

UNIVERSIDADE SÃO FRANCISCO

RELATÓRIO PROJETO DE INSTRUMENTAÇÃO ELETROELETRÔNICA

SIMULADOR DE ESTUFA

Campus:	BRAGANÇA PAULISTA		Curso:	Eng. Computação/Eletrica	
Disciplina:	Instrumentação Eletrônica		Professor	Andre Renato Bakalereskis	
Semestre:		Turma:		Data:	24/05/2023
NOME				R.A.	
Ana Flávia Alves Pereira				001202004440	
Gabriel de Assis Gomes				001202010614	
Pedro de Camargo Zago				001202002325	
Samuel William dos Santos				001201914261	
Samuel Amaral Pereira				001202005737	



1. INTRODUÇÃO

Os conversores analógico-digitais (ADCs) são dispositivos que convertem um sinal analógico, como uma tensão elétrica, em um sinal digital, que pode ser processado por um computador ou microcontrolador. Os sensores fotoelétricos e indutivos são exemplos de sensores que produzem sinais analógicos, que podem ser convertidos em sinais digitais por um ADC. O ESP32 é uma plataforma de hardware e software de código aberto que pode ser usada para controlar e interagir com diferentes dispositivos, incluindo sensores e conversores ADC. Ao utilizar esses conceitos de forma paralela, é possível criar projetos que envolvam a leitura e processamento de sinais analógicos gerados por sensores, bem como a conversão de sinais analógicos em sinais digitais para posterior processamento. Isso pode ser útil em diversas aplicações, incluindo sistemas de controle, automação, medição de temperatura, níveis de líquidos, controle de iluminação, entre outros.

Neste experimento utilizou-se um sensor DHT22 para a captação de umidade e temperatura e a montagem de um sensor utilizando um LDR para dar as informações de luminosidade da estufa.

2. INSTRUMENTOS UTILIZADOS NA PRÁTICA

- ESP32
- DHT22
- Resistores de 10kΩ
- LDR 10mm
- Jumpers
- Protoboard
- Maquete de uma estufa

3. COMPONENTES

3.1. SENSOR FOTOELÉTRICO – LDR

Um sensor fotoelétrico é um dispositivo que utiliza a luz para detectar a presença ou ausência de objetos em um ambiente. Pode atuar tanto como um transdutor quanto como um sensor. O sensor fotoelétrico pode ser definido como um transdutor, pois é capaz de converter energia luminosa em energia elétrica, assim como as fotocélulas que convertem diretamente a luz em energia elétrica.

Existem também sensores que convertem a luz em uma variação de uma grandeza elétrica específica, como resistência elétrica ou tensão, por exemplo. Exemplos de sensores fotoelétricos incluem LDRs e fotodiodos. Esses sensores emitem um feixe de luz e medem a quantidade de luz refletida de volta para o sensor. Dessa forma, quando um objeto atravessa o caminho do feixe de luz, a quantidade de luz refletida é alterada e o sensor detecta essa mudança.

Esses sensores são amplamente utilizados em aplicações industriais, como em linhas de produção automatizadas, sistemas de controle e sistemas de controle de acesso. Eles são adequados para detecção de objetos opacos, translúcidos ou refletivos, dependendo do tipo de sensor e da fonte de luz utilizada.

Imagem 1 - LDR

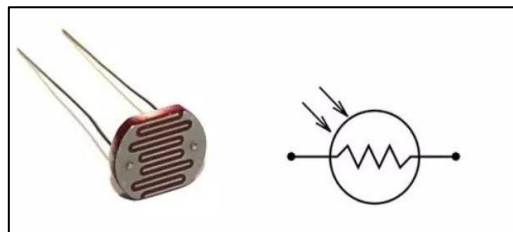
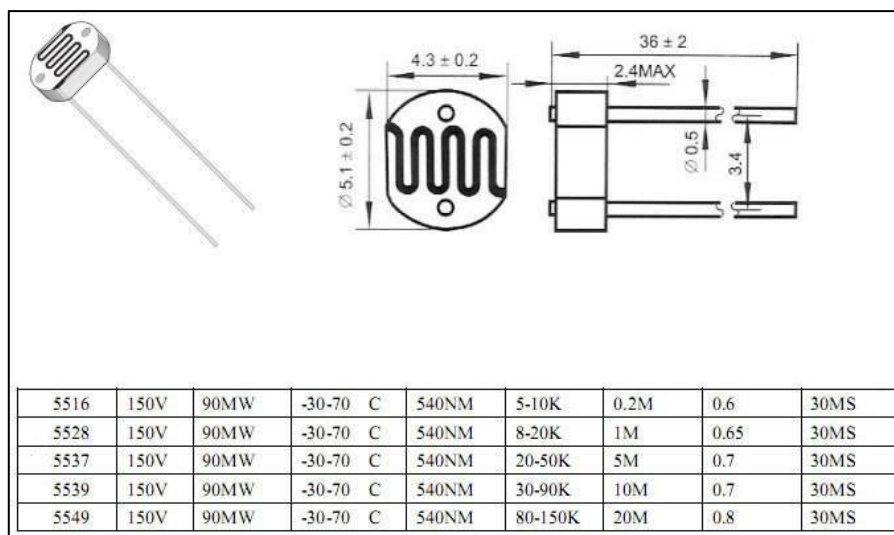


Imagem 2 – Datasheet LDR



3.2. SENSOR DE UMIDADE E TEMPERATURA – DHT22

O sensor DHT22 é um dispositivo eletrônico usado para medir a umidade relativa e temperatura do ar. Com sua capacidade de fornecer leituras precisas, é amplamente

utilizado em várias aplicações, desde monitoramento climático em residências até controle industrial. O sensor utiliza um sensor capacitivo para medir a umidade relativa, convertendo as variações em leituras confiáveis. Além disso, possui um termistor interno para medir a temperatura ambiente e transmite as informações por meio de um protocolo digital para outros dispositivos ou sistemas.

Uma das principais vantagens do sensor DHT22 é a sua precisão e estabilidade. Com uma faixa de medição de umidade de 0% a 100% e uma precisão de $\pm 2\%$, juntamente com uma faixa de medição de temperatura de -40°C a 80°C e uma precisão de $\pm 0,5^{\circ}\text{C}$, oferece resultados confiáveis. Além disso, é fácil de usar, exigindo apenas uma conexão de alimentação e comunicação para funcionar corretamente.

Graças à sua confiabilidade e precisão, o sensor DHT22 é amplamente empregado em várias aplicações, como sistemas de controle de climatização, estações meteorológicas e dispositivos de monitoramento de ambientes internos. Ele fornece informações valiosas sobre as condições do ar, auxiliando no controle de processos e no monitoramento ambiental para garantir conforto e segurança em diferentes cenários.

Imagem 3 – Datasheet DHT22

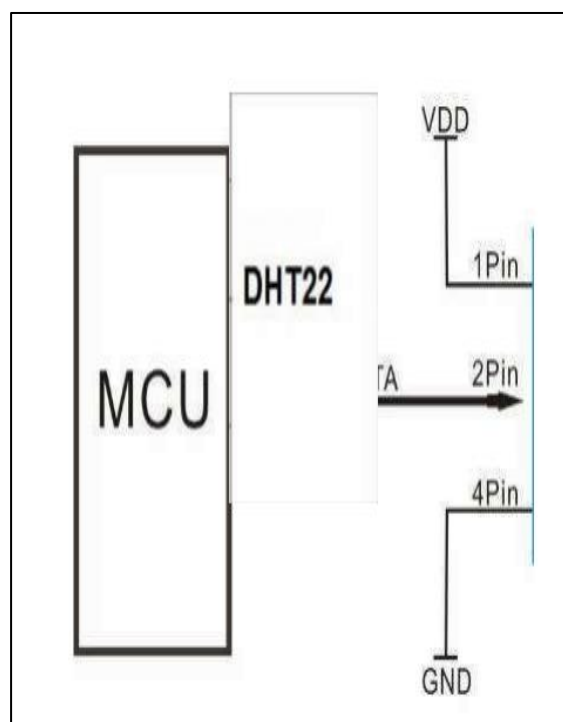
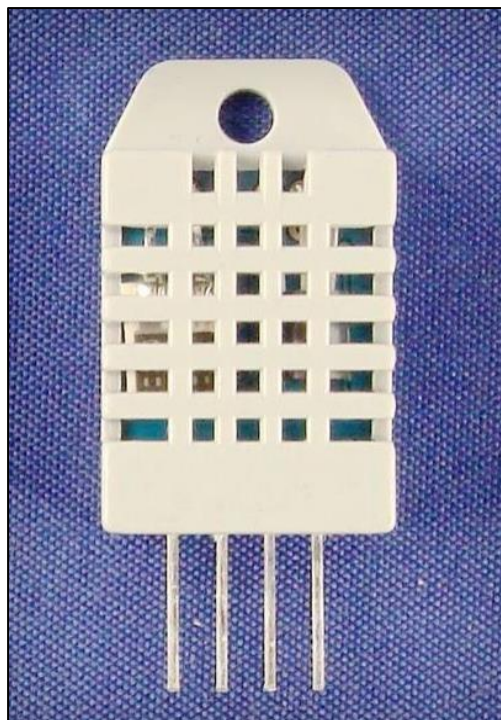


Imagem 4 – Componente DHT22



3.3. MICROCONTROLADOR – ESP32

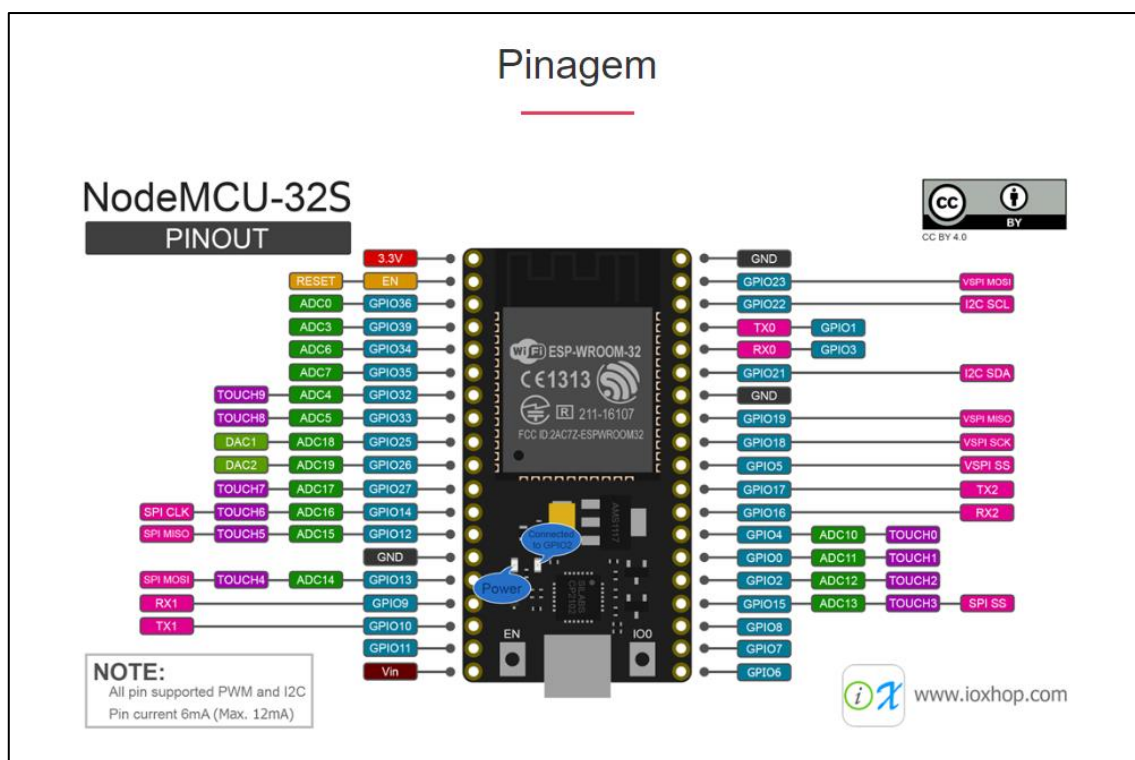
O ESP32 é um microcontrolador de baixo custo e com baixo consumo de energia, desenvolvido pela Express Systems. Ele é projetado para ser um microcontrolador altamente integrado, capaz de executar várias tarefas simultaneamente. Destina-se especialmente a fornecer versatilidade, robustez e confiabilidade em muitas aplicações. É amplamente utilizado em projetos de IoT, devido à sua capacidade de se conectar a redes Wi-Fi e Bluetooth. Com ele é possível desenvolver sistemas de monitoramento e controle remoto, dispositivos de automação residencial, soluções de segurança, aplicações de sensoriamento e muito mais.

3.3.1. ARQUITETURA – CARACTERÍSTICAS

- Microprocessador dual-core Tensilica LX6 240 MHz com desempenho de 600 DMIPS;
- Possui 520 KBy SRAM, há 16 MB de memória flash na placa;
- O chip é alimentado por uma tensão de 2,2 a 3,6 V;
- Temperatura ambiente possível de -40 °C a 125 °C;
- Em sono profundo, o ESP32 consome 2,5 μ A, de acordo com o fabricante;
- Baixo consumo de energia é fornecido por um processador ULP (Ultra-Low-Power)

- Até 8 Kbytes de dados e programas podem ser armazenados na SRAM do RTC (relógio de tempo real) para que o acesso a temporizadores, interrupções e periféricos seja possível, mesmo no modo de hibernação profunda;
- Para se comunicar com o mundo externo, o SoC (Sistema em um Chip) inclui o componente HT40 802.11b / g / n WiFi e a funcionalidade Bluetooth;
- Além da antena PCB integrada, um componente externo pode ser conectado por meio de um conector IPEX;
- Como sensores, o ESP32 possui um sensor Hall, uma touch button capacitivo, um amplificador analógico de baixo sinal e um quartzo de cristal de 32 kHz.

Imagem 5 – ESP32 Esquema



4. FUNÇÃO DO SENSOR MONTADO PELO GRUPO

O sensor de luminosidade utilizado neste projeto é composto por um LDR em conjunto com um resistor e integrado ao ESP32. Esse sistema foi projetado para realizar medições de luminosidade do ambiente.

No circuito, o LDR é conectado em série com um resistor fixo. Essa configuração forma um divisor de tensão, onde a tensão é distribuída proporcionalmente entre os dois

resistores de acordo com suas resistências. O ESP32, que atua como microcontrolador, está configurado para medir a tensão resultante do divisor de tensão.

Quando a luminosidade do ambiente aumenta, a resistência do LDR diminui, o que resulta em uma queda de tensão no LDR. Consequentemente, a tensão medida pelo ESP32 é reduzida. Por outro lado, em ambientes menos iluminados, a resistência do LDR aumenta, resultando em uma tensão maior medida pelo ESP32.

O ESP32, por sua vez, realiza a conversão analógico-digital da tensão medida e utiliza algoritmos e lógica de programação para interpretar os valores obtidos e fornecer uma medição de luminosidade em um formato adequado para exibição ou tomada de decisões.

Dessa forma, o sistema com o LDR, o resistor e o ESP32 são capazes de captar variações na luminosidade do ambiente e fornecer leituras que podem ser utilizadas para diversas finalidades, como controle de iluminação automatizado, monitoramento ambiental ou análise de dados de luminosidade.

5. METODOLOGIA

Para a elaboração do projeto foi necessário o desenvolvimento de uma lógica em linguagem C para que o ESP32 processasse as informações dos sensores e convertesse os mesmos em valores sinais analógicos. Para armazenar os resultados de cada uma das medições foi utilizado um banco de dados MySQL que será consumido posteriormente pela nossa dashboard e em tempo real, irá nós mostrar os a temperatura, luminosidade e humidade do ambiente da estufa.

Abaixo está o código que utilizamos para obtermos os resultados:

```
#include "DHT.h" // Inclui a biblioteca DHT
#include <WiFi.h> // Inclui a biblioteca WiFi

#define LDR_pin 34 // Pino do sensor de luminosidade (LDR)
#define DHT_pin 4 // Pino do sensor DHT
#define DHT_type DHT22 // Tipo de sensor DHT 22)
DHT dht(DHT_pin, DHT_type); // Criação de uma instância do objeto DHT

int luminosity = 0; // Variável para armazenar a luminosidade
```

```

const char * ssid = ""; // Nome da rede WiFi
const char * password = ""; // Senha da rede WiFi
const char * host = ""; // Endereço do host para conexão

void setup() {
    Serial.begin(9600); // Inicia a comunicação serial
    dht.begin(); // Inicializa o sensor DHT
    Serial.println();
    Serial.println();
    Serial.print("Conectando à rede "); // Imprime mensagem indicando a tentativa de conexão à
redeF
    Serial.println(ssid); // Imprime o nome da rede WiFi
    WiFi.begin(ssid, password); // Inicia a conexão WiFi
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi conectado"); // Imprime mensagem indicando que a conexão WiFi foi
estabelecida
    Serial.println("Endereço IP: ");
    Serial.println(WiFi.localIP()); // Imprime o endereço IP atribuído ao dispositivo
}

void loop() {
    luminosidade = analogRead(LDR_pin); // Lê o valor de luminosidade do sensor LDR
    float temperature = dht.readTemperature(); // Lê a temperatura do sensor DHT
    float humidity = dht.readHumidity(); // Lê a umidade do sensor DHT
    if (isnan(temperature) || isnan(humidity)) { // Verifica se temperature e humidity são números
        Serial.println("Falha na comunicação com o DHT22"); // Imprime mensagem indicando falha na
comunicação com o sensor DHT
    } else {

```



```

Serial.print("Luminosidade do ambiente: ");
Serial.print(luminosity);
Serial.print(" - Umidade: ");
Serial.print(humidity);
Serial.print("%");
Serial.print(" - Temperatura: ");
Serial.print(temperature);
Serial.println(" *C");
Serial.print("Conectando a ");
Serial.println(host); // Imprime o endereço do host para conexão
WiFiClient client;
const int httpPort = 80;
if (!client.connect(host, httpPort)) { // Realiza a conexão com o host
    Serial.println("Falha na conexão"); // Imprime mensagem indicando falha na conexão com o
host
    return;
}
client.print(String("GET http://localhost/dashboard_arduino/connect.php?") + ("&temperature=")
+ temperature + ("&humidity=") + humidity + ("&luminosity=") + luminosity + " HTTP/1.1\r\n" +
"Host: " + host + "\r\n" + "Conexão: fechada\r\n\r\n");

unsigned long timeout = millis();
while (client.available() == 0) {
    if (millis() - timeout > 1000) {
        Serial.println(">>> Timeout no cliente !"); // Imprime mensagem indicando timeout no cliente
        client.stop();
        return;
    }
}
while (client.available()) {
    String line = client.readStringUntil('\r');
    Serial.print(line);
}

```

```

Serial.println();

Serial.println("Fechando conexão");

}

delay(3000);

}

```

6. CIRCUITO

Imagem 6 – Esquema elétrico do projeto

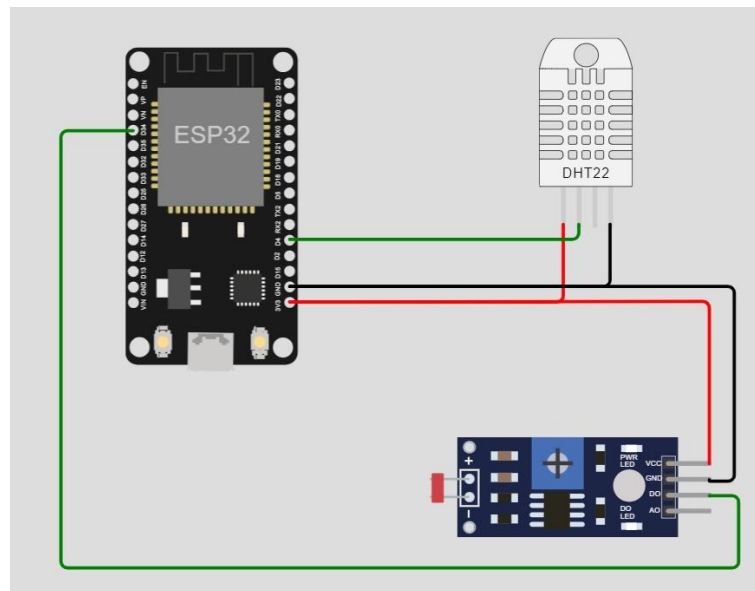
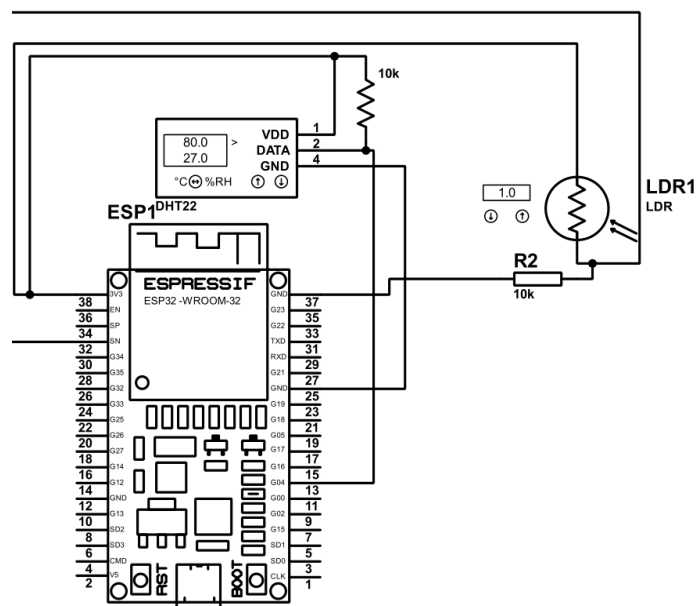


Imagem 7 – Diagrama do projeto



7. RESULTADOS

Sensor de luminosidade (LDR): é retornado valores que variam de 0 a 4095, quanto maior o valor, maior é a luminosidade no ambiente que está o sensor. Tendo esses dados, na dashboard é exibido um valor em porcentagem, exemplo: valor de 2500, terá uma porcentagem de 61% de luminosidade no ambiente.

Sensor de temperatura e humidade (DHT22): o sensor fornece leituras de temperatura em graus Celsius (°C) e umidade relativa em porcentagem (%).

Imagem 8 – Dados capturados pelos sensores

temperature	humidity	created_at	luminosity
22.1	59	2023-05-18 16:30:56	2379
21.9	57	2023-05-18 16:31:00	2384
22	58	2023-05-18 16:31:05	2379
22	58	2023-05-18 16:31:09	2344
22.1	57	2023-05-18 16:31:13	2352
22.1	57	2023-05-18 16:31:17	2364
20.6	61	2023-05-19 15:01:01	1991
20.4	61	2023-05-19 15:01:05	1987
20.5	61	2023-05-19 15:01:09	1986
20.5	61	2023-05-19 15:01:13	1986
20.5	60	2023-05-19 15:01:18	1946
20.6	60	2023-05-19 15:01:22	1978
20.5	60	2023-05-19 15:01:26	1981

8. MONTAGEM DA MAQUETE

Imagem 9 – Montagem do circuito

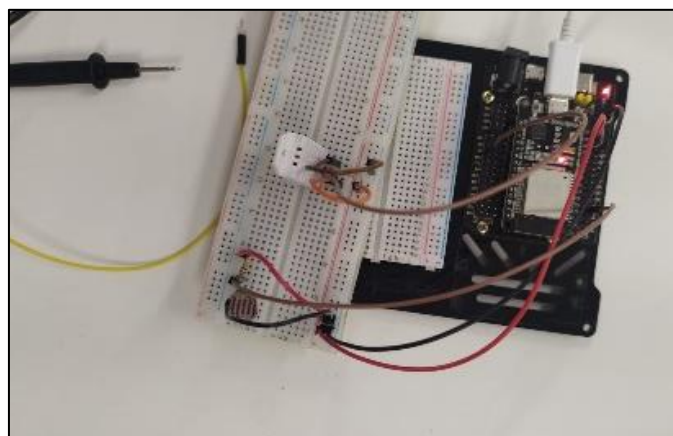


Imagem 10 – Montagem do circuito

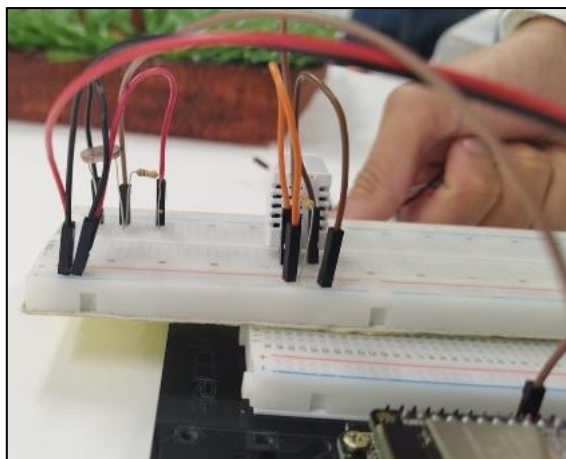


Imagem 11 – Montagem do circuito

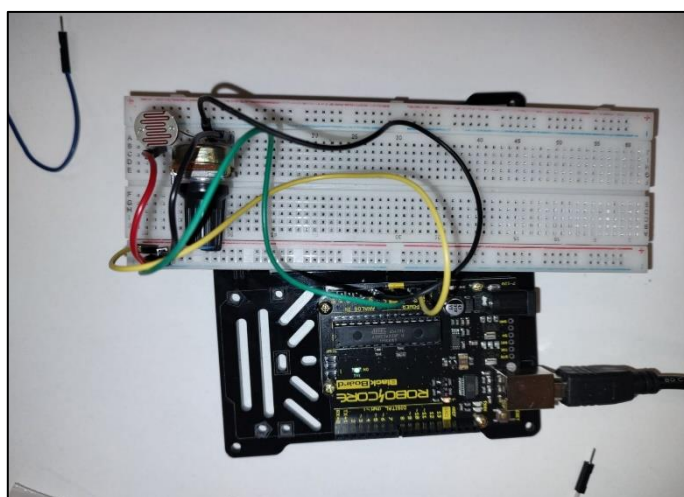


Imagem 12 – Montagem do circuito

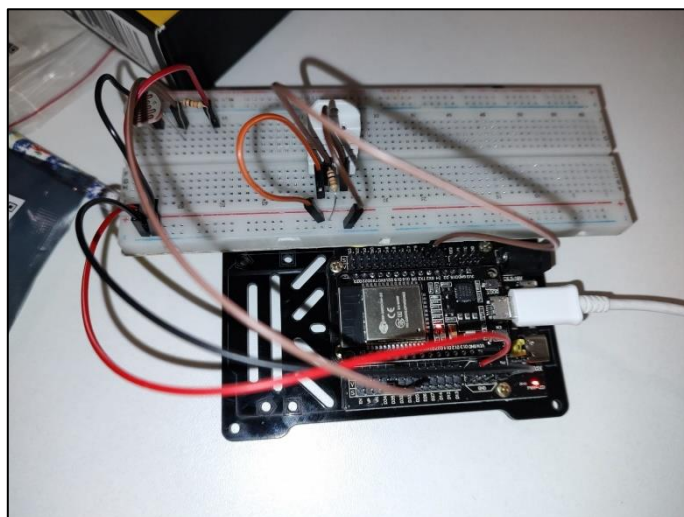
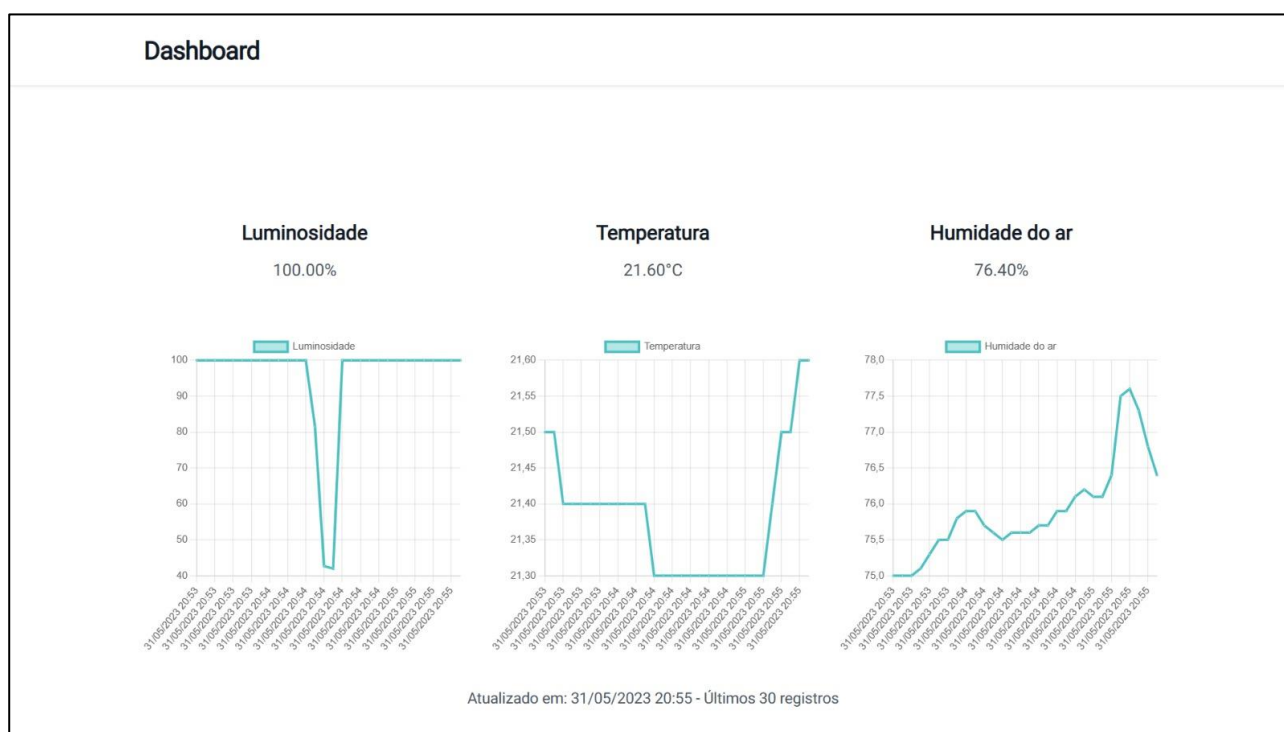


Imagem 13 – Maquete da estufa



Imagem 13 – Dashboard da estufa



9. CONCLUSÃO

Neste projeto, utilizamos o microcontrolador de código aberto ESP32, juntamente com sensores de luminosidade (LDR) e umidade/temperatura (DHT22), para criar um sistema de monitoramento de estufa. O ESP32 funcionou como o cérebro do sistema, lendo os sinais dos sensores e convertendo-os em valores digitais para processamento.

O sensor LDR mediu a luminosidade do ambiente, enquanto o sensor DHT22 forneceu informações de temperatura e umidade relativa. Com esses sensores, pudemos captar dados importantes para o controle do ambiente da estufa, permitindo ações adequadas, como acionamento automático de sistemas de iluminação ou ajuste de umidade e temperatura.

Através da programação em linguagem C e uso de bibliotecas, lemos os sensores e processamos os dados no ESP32. Os valores digitais foram enviados para um banco de dados MySQL por conexão WiFi, possibilitando o armazenamento em tempo real e a exibição dos dados em uma dashboard, mostrando luminosidade, temperatura e umidade da estufa.

Esse projeto demonstrou a versatilidade do ESP32 para o controle e interação com sensores. A combinação dos sensores LDR e DHT22 proporcionou informações valiosas para monitorar e controlar a estufa. Além disso, o uso do microcontrolador de código aberto e o acesso a bibliotecas ampliaram as possibilidades de aplicação e personalização do sistema conforme necessário.

Assim, concluímos que a integração do ESP32 com sensores e sua capacidade de processar dados analógicos e digitais têm um grande potencial em projetos de automação, controle e monitoramento em diversas áreas, incluindo agricultura, indústria e domótica.

10. REFERENCIAS

<https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>

<https://deinfo.uepg.br/~alunoso/2019/SO/ESP32/HARDWARE>

<https://lobodarobotica.com/blog/o-que-e-esp32-pra-que-serve-quando-usar>

<https://curtocircuito.com.br/blog/Categoria%20IoT/conhecendo-esp32>