Universidade Federal de Campina Grande

Unidade Acadêmica de Engenharia Elétrica

Instrumentação Eletrônica 2024.1

Prof. Jaidilson Jó da Silva

SISTEMA INTELIGENTE DE MONITORAMENTO DE TEMPERATURA E UMIDADE BASEADO EM ESP32

GUSTAVO BORBUREMA RAMOS MEDEIROS

JOSÉ ROBERTO DO NASCIMENTO ARCANJO

LUCAS RODRIGUES ALBINO

ROGÉRIO MOREIRA ALMEIDA

THIAGO HENRIQUE DE OLIVEIRA SILVA

Campina Grande - PB
2024

Conteúdo

1 Introdução

Medir e controlar grandezas físicas são processos fundamentais para o pleno funcionamento de diversos sistemas. Nesse contexto, a Engenharia Elétrica, por meio da Instrumentação Eletrônica, desempenha o papel de conectar dados físicos a sistemas de medição e controle dessas grandezas por meio da interação entre sensores, circuitos eletrônicos, atuadores e interfaces de monitoramento.

Morris, em sua obra Measurement and Instrumentation: Theory and Application [1], destaca que as técnicas de medição acompanham a história da civilização desde os primórdios. Um exemplo clássico é o comércio baseado em escambo, que dependia da precisão de instrumentos de medição de massa para garantir a justiça nas trocas comerciais.

Nos dias atuais, as medições se concentram em sistemas eletrônicos, desde os simples termômetros de uso geral até sistemas complexos de monitoramento em grandes indústrias. Nesse contexto o monitoramento e o controle por meio de sistemas online vêm ganhando muita adesão devido à praticidade e a crescente confiabilidade nesses sistemas.

Dentre as mais diversas grandezas que podem ser medidas e controladas, como vibração, fluxo, velocidade, taxa de transmissão de dados. Este trabalho investiga a medição e o controle de temperatura e umidade em ambientes físicos do cotidiano, e propõe um sistema baseado em ESP32 e interface Web para aplicações práticas em ambientes residenciais.

No capítulo 2 foi investigado no estado da arte os princípios dos sistemas de intrumentação eletrônica de modo geral e no campo da medição de temperatura e umidade, e os instrumentos e técnicas que estão associadas nesse contexto. No capítulo 3 está descrito o projeto implementado abrangendo os sensores, o circuito de condicionamento dos sinais, a implementação da interface Web e as aplicações em que esse sistema pode ser empregado.

2 Fundamentação Teórica

No livro *Principles of measurement systems*, o autor J. P. Bentley [2] destaca que os sistemas de instrumentação geralmente seguem um padrão estrutural, que pode ser separado em quatro partes, que estão representadas na Figura 1. Na primeira, um sensor específico transforma uma grandeza física a ser medida em um sinal elétrico. Na segunda parte, o sinal passa para um circuito de condicionamento que faz a sua amplificação e filtragem, para que o seu processamento ocorra sem interferências. Na terceira parte, o sinal é então processado e transformado em uma forma adequada para apresentação. Finalmente, os dados são visualizados em uma interface adequada ao usuário, como um display digital ou gráfico.

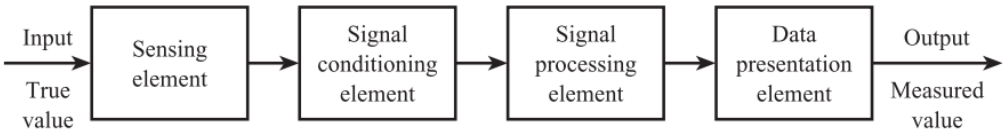

Figura 1: *Estrutura geral de um sistema de medição. Fonte: [2]*

A medição de temperatura é apontada como sendo a mais realizada de modo geral. Sua importância é explicada pela grande quantidade de processos que dependem de valores específicos de temperatura, além de dispositivos que necessitam controlar a própria temperatura durante o seu funcionamento. Associada à medição de temperatura, Morris[1] destaca que existe uma diferença no processo de medição quando comparada com outras grandezas, a exemplo de massa, distância, etc.

O fato é que, ao considerar a diferença de temperatura entre dois corpos, não há uma adição direta das temperaturas quando esses corpos são unidos, como ocorre com grandezas como comprimento ou massa, onde as magnitudes podem ser somadas. Isso torna mais complexa a definição de um padrão absoluto para temperatura, necessitando de pontos de referência fixos, como os pontos de fusão e ebulição de substâncias.

Assim, a temperatura não pode ser medida diretamente em relação a um padrão fundamental, mas sim por meio de referências bem estabelecidas, como a Escala Internacional de Temperatura Prática (IPTS), que utiliza transições claramente definidas entre os estados sólido, líquido e gasoso para padronizar as medições.

Segundo Morris [1], a medição de umidade é feita de diferentes maneiras. Destacam-se: O higrômetro elétrico, o psicrômetro e o medidor de ponto de orvalho, que se destaca por seu alto grau de precisão.

O higrômetro elétrico mede a variação de capacitância ou condutividade de um material higroscópico mediante a variação dos níveis de umidade. Já o psicrômetro utiliza dois sensores de temperatura, um seco e outro envolto em um pavio úmido, e mede a diferença de temperatura causada pela evaporação, que é relacionada ao nível de umidade.

Por fim, o medidor de ponto de orvalho utiliza um espelho resfriado eletricamente, que detecta a formação de orvalho em sua superfície, determinando o ponto em que o ar está saturado. Apesar de sua alta precisão, o medidor de ponto de orvalho requer cuidados com a limpeza do espelho, pois até pequenas contaminações podem causar grandes variações na medição.

O DHT11 [3] (Figura 2) é um sensor digital de temperatura e umidade que é capaz de medir ambas as grandezas considerando diferentes princípios físicos.

Este sensor utiliza um material higroscópico, que identifica variações de umidade a partir da variação da capacitância do material, que é afetada por essa variação de umidade. A precisão da medição de umidade relativa desse sensor varia de 20% a 90% com uma precisão de ±5% a 25°C.

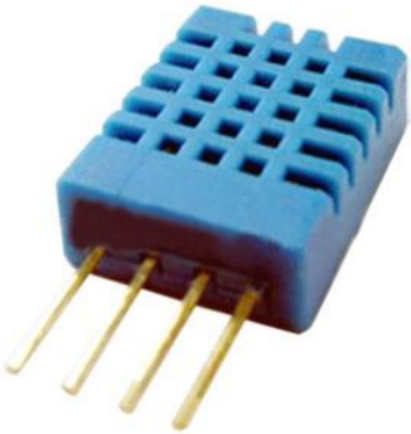

Figura 2: *Sensor DHT11. Fonte: [3]*

Para a medição de temperatura, o DHT11 utiliza um termistor que atua através da variação de sua resistência em função da temperatura do ambiente.

Desse modo quando a temperatura muda, a corrente que passa através do termistor sofre uma alteração e assim torna possível identificar os níveis de temperatura do ambiente. Vale destacar que este sensor é adequado para medir temperaturas entre 0°C e 50°C, com uma precisão de ±2°C.

3 Desenvolvimento do projeto

3.1 SISTEMA PARA MONITORAMENTO E CONTROLE DA TEMPERATURA E UMIDADE

Com base no contexto discutido nos Capítulos 1 e 2, foi desenvolvido um sistema para medição e controle de temperatura e umidade. Este sistema fornece leituras em tempo real dessas grandezas e permite o controle automático de atuadores específicos, por meio de uma interface web, proporcionando uma solução eficiente e de fácil acesso para monitoramento remoto.

Para a construção do projeto (Figura 3) foi utilizada um microcontrolador com alta capacidade de integração em diversos sistemas, o ESP32 [4]. Com ele é possível estabelecer conexão Wi-Fi e Bluetooth, o que o torna adequado para diversas aplicações de Internet das Coisas (IoT).

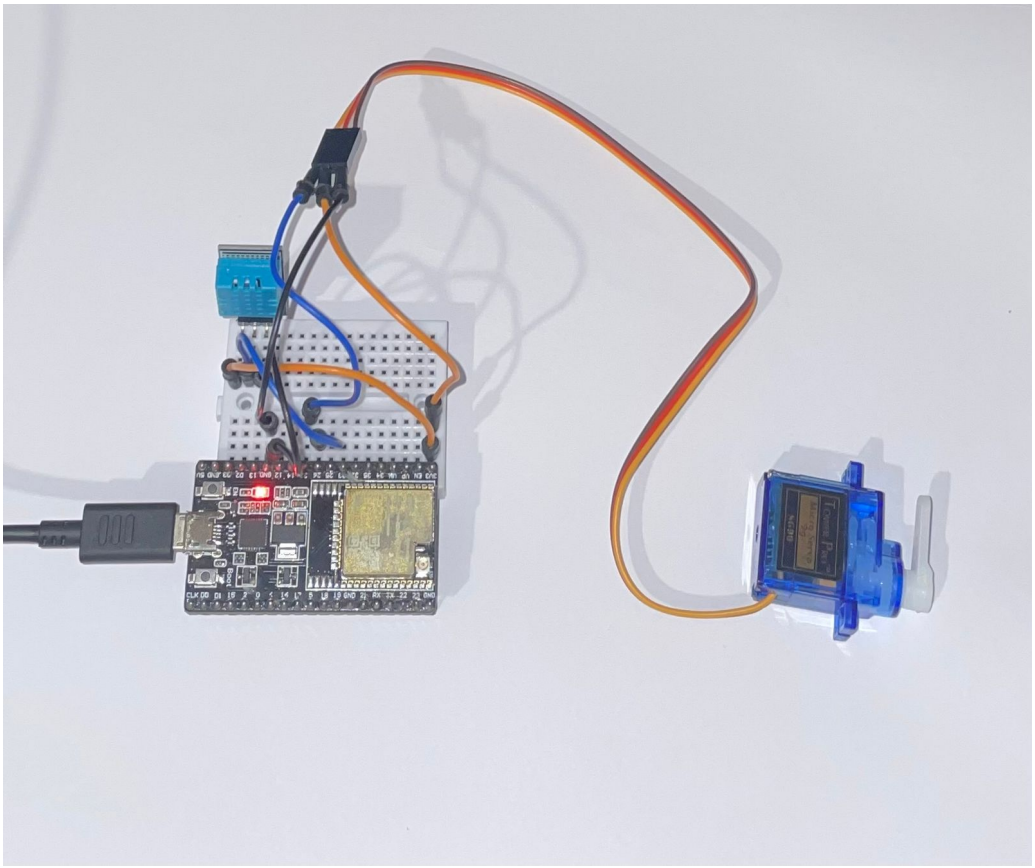

Figura 3: *Montagem do projeto*

Também foi utilizado um sensor de temperatura e umidade DHT11, e um servo motor para criar um sistema que monitora em tempo real as condições ambientais e exibe essas informações em uma página web acessível via Wi-Fi.

Além disso, o sistema possui um mecanismo para acionar o servo motor quando a temperatura ultrapassa um determinado limite, fazendo o motor oscilar entre duas posições (0° e 180°) e informando o estado do motor na interface web.

O código-fonte completo deste projeto está descrito no Apêndice A e também disponível publicamente no repositório do GitHub, acessível em [5]. O repositório contém a versão mais atualizada do código, além de instruções detalhadas sobre a implementação e o uso do sistema.

3.2 FUNCIONALIDADES

- **Monitoramento de Temperatura e Umidade:** Leitura contínua dos valores do sensor DHT11.
- **Interface Web em Tempo Real:**Uma página web moderna com tema escuro que mostra a temperatura e a umidade em tempo real, além do estado do servo motor.
- **Acionamento do Servo Motor:**Quando a temperatura ultrapassa um valor limite (28°C por padrão), o servo motor começa a oscilar, e essa ação é refletida na página web.

3.3 COMPONENTES UTILIZADOS

- **ESP32 (DOIT DEV KIT V1):**Microcontrolador responsável por gerenciar o sensor, o servo motor e a página web.
- **Sensor DHT11:** Sensor responsável pela leitura da temperatura e umidade.
- **Servo Motor:** Motor controlado que oscila entre 0° e 180° quando a temperatura excede o limite.

3.4 CIRCUITO DE CONDICIONAMENTO DO SINAL

O circuito de condicionamento do sinal para este projeto está representado na Figura 4. Ele conecta o sensor DHT11 ao ESP32 da seguinte maneira: o terminal V_{CC} é ligado ao pino de 5 V (fio vermelho), o GND ao terra (fio preto), e o pino de sinal é conectado ao GPIO26 do ESP32 (fio azul).

Os dados de temperatura e umidade são transmitidos em tempo real por meio de um sinal digital, permitindo ao ESP32 ler as medições e controlar outros dispositivos conforme necessário.

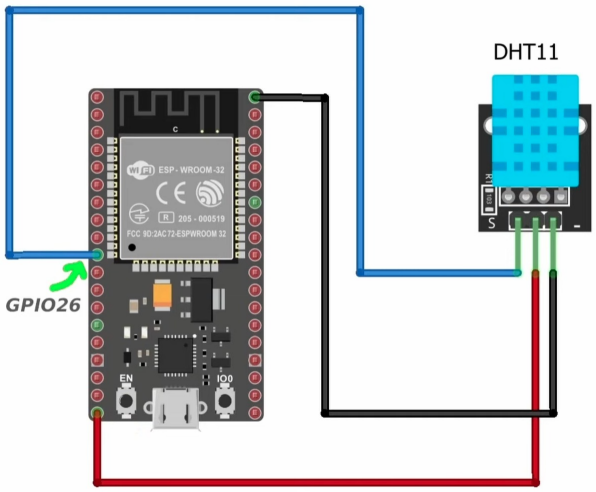

Figura 4: *Circuito de condicionamento do sinal*

3.5 VISUALIZAÇÃO DAS GRANDEZAS E CONTROLE DO ATUADOR

A visualização dos dados, bem como o estado do acionamento do atuador é feita via uma interface Web. O ESP32 atua como um servidor, de modo a transmitir os dados coletados para um endereço de IP.

A interface foi desenvolvida utilizando HTML e JavaScript (Apêndice A), o que tornou possível uma atualização constante das leituras.

A Figura 5 mostra o feedback do sistema durante a conexão com a interface Web, que está representada na Figura 6.

3.6 APLICAÇÕES

Considerando as limitações do sensor e do microcontrolador, este projeto é adequado para ambientes que não possuam níveis extremos de temperatura ou umidade. Pode ser aplicado em diversos contextos, como:

- Ambientes residenciais para monitoramento de conforto térmico;
- Escolas e escritórios;
- Estufas, onde o controle da temperatura e umidade é essencial para o desenvolvimento de plantas;

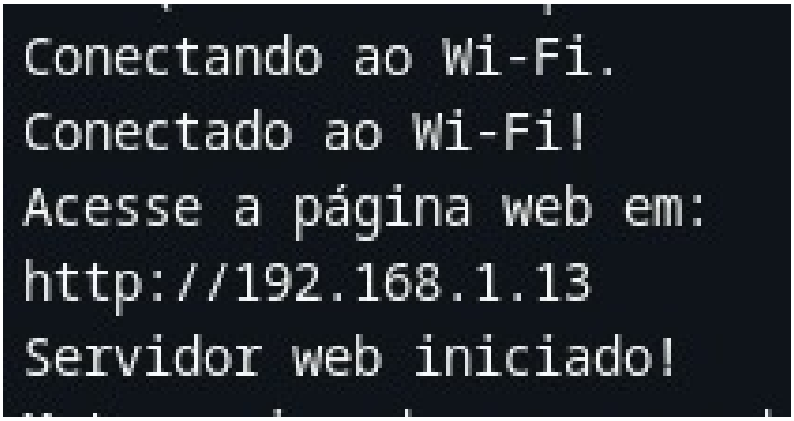

Figura 5: *Conexão da interface Web*

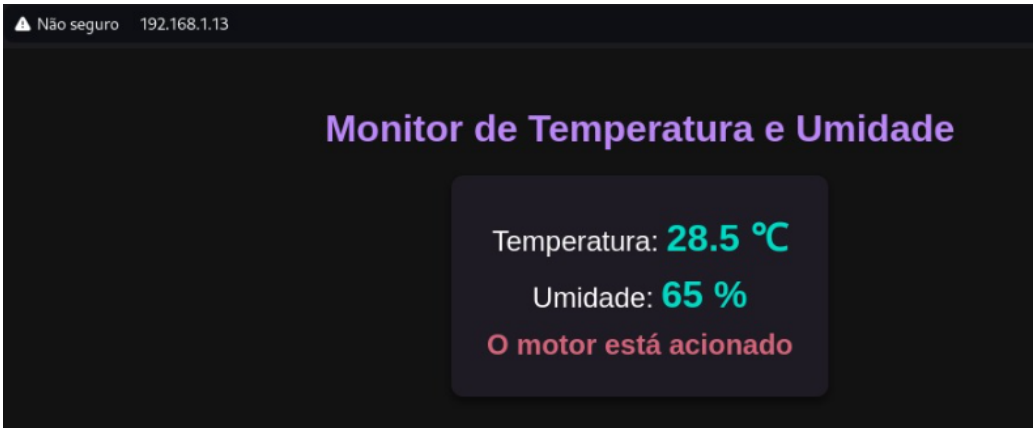

Figura 6: *Interface Web*

- Armazéns e estoques que requerem condições específicas de temperatura e umidade para preservar materiais ou produtos.

Além das aplicações já mencionadas em ambientes residenciais, o sistema pode ser utilizado em diferentes contextos de controle e monitoramento ambiental. Algumas possibilidades incluem:

- **Automação residencial:** O sistema pode ser integrado com sistemas maiores de automação para controlar dispositivos como ar-condicionado ou aquecedores de maneira automática;
- **Pequenos negócios:** Restaurantes, farmácias e clínicas veterinárias podem se beneficiar de monitoramento contínuo das condições ambientais para garantir a preservação de alimentos, medicamentos e animais em ambientes controlados.
- **Monitoramento de equipamentos eletrônicos:**Em ambientes com servidores e outros equipamentos sensíveis ao calor, o sistema pode auxiliar no monitoramento da temperatura ambiente e acionar alertas automáticos para evitar superaquecimento.

4 Considerações Finais

Este trabalho realizou uma avaliação geral a cerca da importância da instrumentação eletrônica nos mais diversos contextos e direcionou o foco para a medição de temperatura e umidade. Nesse contexto, foram apresentados os princípios físicos e técnicas de medição que envolvem tais grandezas.

Neste cenário foi proposto um sistema de monitoramento e controle de temperatura e umidade em ambientes não severos, se baseando na utilização do microcontrolados ESP32 e do sensor DHT11 que é capaz de medir temperatura e umidade com precisão.

Evidencia-se que o sistema demonstrou-se uma solução eficaz para diversas aplicações no cotidiano, principalmente em ambientes residenciais e pequenos negócios, devido a integração de sensores e uma interface web acessível que permite ao usuário monitorar em tempo real as condições ambientais e automatizar ações conforme necessário, como o acionamento de atuadores.

Por fim, destaca-se o potencial para futuras expansões, como a inclusão de mais sensores e atuadores, graças à praticidade e flexibilidade do projeto.

A Código fonte do projeto

```
#include <Arduino.h>
#include <WiFi.h>
#include <WebServer.h>
#include <DHT.h>
#include <Adafruit_Sensor.h>
#include <ESP32Servo.h>


const char* ssid = "REDE-WIFI";
const char* password = "SENHA-WIFI";


WebServer server(80);


#define DHTPIN 26
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);


Servo servo;
#define SERVOPIN 27
const int servoOnPos = 180;
const int servoOffPos = 0;
bool motorAcionado = false;
bool oscilarMotor = false;
unsigned long lastOscillationTime = 0;
const int oscillationInterval = 1000;

const float tempThreshold = 28.0;

String getPage() {
  String motorState = motorAcionado ? "O motor está acionado" : "O motor está
    desativado";

  String html = "<!DOCTYPE html><html>";
  html += "<head><meta charset='utf-8'><meta name='viewport' content='width=device-
    width, initial-scale=1.0'>";
  html += "<title>Monitor de Temperatura e Umidade</title>";
  html += "<style>body{background-color:#121212;color:#ffffff;font-family:Arial,
    sans-serif;text-align:center;margin-top:50px;}";
  html += "h1{color:#BB86FC;}p{font-size:1.5rem;margin:10px;}#container{display:
    inline-block;padding:20px;border-radius:10px;background-color:#1F1B24;box-shadow
    :0 4px 8px rgba(0,0,0,0.5);}";
  html += "#temp, #humidity{font-size:2rem;font-weight:bold;color:#03DAC5;}";
  html += "#motor{font-size:1.5rem;color:" + String(motorAcionado ? "#03DAC5" : "#
    CF6679") + ";font-weight:bold;}</style>";
  html += "<script>setInterval(function(){getData();}, 2000);function getData(){var
    xhttp=new XMLHttpRequest();xhttp.onreadystatechange=function(){if(this.
    readyState==4&&this.status==200){var data=JSON.parse(this.responseText);document
    .getElementById('temp').innerHTML=data.temperature+' &#8451;';document.
```

```
    getElementById('humidity').innerHTML=data.humidity+' %';document.getElementById(
    'motor').innerHTML=data.motorState;}};xhttp.open('GET', '/data', true);xhttp.
    send();}</script>";
  html += "</head><body>";
  html += "<h1>Monitor de Temperatura e Umidade</h1>";
  html += "<div id='container'>";
  html += "<p>Temperatura: <span id='temp'>-- &#8451;</span></p>";
  html += "<p>Humidade: <span id='humidity'>-- %</span></p>";
  html += "<p id='motor'>" + motorState + "</p>";
  html += "</div></body></html>";
  return html;
}

void controlaServo(float temperatura) {
  if (temperatura > tempThreshold) {
    if (!oscilarMotor) {
      oscilarMotor = true;
      motorAcionado = true;
      Serial.println("Motor acionado e começando a oscilar.");
    }
  } else {
    if (oscilarMotor) {
      oscilarMotor = false;
      servo.write(servoOffPos);
      motorAcionado = false;
      Serial.println("Motor desativado e oscilação parada.");
    }
  }

  if (oscilarMotor) {
    unsigned long currentTime = millis();
    if (currentTime - lastOscillationTime >= oscillationInterval) {
      static bool oscilarParaDireita = true;
      if (oscilarParaDireita) {
        servo.write(servoOnPos);
        Serial.println("Motor para a direita (180 graus).");
      } else {
        servo.write(servoOffPos);
        Serial.println("Motor para a esquerda (0 graus).");
      }
      oscilarParaDireita = !oscilarParaDireita;
      lastOscillationTime = currentTime;
    }
  }
}

void handleData() {
  float temp = dht.readTemperature();
  float humidity = dht.readHumidity();

  if (isnan(temp) || isnan(humidity)) {
    server.send(500, "text/plain", "Erro ao ler o sensor DHT");
    return;
  }

  controlaServo(temp);

  // Monta a resposta JSON
```

```
98    String json = "{\"temperature\": " + String(temp, 1) + ", \"humidity\": " + String
        (humidity, 1) + ", \"motorState\": \"" + (motorAcionado ? "O motor está acionado
        " : "O motor está desativado") + "\"}";
99    server.send(200, "application/json", json);
100 }
101
102 void handleRoot() {
103   String html = getPage();
104   server.send(200, "text/html", html);
105 }
106
107 void setup() {
108   Serial.begin(115200);
109
110   dht.begin();
111   delay(2000);
112
113   servo.attach(SERVOPIN);
114   servo.write(servoOffPos);
115
116   // Conectando ao Wi-Fi
117   Serial.print("Conectando ao Wi-Fi");
118   WiFi.begin(ssid, password);
119   while (WiFi.status() != WL_CONNECTED) {
120     delay(1000);
121     Serial.print(".");
122   }
123   Serial.println("");
124   Serial.println("Conectado ao Wi-Fi!");
125
126   Serial.println("Acesse a página web em: ");
127   Serial.print("http://");
128   Serial.println(WiFi.localIP());
129
130   server.on("/", handleRoot);
131
132   server.on("/data", handleData);
133
134   server.begin();
135   Serial.println("Servidor web iniciado!");
136 }
137
138 void loop() {
139   server.handleClient();
140 }
```

Código 1: *Codigo Fonte*

Referências Bibliográficas

[1] A. S. Morris e R. Langari, *Measurement and instrumentation: theory and application*. Academic Press, 2011.

[2] J. P. Bentley, *Principles of measurement systems*. Pearson education, 2005.

[3] Mouser, *DHT11*, Acesso: 15/09/2024, 2024. URL: https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf?srsltid=AfmBOoofLRAohSRsLPb7r3sNojeC9QG0yRViROgiO5ZNRx7Bw15pmNGO.

[4] DOIT, *esp32 devkit v1*, Acesso: 15/09/2024, 2024. URL: https://roboeq.ir/files/id/4034/name/ESP32%20MODULE.pdf/.

[5] L. R. Albino, *ESP32-DHT11*, https://github.com/LucasRAlbino/ESP32-DHT11, Acessado em: 29/09/2024, 2024.