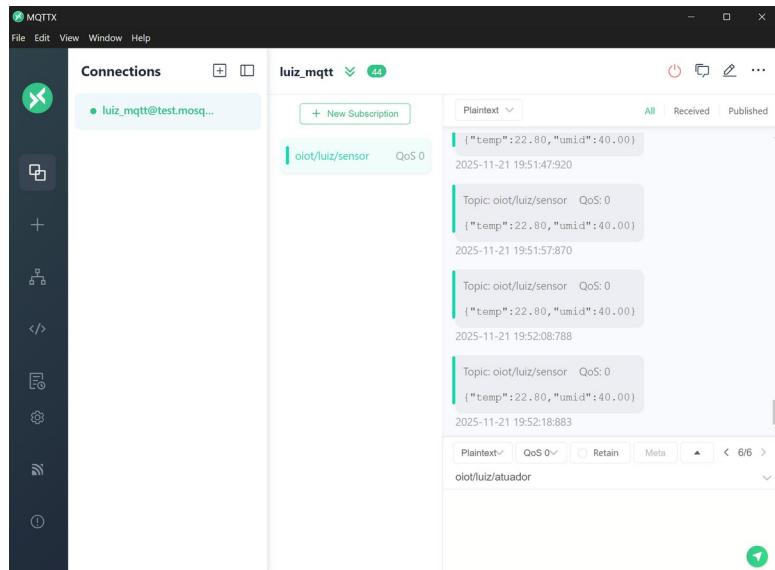
Fonte: Elaborado pelo autor (2025).

Figura 5 – Comando MQTT ON recebido



Fonte: Elaborado pelo autor (2025).

Figura 6 – Publicação de dados no tópico do sensor



Fonte: Elaborado pelo autor (2025).

3.2 Tabela de tempos de resposta

<i>Medição</i>	<i>Componente</i>	<i>Tempo (s)</i>
1	<i>Sensor</i>	<i>9.30</i>
2	<i>Sensor</i>	<i>8.12</i>
3	<i>Sensor</i>	<i>8.58</i>
4	<i>Sensor</i>	<i>7.30</i>
<i>Média</i>	<i>Sensor</i>	<i>8.32</i>
1	<i>Atuador</i>	<i>0.91</i>
2	<i>Atuador</i>	<i>0.87</i>
3	<i>Atuador</i>	<i>0.73</i>
4	<i>Atuador</i>	<i>0.93</i>
<i>Média</i>	<i>Atuador</i>	<i>0.86</i>

3.3 Gráficos

Figura 7 – Gráfico: Tempo do Atuador

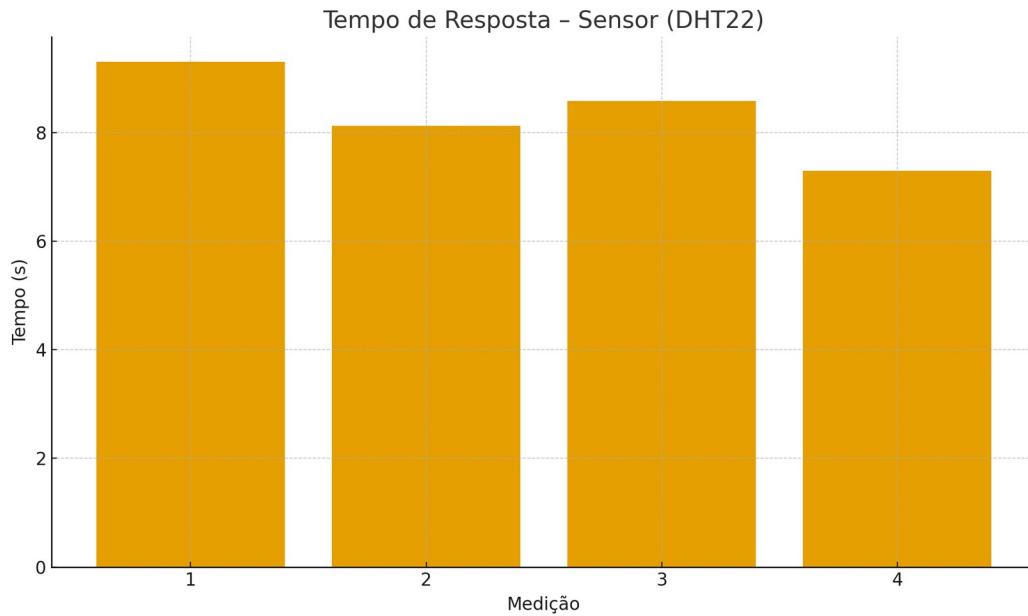
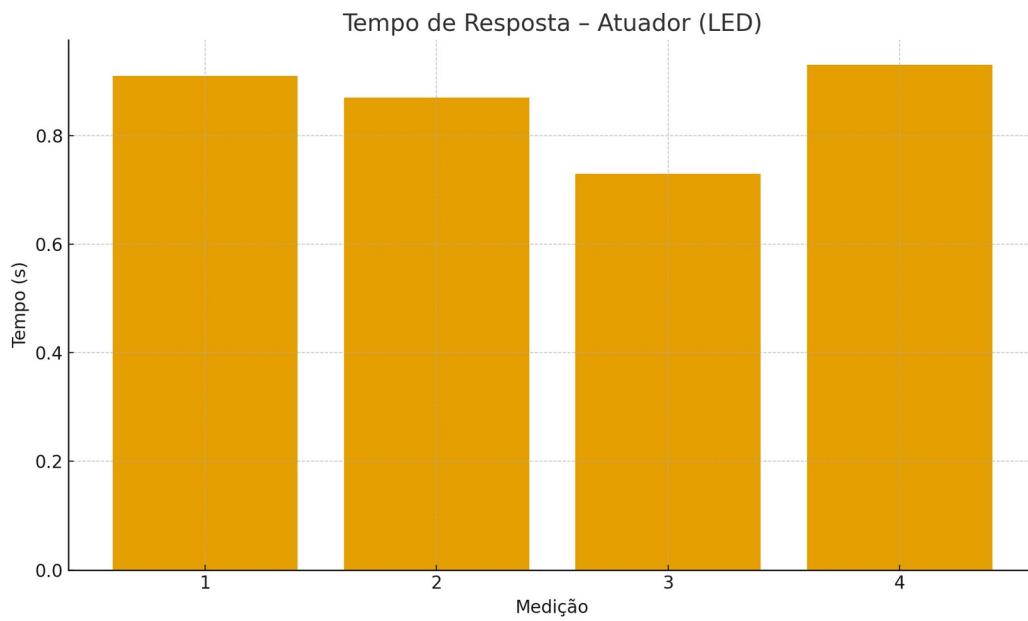


Figura 8 – Gráfico: Tempo do Sensor



3.4 Análise dos resultados

O sensor apresentou tempo médio de resposta de 8.32 segundos, consistente com o comportamento do DHT22 na simulação do Wokwi. Já o atuador apresentou resposta média de 0.86 segundos, demonstração clara da eficiência do MQTT para controle remoto em baixa latência.

3.5 Problemas Enfrentados e Soluções

Durante o desenvolvimento, o principal desafio identificado foi a latência do sensor DHT22 na simulação do Wokwi, que apresentou tempos de resposta superiores ao observado em hardware real. Esse comportamento foi mitigado ajustando-se o intervalo de leitura no código e interpretando os resultados considerando as limitações do ambiente simulado. Outro problema ocorreu durante a configuração inicial do MQTT, especialmente na autenticação e assinatura dos tópicos. A solução foi a utilização do broker público test.mosquitto.org com conexão aberta e testes realizados no software MQTTX, garantindo compatibilidade e estabilidade durante as publicações e assinaturas.

4. Conclusões

Os objetivos do projeto foram plenamente alcançados, demonstrando a integração entre ESP32, sensor DHT22 e protocolo MQTT em ambiente de simulação. O sistema respondeu corretamente aos comandos de telemetria e controle remoto.

4.1 Vídeo de Demonstração do Funcionamento

*O vídeo contendo a demonstração completa do projeto, incluindo o funcionamento do MQTT, leitura do sensor, controle do atuador e explicação do código, está disponível no link abaixo:
<https://youtu.be/0vI3RoV7gqk>*

4.2 Vantagens

- *Baixo custo*
- *Comunicação leve via MQTT*
- *Fácil expansão*
- *Suporte a diversos sensores e atuadores*

4.3 Desvantagens

- *Dependência de conexão Wi-Fi*
- *Latência do sensor DHT22 é relativamente alta*
- *Dependência de broker público para testes*

4.4 Melhorias futuras

- *Interface web para visualização*
- *Implementação em hardware real*
- *Adição de novos sensores*

5. Referências

MQTT. Disponível em: <http://mqtt.org>

ESPRESSIF. ESP32 Documentation. Disponível em: <https://docs.espressif.com>

DHT22 Datasheet. Aosong Electronics.