

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ
CAMPUS CURITIBA
ENGENHARIA DE SOFTWARE

FELIPE CARNEIRO RIBEIRO
HENRIQUE DA COSTA RICARDO GONÇALVES
LEONARDO HENRIQUE CORREA

MONITORAMENTO DE TEMPERATURA E UMIDADE

CURITIBA

2025

FELIPE CARNEIRO RIBEIRO
HENRIQUE DA COSTA RICARDO GOLÇALVES
LEONARDO HENRIQUE CORREA

MONITORAMENTO DE TEMPERATURA E UMIDADE

Trabalho Docente Estruturante apresentado ao curso Engenharia de Software da Instituição Pontifícia Universidade Católica do Paraná - *Campus* Curitiba, como requisito parcial da disciplina Performance em Sistemas Ciberfísicos. Área de concentração: Sistemas embarcados e redes IoT.

Orientador: Prof. Fabio Garcez Bettio.

CURITIBA

2025

LISTA DE FIGURAS

Figura 1	— Esquemático do Circuito com ESP32.....	9
Figura 2	— Arquitetura do Sistemas.....	11
Figura 3	— Resultado do Teste do Sensor DHT22.....	12
Figura 4	— Resultado do Teste do Módulo Wi-Fi.....	14
Figura 5	— Resultado do Teste do Web Server.....	17
Figura 6	— Circuito Montado no Protoboard.....	17
Figura 7	— Interface Web do Sistema.....	17

LISTA DE TABELAS

Tabela 1 — Especificações do Sensor DHT22.....	9
Tabela 2 — Cronograma do Projeto.....	19

LISTA DE SIGLAS

DHT	Digital Humidity and Temperature
ESP	Espressif Systems
GPIO	General Purpose Input/Output
HTML	HyperText Markup Language
IOT	Internet das Coias
MQTT	Message Queuing Telemetry Transport
WIFI	Wireless Fidelity

SUMÁRIO

1 INTRODUÇÃO.....	7
1.1 OBJETIVO DO PROJETO	7
1.2 JUSTIFICATIVA	7
1.3 RELEVÂNCIA ACADÊMICA E PRÁTICA	7
2 FUNDAMENTAÇÃO TEÓRICA	8
2.1 Internet das Coisas (IoT).....	8
2.2 ESP32.....	8
2.3 Sensor DHT22	8
2.3 Protocolos de Comunicação	8
3 MATERIAIS E MÉTODOS	9
3.1 Tecnologias Utilizadas	9
3.1.1 Hardware.....	9
3.1.2 Software e Ferramentas.....	9
3.1.3 Bibliotecas Utilizadas.....	10
3.2 Arquitetura do Sistema	10
4 TESTES ISOLADOS.....	11
4.1 Sensor DHT22	11
4.2 MÓDULO WI-FI.....	12
4.3 WEB SERVER.....	13
5 RESULTADOS E DISCUSSÃO.....	17
5.1 ANÁLISE DE DESEMPENHO	17
5.2 LIMITAÇÕES E MELHORIAS.....	17
6 CONCLUSÃO.....	17
7 CRONOGRAMA	18
8 USO DO GIT E ORGANIZAÇÃO DA EQUIPE	18
REFERÊNCIAS.....	19

1 INTRODUÇÃO

A Internet das Coisas (IoT) tem transformado a forma como interagimos com o ambiente, possibilitando a coleta, transmissão e análise de dados em tempo real para diversas aplicações [1]. O monitoramento de variáveis ambientais, como temperatura e umidade, é essencial em setores como agricultura, saúde, indústria e tecnologia da informação, onde condições controladas garantem a qualidade de produtos, a segurança de equipamentos e o bem-estar humano. No entanto, soluções comerciais para monitoramento ambiental frequentemente são caras e complexas, limitando seu uso em pequenos negócios ou projetos acadêmicos.

Diante desse cenário, este projeto propõe o desenvolvimento de um sistema IoT baseado no microcontrolador ESP32, que integra o sensor DHT22 para monitoramento em tempo real de temperatura e umidade, com dados exibidos em uma interface web acessível via Wi-Fi. A solução visa ser de baixo custo, escalável e de fácil implementação, aplicável em contextos como residências, laboratórios, estufas e data centers.

Este trabalho tem como objetivo desenvolver e validar um sistema IoT para monitoramento de temperatura e umidade, utilizando o ESP32 e o sensor DHT22, com transmissão de dados via Wi-Fi e visualização em uma interface web.

1.1 OBJETIVO DO PROJETO

O objetivo principal é desenvolver um sistema IoT de baixo custo para monitoramento de temperatura e umidade, utilizando o microcontrolador ESP32 e o sensor DHT22. O sistema permite a coleta de dados em tempo real, sua transmissão via Wi-Fi e a visualização em uma interface web acessível por dispositivos conectados à mesma rede.

1.2 JUSTIFICATIVA

O objetivo principal é desenvolver um sistema IoT de baixo custo para monitoramento de temperatura e umidade, utilizando o microcontrolador ESP32 e o sensor DHT22. O sistema permite a coleta de dados em tempo real, sua transmissão via Wi-Fi e a visualização em uma interface web acessível por dispositivos conectados à mesma rede.

1.3 RELEVÂNCIA ACADÊMICA E PRÁTICA

O projeto alinha-se às demandas por soluções IoT em aplicações práticas, enquanto contribui para a formação acadêmica em Engenharia de Software, abordando conceitos de programação, eletrônica e comunicação de dados. A implementação prática fomenta a interdisciplinaridade, integrando hardware, software e redes, além de preparar os alunos para desafios do mercado de tecnologia.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Internet das Coisas (IoT)

A IoT refere-se à interconexão de dispositivos físicos à internet, permitindo a coleta e troca de dados para automação e monitoramento [1]. Sistemas IoT combinam sensores, microcontroladores, protocolos de comunicação (como Wi-Fi e MQTT) e interfaces de usuário para criar soluções inteligentes. No contexto deste projeto, a IoT é aplicada para monitorar variáveis ambientais em tempo real, com potencial para integração com plataformas de armazenamento na nuvem, como Firebase.

2.2 ESP32

O ESP32, desenvolvido pela Espressif Systems, é um microcontrolador de 32 bits com processador dual-core Tensilica Xtensa LX6, suporte a Wi-Fi (802.11 b/g/n) e Bluetooth, e diversos pinos GPIO para conexão de sensores e atuadores [4]. Sua versatilidade e baixo custo (aproximadamente R\$30-R\$50) tornam-no ideal para projetos IoT. O ESP32 suporta programação via Arduino IDE, facilitando o desenvolvimento de firmware.

2.3 Sensor DHT22

O DHT22 (AM2302) é um sensor digital de temperatura e umidade com resolução de 0,1 °C para temperatura (faixa: -40 °C a 80 °C, precisão: $\pm 0,5$ °C) e 0,1% para umidade (faixa: 0% a 100%, precisão: $\pm 2\%$) [3]. Seu tempo de resposta é de 2 segundos, com consumo de energia de 2,5 mA (máximo).

2.3 Protocolos de Comunicação

O sistema utiliza Wi-Fi para transmissão de dados e hospedagem de uma interface web. Opcionalmente, o protocolo MQTT (Message Queuing Telemetry Transport) pode ser implementado para comunicação eficiente entre dispositivos [5]. O MQTT é leve, baseado em publicação/assinatura, e ideal para dispositivos com recursos limitados.

3 MATERIAIS E MÉTODOS

3.1 Tecnologias Utilizadas

3.1.1 Hardware

- ESP32: Microcontrolador com Wi-Fi e 30 pinos GPIO.
- Sensor DHT22: Sensor digital com saída de temperatura e umidade.
- Protoboard e jumpers: Para conexões modulares.
- Fonte USB 5V: Alimentação do ESP32 (corrente típica: 200-500 mA).

Tabela 1: Especificações do Sensor DHT22

Parâmetro	Especificação
Faixa de Temperatura	-40 °C a 80 °C
Precisão de Temperatura	±0,5 °C
Faixa de Umidade	0% a 100%
Precisão de Umidade	±2%
Tempo de Resposta	2 segundos
Consumo de Energia	2,5 mA (máximo)

Fonte: [3].

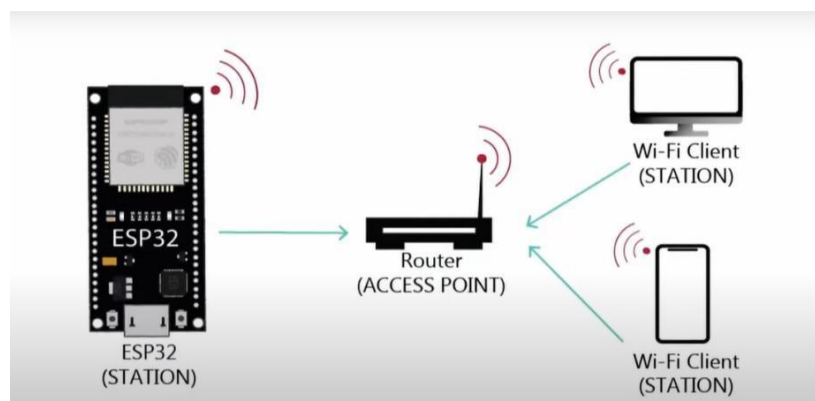


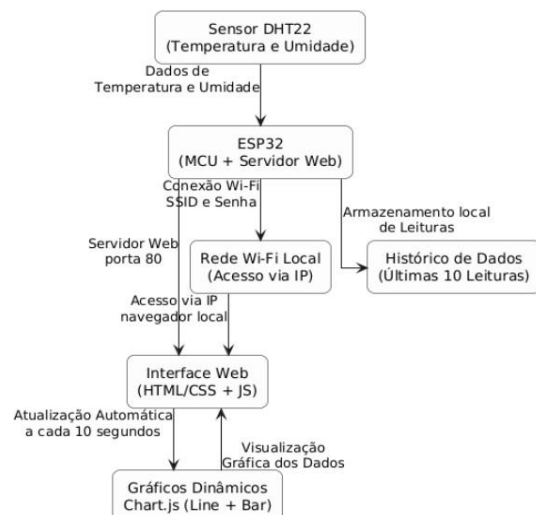
Figura 1: Esquemático do Circuito com ESP32

Fonte: Elaborado pelos autores (2025).

3.1.2 Software e Ferramentas

- Arduino IDE: Plataforma open-source para programação do ESP32[7].

Monitoramento de Temperatura e Umidade com ESP32 + DHT22



Arquitetura do Sistema - Monitoramento com ESP32



- Wokwi: Simulador online para validação de código [8].
- HTML/CSS: Para desenvolvimento da interface web
- MQTT (opcional): Protocolo para comunicação avançada

3.1.3 Bibliotecas Utilizadas

- Wifi.h: Gerencia conexões Wi-Fi.
- WebServer.h: Implementa um servidor HTTP no ESP32
- DHT.h: Interface com o sensor DHT22
- PubSubClient.h (opcional): Suporte ao protocolo MQTT[6].

3.2 Arquitetura do Sistema

O sistema integra o sensor DHT22 ao ESP32, que lê os dados e os disponibiliza via Wi-Fi em uma página HTML. A interface web exibe temperatura e umidade em tempo real, com atualização automática. Futuras melhorias podem incluir integração com um broker MQTT ou Firebase para armazenamento na nuvem.

Figura 2: Arquitetura do Sistema

Fonte: Elaborado pelos autores (2025).

4 TESTES ISOLADOS

4.1 Sensor DHT22

Objetivo: Verificar a leitura de temperatura e umidade no pino 4 do ESP32.

Procedimento:

- Conexões: VCC ao 3V3, GND ao GND, Data ao pino 4.
- Código carregado via Arduino IDE para leitura seriada.

Código:

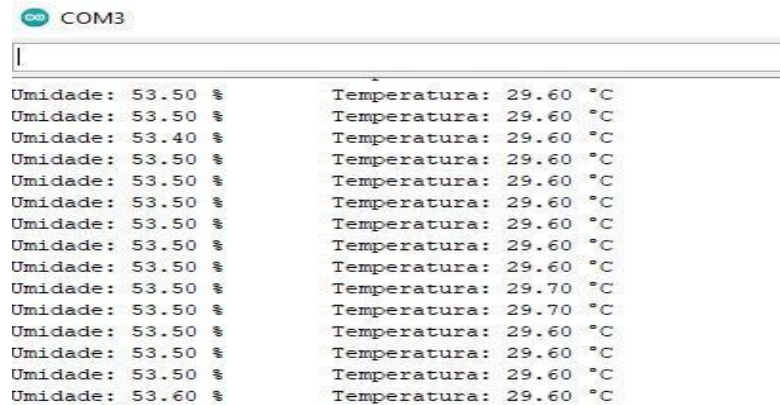
```
#include "DHT.h"
#define DHTPIN 4
#define DHTTYPE DHT 22

DHT dht (DHTPIN, DHTTYPE) ;

void setup() {
    Serial.begin(115200);
    Dht.begin();
}

Void loop() {
    float temp = dht.redTemperature();
    float umid = dht.redHumidity();
    Serial.print("Temperatura: ");
    Serial.print(temp);
    Serial.print(" °C| Umidade: ");
    Serial.print(umid);
    Serial.println("%");
    delay(2000);
}
```

Este código inicializa o sensor DHT22 e exibe leituras a cada 2 segundos no monitor serial. A biblioteca DHT.h simplifica a comunicação com o sensor.



Umidade: 53.50 %	Temperatura: 29.60 °C
Umidade: 53.50 %	Temperatura: 29.60 °C
Umidade: 53.40 %	Temperatura: 29.60 °C
Umidade: 53.50 %	Temperatura: 29.60 °C
Umidade: 53.50 %	Temperatura: 29.60 °C
Umidade: 53.50 %	Temperatura: 29.60 °C
Umidade: 53.50 %	Temperatura: 29.60 °C
Umidade: 53.50 %	Temperatura: 29.60 °C
Umidade: 53.50 %	Temperatura: 29.60 °C
Umidade: 53.50 %	Temperatura: 29.70 °C
Umidade: 53.50 %	Temperatura: 29.70 °C
Umidade: 53.50 %	Temperatura: 29.60 °C
Umidade: 53.50 %	Temperatura: 29.60 °C
Umidade: 53.50 %	Temperatura: 29.60 °C
Umidade: 53.60 %	Temperatura: 29.60 °C

Figura 3: Resultado do Teste do Sensor DHT22

Fonte: Elaborado pelos autores (2025).

4.2 MÓDULO WI-FI

Objetivo: Validar a conexão do ESP32 à rede Wi-Fi.

Procedimento:

- Configuração das credenciais de rede no código.
- Monitoramento da conexão via saída serial.

Código:

```
#include <WiFi.h>
const char *ssid = "REDE_WIFI";
const char *password = "SENHA";

void setup()
{
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  pinMode(2, OUTPUT);
  delay(10);

  Serial.print("Connecting to ");
  Serial.println(ssid);

  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
```

```

        Serial.print(".");
    }

    Serial.println("WiFi connected.");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}

void ST_conexao() {
    if(WiFi.status() == WL_CONNECTED)
        digitalWrite(2, HIGH);
    else
        digitalWrite(2, LOW);
}

void loop() {
    ST_conexao()

```

O código estabelece a conexão Wi-Fi. O endereço IP é exibido para acesso à interface web.

```

ets Jul 29 2019 12:21:46

rst:0x1 (POWERON_RESET),boot:0x17 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:4888
load:0x40078000,len:16516
load:0x40080400,len:4
load:0x40080404,len:3476
entry 0x400805b4
Connecting to troco senha por cerveja
...WiFi connected.
IP address:
192.168.68.118

```

Figura 4: Resultado do Teste do Módulo Wi-Fi

Fonte: Elaborado pelos autores (2025).

4.3 WEB SERVER

Objetivo: Hospedar uma página web com dados do sensor.

Procedimento:

- Implementação de um servidor HTTP no ESP32.
- Exibição de temperatura e umidade em uma página HTML.

Código:

```
#include
<WiFi.h>
#include
<WebServer.h>
#include
"DHT.h"

const char* ssid = "Rede
WiFi"; const char* password
= "Senha WiFi";

#define DHTPIN
4 #define
DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

WebServer server(80);

void handleRoot() {
    float h =
    dht.readHumidity();
    float t =
    dht.readTemperature();
    if (isnan(h) ||
    isnan(t)) {
        server.send(500, "text/plain", "Falha ao ler do sensor
        DHT!"); return;
    }

    String html = "<!DOCTYPE html><html><head><meta
```

```

    charset='UTF-8'>"; html += "<title>Clima
    Local</title></head><body>";

    html += "<h1>Dados do Sensor DHT22</h1>";
    html += "<p>Temperatura: " + String(t) +
    " ºC</p>"; html += "<p>Umidade: " + String(h)
    + " %</p>";

    html += "</body></html>";
    server.send(200, "text/html",
    html);
}

void setup()
{
    Serial.begin(1152
    00); dht.begin();
    WiFi.begin(ssid,
    password);

    Serial.print("Conectando ao
    WiFi"); while (WiFi.status() !=
    WL_CONNECTED) {
        delay(500);
        Serial.print
        (".");
    }

    Serial.println("\nConectado!");
    Serial.print("IP: ");
    Serial.println(WiFi.localIP());
    server.on("/", handleRoot);
    server.begin();

    Serial.println("Servidor HTTP iniciado");
}

void loop() {
    server.handleClient();

```

Este código cria um servidor web no ESP32, acessível pelo endereço IP, que exibe os dados do sensor em uma página HTML simples.

Dados do Sensor DHT22

Temperatura: 29.20 °C

Umidade: 52.30 %

Figura 5: Resultado do Teste do Web Server

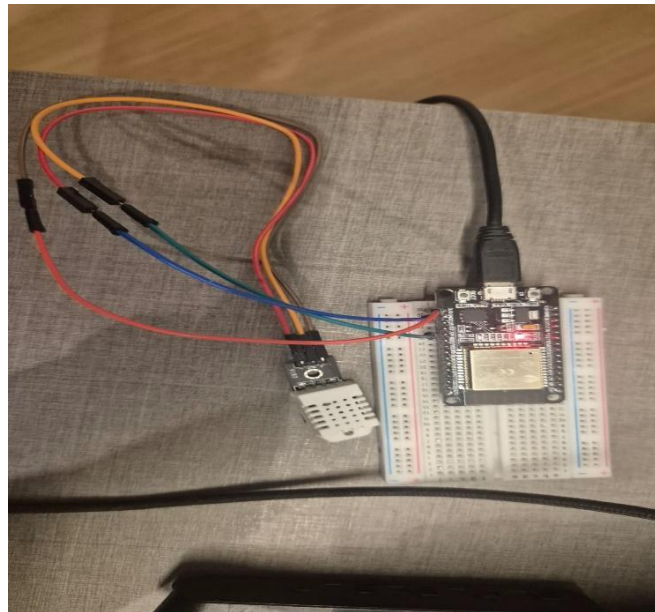
Fonte: Elaborado pelos autores (2025).

Figura 6: Circuito Montado no Protoboard

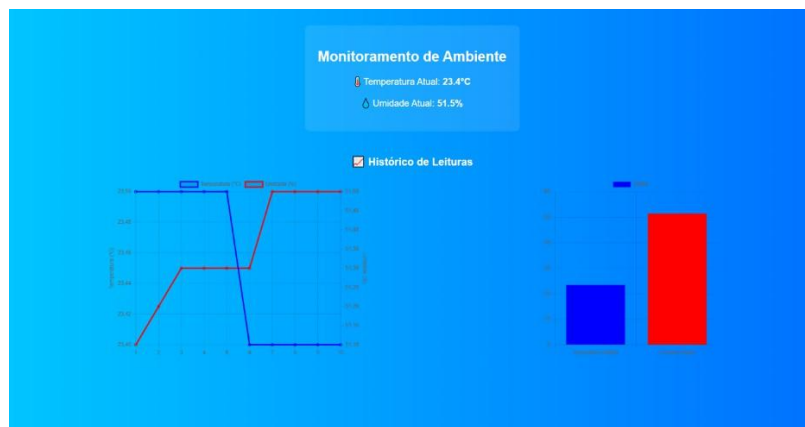
Fonte: Elaborado pelos autores (2025).

Figura 7: Interface Web do Sistema

Fonte: Elaborado pelos autores (2025).



Descrição: A página web do projeto permite acompanhar em tempo real a



temperatura e umidade medidas pelo sensor DHT22, com atualização automática

a cada 10 segundos. Os dados são apresentados em:

- Gráfico de linha com as últimas 10 leituras
- Gráfico de barras com as médias de temperatura e umidade.
- Indicadores atuais em destaque na tela.

A interface é gerada diretamente pelo ESP32 usando HTML, CSS e JavaScript. A biblioteca Chart.js é utilizada para exibir os gráficos, enquanto o endpoint /dados fornece os valores em formato JSON para atualização dinâmica.

Código do projeto: <https://github.com/Riquehenri/iot-monitoramento-temp>

5 RESULTADOS E DISCUSSÃO

Os testes realizados confirmaram a funcionalidade do sistema. O sensor DHT22 apresentou leituras consistentes, com temperatura variando entre 20 °C e 30 °C e umidade entre 40% e 60% em condições ambientais controladas.

5.1 ANÁLISE DE DESEMPENHO

O sistema demonstrou baixa latência na transmissão de dados, com tempo de resposta inferior a 1 segundo para acesso à página web na rede local. A precisão do DHT22 foi adequada para aplicações não críticas, mas pode ser limitada em ambientes com variações rápidas de temperatura ou umidade devido ao tempo de resposta de 2 segundos [3]. O consumo de energia do ESP32 foi estimado em 200 mA durante operação contínua, viável para alimentação via USB, mas sugerindo a necessidade de otimização para aplicações alimentadas por bateria.

5.2 LIMITAÇÕES E MELHORIAS

- **Limitações:** A interface web é básica e carece de recursos gráficos avançados, como gráficos históricos. O sistema opera apenas em redes locais, limitando o acesso remoto sem configurações adicionais.
- **Melhorias:** Implementar MQTT para comunicação com brokers externos, integrar Fire- base para armazenamento na nuvem, e desenvolver uma interface com bibliotecas JavaScript para visualização de dados históricos [5].

6 CONCLUSÃO

O projeto alcançou o objetivo de desenvolver um sistema IoT de baixo custo para monitoramento de temperatura e umidade, utilizando o ESP32 e o sensor DHT22. A solução demonstrou viabilidade técnica, com leituras precisas e interface web funcional, atendendo às necessidades de aplicações em pequena escala. O desenvolvimento proporcionou aprendizado em eletrônica embarcada, programação e redes, alinhado aos objetivos acadêmicos da disciplina Performance de Sistemas Ciberfísicos.

Para futuras iterações, recomenda-se a integração com plataformas de nuvem, como Firebase, para armazenamento e acesso remoto, e a implementação de MQTT para comunicação escalável. Além disso, a adição de gráficos interativos e alertas baseados em limites de temperatura e umidade pode ampliar as aplicações do sistema.

7 CRONOGRAMA

Tabela 2: Cronograma do Projeto

Semana	Atividade
1-2	Estudo de viabilidade e montagem do circuito
3-4	Programação do firmware no ESP32
5-6	Desenvolvimento da interface web
7-8	Testes integrados e entrega do projeto

Fonte: Elaborado pelos autores (2025).

8 USO DO GIT E ORGANIZAÇÃO DA EQUIPE

Foi criado um repositório no GitHub para melhor controle do projeto e versionamento. No repositório, o código Monitoramento do projeto ainda está sendo atualizado, e uma página README contém as instruções necessárias para entendê-lo, além de suas funcionalidades, componentes utilizados e explicações sobre a estrutura do código <https://github.com/Riquehenri/iot-monitoramento-temperatura-umidade>.

REFERÊNCIAS

- [1] Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A survey. *Computer Networks*, v. 54, n. 15, p. 2787-2805, 2010.
- [2] Barreto, R. A.; Costa, J. L. *Internet das coisas (IoT) com ESP32 e Arduino*. São Paulo: Novatec Editora, 2021.
- [3] Aosong Electronics. DHT22 AM2302 Datasheet. Disponível em: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>. Acesso em: 20 maio 2025.
- [4] Espressif Systems. ESP32 Technical Reference Manual. Disponível em: <https://www.espressif.com/en/products/socs/esp32/resources>. Acesso em: 20 maio 2025.
- [5] HiveMQ. MQTT Essentials. Disponível em: <https://www.hivemq.com/mqtt-essentials>. Acesso em: 20 maio 2025.
- [6] Knolleary. PubSubClient Library for Arduino. GitHub. Disponível em: <https://github.com/knolleary/pubsubclient>. Acesso em: 20 maio 2025.
- [7] Monk, S. *Programming Arduino: Getting Started with Sketches*. 2. ed. New York: McGraw-Hill Education, 2016.
- [8] Wokwi. Wokwi Online Simulator for ESP32, Arduino and IoT Projects. Disponível em: <https://wokwi.com/>. Acesso em: 20 maio 2025.