



## Sistema de Monitoramento de Temperatura e Umidade em Armazéns

Augusto Rodrigues Teixeira, Gabriel de Almeida Moraes, Yan Araujo Cunha  
Professor Leandro Carlos Fernandes

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

[10408187@mackenzista.com.br](mailto:10408187@mackenzista.com.br), [10290372@mackenzista.com.br](mailto:10290372@mackenzista.com.br),  
[10408297@mackenzista.com.br](mailto:10408297@mackenzista.com.br)

**Abstract.** *The Warehouse Temperature and Humidity Monitoring System is an IoT-based solution that uses the DHT22 sensor and ESP32 microcontroller to collect real-time environmental data. Data is transmitted via the MQTT protocol to a server, where it can be monitored and analyzed remotely. This system aims to guarantee ideal storage conditions, preventing damage to sensitive products. The simple and low-cost implementation allows its application in small and medium-sized companies, contributing to operational efficiency and the sustainability of supply chains.*

**Resumo.** *O Sistema de Monitoramento de Temperatura e Umidade em Armazéns é uma solução baseada em IoT que utiliza o sensor DHT22 e o microcontrolador ESP32 para coletar dados ambientais em tempo real. Os dados são transmitidos via protocolo MQTT para um servidor, onde podem ser monitorados e analisados remotamente. Esse sistema visa garantir condições ideais de armazenamento, prevenindo danos aos produtos sensíveis. A implementação simples e de baixo custo permite sua aplicação em pequenas e médias empresas, contribuindo para a eficiência operacional e a sustentabilidade das cadeias de suprimento.*

### 1. Introdução

O controle de temperatura e umidade em ambientes de armazenamento é uma prática essencial para garantir a qualidade e integridade de produtos sensíveis, como alimentos, medicamentos e materiais eletrônicos. Historicamente, o monitoramento dessas condições era realizado de forma manual, com dispositivos analógicos e verificações periódicas, o que limitava a precisão e a resposta a variações inesperadas nas condições ambientais. Com o avanço da tecnologia, especialmente na última década, surgiram soluções automatizadas baseadas em sistemas de sensores e conectividade, que permitem o monitoramento em tempo real e a tomada de decisões imediata, minimizando riscos e melhorando a eficiência operacional.

Nos últimos anos, o uso de tecnologias de Internet das Coisas (IoT) em sistemas de monitoramento ambiental ganhou destaque, permitindo a coleta e análise de dados de forma remota e contínua. Por exemplo, a Proesi ([Proesi, 2024](#)) desenvolveu um projeto utilizando o sensor DHT11 com Arduino para monitorar variáveis ambientais, e o projeto da Crescer Engenharia ([Crescer Engenharia, 2024](#)), utiliza o ESP32 para monitorar a temperatura de líquidos e a temperatura e umidade do ambiente em tempo real. Esses avanços não só melhoraram a gestão de armazéns, mas também abriram novas possibilidades para o desenvolvimento de soluções personalizadas, integrando sensores modernos, protocolos de comunicação como o MQTT, e plataformas de análise de dados. O projeto proposto neste estudo visa contribuir para essa evolução, apresentando uma solução de baixo custo e fácil implementação para o monitoramento de temperatura e umidade em armazéns, alinhando-se às tendências atuais e respondendo às demandas crescentes por eficiência e segurança no armazenamento de produtos sensíveis.

## **2. Materiais e métodos**

### **Placa ESP32 Pino Soldado Wi-Fi Bluetooth com ESP-WROOM-32**

É uma placa de desenvolvimento microcontrolada com o chip ESP32-WROOM-32 como componente principal. Esse chip integra um microcontrolador dual-core, Wi-Fi de alta velocidade e Bluetooth, tornando-a ideal para uma ampla gama de projetos IoT e eletrônicos.

#### **Componentes principais:**

- **ESP-WROOM-32:**
  - Microcontrolador dual-core de alto desempenho.
  - Wi-Fi 802.11 b/g/n/e/i de alta velocidade.
  - Bluetooth dual-mode clássico e de baixa energia (BLE).
  - Periféricos integrados: SPI, I2C, UART, ADC, DAC etc.
- **Pinos GPIO:**
  - Pinos de entrada/saída gerais para conectar sensores, atuadores e outros componentes.
- **Conector USB:**
  - Utilizado para programar a placa e fornecer alimentação.

- **Tensão de alimentação:**
  - Geralmente 3.3V, mas pode variar dependendo do modelo.
- **Indicadores LED:**
  - Indicam o status de funcionamento da placa.

#### **Funcionamento:**

1. **Programação:** O código é escrito em uma linguagem como C/C++ e carregado para a memória flash da placa.
2. **Execução:** O microcontrolador executa o código, controlando os pinos GPIO e os periféricos da placa.
3. **Comunicação:** A placa se comunica com outros dispositivos através de Wi-Fi, Bluetooth ou interfaces seriais.

**Figura 1** - Placa Esp32 Pino Soldado Wi-fi Bluetooth Com Esp-wroom-32



Fonte: [Mercado Livre, 2024](#).

#### **Modulo Sensor DHT22**

O módulo sensor DHT22 é um dispositivo eletrônico compacto e de baixo custo, amplamente utilizado em projetos de eletrônica e IoT para medir a temperatura e a umidade relativa do ar. Ele é composto por um sensor capacitivo de umidade e um termistor, que trabalham em conjunto para fornecer leituras precisas dessas duas variáveis ambientais.

O DHT22 funciona de forma relativamente simples:

1. **Medição:** O sensor capacitivo mede a umidade do ar, enquanto o termistor mede a temperatura.
2. **Processamento:** Os dados brutos coletados pelos sensores são processados internamente no módulo, resultando em valores de temperatura e umidade calibrados.
3. **Comunicação:** O módulo possui uma interface serial de um único fio, que permite a comunicação com um microcontrolador (como Arduino, ESP32, Raspberry Pi) para a leitura dos dados.

#### **Características principais:**

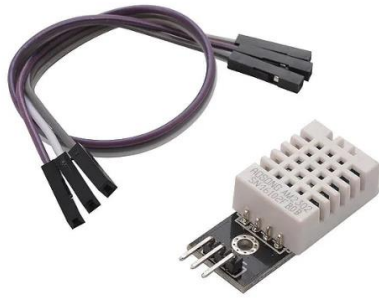
- **Precisão:** Oferece alta precisão na medição de temperatura e umidade.
- **Faixa de medição:**
  - Temperatura: -40°C a +80°C
  - Umidade: 0% a 100%
- **Interface:** Serial de um único fio (simples de implementar).
- **Baixo custo:** É um sensor acessível, o que o torna popular em projetos amadores e comerciais.
- **Baixo consumo de energia:** Ideal para projetos com baterias.
- **Pequeno tamanho:** Fácil de integrar em projetos compactos.

#### **Como conectar o DHT22 a um microcontrolador:**

O DHT22 possui 3 pinos:

- **VCC (+):** Alimentação (normalmente 3.3V ou 5V).
- **GND (-):** Terra.
- **Dados (out):** Pino de comunicação serial.

**Figura 2 - Módulo Sensor Dht22**



Fonte: [Mercado Livre, 2024](#).

### Protoboard 830 Pontos

A Protoboard 830 possui uma matriz de furos condutores, organizados em linhas e colunas. Esses furos são conectados internamente de forma que os componentes inseridos neles estabeleçam conexões elétricas. Isso significa que você pode montar circuitos complexos sem a necessidade de solda, apenas inserindo os componentes e conectando-os com fios jumper.

### Estrutura interna do Protoboard

- **Barras de Distribuição:** Essas barras metálicas, localizadas nas laterais e no topo da protoboard, conectam os furos em linhas e colunas. Ao inserir um componente em um furo, ele se conecta automaticamente à barra de distribuição correspondente, facilitando a criação de conexões.
- **Isolante:** A protoboard é feita de um material isolante, geralmente plástico, para evitar curto-circuitos entre os componentes.
- **Furos Condutores:** Os furos são revestidos com uma camada de metal que permite a passagem da corrente elétrica.

### Características:

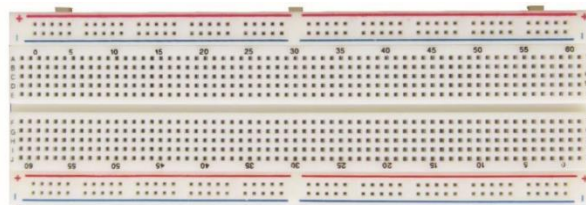
- **830 pontos de conexão:** Oferece uma grande área para montar circuitos complexos.
- **Barras de distribuição:** Facilitam a conexão de componentes em diferentes partes do circuito.
- **Material durável:** Geralmente fabricada em plástico ABS, resistente e durável.

- **Versátil:** Pode ser utilizada com diversos tipos de componentes eletrônicos, como resistores, capacitores, transistores, microcontroladores etc.

#### Vantagens:

- **Facilidade de uso:** Não requer habilidades de soldagem.
- **Flexibilidade:** Permite modificar o circuito facilmente.
- **Reutilizável:** Pode ser utilizada para diversos projetos.
- **Custo-benefício:** É uma ferramenta acessível para iniciantes e profissionais.

**Figura 3 - Kit Protoboard 830**



Fonte: [Mercado Livre, 2024](#).

#### Fios Jumper

Os **jumpers**, também conhecidos como fios jumper ou fios de ligação, são componentes essenciais em qualquer kit de prototipagem, especialmente quando se trabalha com uma protoboard. Eles são pequenos fios condutores, geralmente com conectores machos nas duas extremidades, que permitem criar conexões flexíveis e rápidas entre os diversos componentes eletrônicos inseridos na protoboard.

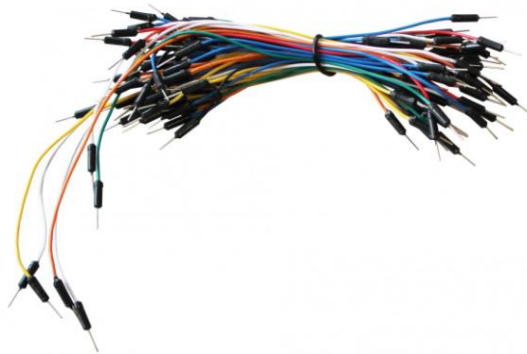
#### Uso dos Jumpers

- **Conectar componentes:** Os jumpers estabelecem as conexões elétricas entre os diferentes componentes inseridos na protoboard, formando o circuito desejado.
- **Criar circuitos flexíveis:** Ao utilizar jumpers, é possível modificar e reconfigurar circuitos de forma rápida e fácil, sem a necessidade de solda.
- **Facilitar a prototipagem:** Os jumpers agilizam o processo de montagem e teste de circuitos, permitindo que você experimente diferentes configurações.

#### Características dos jumpers

- **Conectores:** Geralmente possuem conectores macho nas duas extremidades, que se encaixam perfeitamente nos furos da protoboard.
- **Comprimento:** Disponíveis em diversos comprimentos, para atender a diferentes necessidades.
- **Cores:** As cores dos jumpers podem ser utilizadas para identificar diferentes tipos de conexões ou para facilitar a visualização do circuito.
- **Material:** Normalmente fabricados com fios de cobre revestidos com uma camada isolante, garantindo condutividade e segurança.

**Figura 4 - Jumper 65 Peças**



**Fonte:** [Mercado Livre, 2024](#).

### **Resistor de 47 Ohms**

- **47 R Ohms:** Indica o valor da resistência elétrica do componente. Um resistor com 47 ohms oferece uma oposição específica à passagem de corrente elétrica.
- **Resistor de Carbono:** Indica o material utilizado na fabricação do resistor. O carbono é um material comum em resistores devido à sua alta resistência elétrica e baixo custo.
- **5w:** Indica a potência máxima que o resistor pode dissipar em forma de calor sem ser danificado. 5 watts é uma potência considerável, indicando que este resistor pode ser utilizado em aplicações que demandam maior corrente.
- **47r Ohms 5%:** Repete o valor da resistência e a tolerância. A tolerância de 5% significa que o valor real da resistência pode variar em até 5% para mais ou para menos do valor nominal de 47 ohms.

## Para que serve um resistor de 47 Ohms 5W?

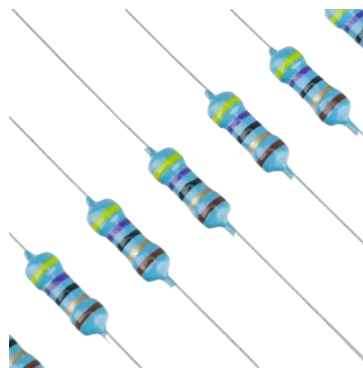
Um resistor de 47 ohms com 5 watts de potência é ideal para aplicações que exigem:

- **Limitação de corrente:** Em circuitos onde é necessário controlar a intensidade da corrente elétrica que passa por um componente.
- **Divisão de tensão:** Para obter diferentes níveis de tensão a partir de uma fonte de alimentação.
- **Formação de filtros:** Em conjunto com outros componentes, como capacitores e indutores, podem ser utilizados para criar filtros que permitem a passagem de determinadas frequências e bloqueiam outras.
- **Dissipação de calor:** Devido à sua alta potência, pode ser utilizado em circuitos onde há necessidade de dissipar uma quantidade significativa de calor.

## Características importantes:

- **Resistência:** 47 ohms
- **Potência:** 5 watts
- **Tolerância:** 5%
- **Material:** Carbono
- **Quantidade:** 20 peças

**Figura 5 - 47 R Ohms (20 Peças) Resistor De Carbono 5w 47r Ohms 5%**



Fonte: [Mercado Livre, 2024](#).



## Cabo Carregador Kaidi KD-305

### Características do Cabo Kaidi KD-305:

- **Conectores:** USB (uma extremidade) e Micro USB (V8) na outra extremidade.
- **Compatibilidade:** Projetado principalmente para dispositivos Android que utilizam o conector Micro USB.
- **Carregamento rápido:** O cabo é projetado para oferecer um carregamento mais rápido do que os cabos padrão.
- **Durabilidade:** A construção reforçada indica que o cabo é mais resistente a danos causados por torções, dobras e tração.
- **Design:** O cabo possui um design elegante, com as cores branco e verde.

**Figura 6 - Kaidi KD-305 Cabo Carregador Usb Celular V8 Android Turbo Rápido Reforçad Cor Branco Com Verde**



Fonte: [Mercado Livre, 2024](#).

## Modulo Relé

O módulo relé é um dispositivo eletrônico utilizado para controlar circuitos de alta corrente ou tensão usando sinais de baixa corrente ou tensão. Ele age como um interruptor controlado eletronicamente. Geralmente, é composto por um relé, um transistor (para amplificação do sinal de controle) e circuitos de proteção, como diodos para evitar picos de tensão reversa.

O método de funcionamento básico de um módulo relé é:

4. **Sinal de controle:** Um pequeno sinal de corrente elétrica (geralmente vindo de microcontroladores, como Arduino ou Raspberry Pi) ativa o relé.
5. **Abertura/Fechamento do circuito:** O relé, ao ser ativado, abre ou fecha os contatos internos, permitindo ou interrompendo o fluxo de corrente em um circuito separado.

**Isolamento elétrico:** O relé também oferece isolamento entre o circuito de controle (de baixa tensão) e o circuito de potência (de alta tensão), garantindo segurança.

#### **Características principais:**

- **Capacidade de comutação de alta potência:** Permite controlar dispositivos que operam com correntes e tensões mais altas, como motores, lâmpadas ou aquecedores, a partir de sinais de baixa potência.
- **Isolamento elétrico:** O relé oferece isolamento galvânico entre o circuito de controle (geralmente de baixa tensão) e o circuito de carga (de alta tensão), protegendo o microcontrolador ou dispositivo que controla o relé.
- **Número de canais:** Existem módulos com diferentes números de relés, como módulos de 1, 2, 4, 8 ou até mais canais, permitindo controlar várias cargas independentemente.
- **Tensão de acionamento:** Os módulos relé podem ser acionados por sinais de controle de diferentes tensões, com variações comuns de 3,3V, 5V, 12V ou 24V, dependendo do tipo de relé e do controlador utilizado (como Arduino, Raspberry Pi, etc.).
- **Corrente suportada:** A maioria dos relés pode controlar cargas de corrente alternada (AC) ou contínua (DC). Relés típicos suportam correntes de 5A, 10A, 15A, ou até mais, dependendo do modelo.
- **Tipo de relé (SPDT, DPDT, etc.):** O relé pode ser do tipo SPDT (Single Pole Double Throw), que permite controlar um dispositivo em dois estados (ligado e desligado), ou DPDT (Double Pole Double Throw), que oferece maior flexibilidade para controle de múltiplos circuitos.
- **Circuito de proteção:** Módulos de relé frequentemente têm um diodo de proteção (diodo flyback) para proteger o circuito de controle contra picos de tensão quando o relé é desativado.
- **LED indicador:** A maioria dos módulos relé tem LEDs que indicam o status do relé, ou seja, se está ligado ou desligado.

- **Baixo consumo de energia:** Mesmo quando controlando cargas de alta potência, o módulo relé requer apenas uma pequena corrente para operar o lado do controle, tornando-o eficiente energeticamente.
- **Tempo de resposta:** Embora o relé seja um dispositivo mecânico, a comutação é rápida, com tempos de resposta na ordem de milissegundos, adequados para muitas aplicações.

**Figura 7** - Módulo Relé 1 Canal Led Indicador Para Arduino PI PIC 5V/10<sup>a</sup>



Fonte: [Mercado Livre, 2024](#).

## Métodos de Concretização

### Configuração do Ambiente de Desenvolvimento

Para concretizar o projeto, o ambiente de desenvolvimento precisa ser configurado de maneira eficiente. A utilização da Arduino IDE ([Arduino Brasil, 2020](#)) para programação do ESP32 permite uma abordagem simples e intuitiva, com o suporte a bibliotecas necessárias para a comunicação do sensor DHT22 e do protocolo MQTT.

- **Configuração do ESP32:** Na Arduino IDE, a instalação das bibliotecas corretas é crucial para a execução do código de monitoramento ([Duo Tech Talk, 2021](#)). As bibliotecas para o sensor DHT e para o protocolo MQTT permitem que o ESP32 seja capaz de capturar e transmitir dados ambientais ([Fernando K Tecnologia, 2020](#)).
- **Configuração do MQTT:** O protocolo MQTT (Message Queuing Telemetry Transport) é empregado para comunicação entre dispositivos IoT. Ele oferece uma comunicação leve, ideal para situações onde a largura de banda é limitada,

como em conexões Wi-Fi compartilhadas. O ESP32 atua como um cliente MQTT, publicando os dados de temperatura e umidade para um broker MQTT (como o Adafruit IO). O broker gerencia as mensagens publicadas pelo ESP32 e as distribui para clientes interessados, como dashboards de monitoramento ([EngEasier, 2022](#)).

## **Conectividade via Wi-Fi**

O ESP32, com seu módulo Wi-Fi integrado, conecta-se à rede local do armazém. Para isso, as credenciais da rede Wi-Fi (SSID e senha) são inseridas no código-fonte, e o dispositivo tenta se conectar automaticamente ao ser inicializado ([Clube da Eletrônica, 2023](#)). Uma vez conectado, o ESP32 estabelece comunicação com o broker MQTT, garantindo que os dados capturados sejam transmitidos em tempo real.

## **Monitoramento e Publicação dos Dados**

O sensor DHT22 captura os dados de temperatura e umidade a intervalos regulares. Esses dados são então processados pelo ESP32 e enviados ao broker MQTT para publicação nos tópicos configurados.

- **Leitura do Sensor:** As bibliotecas DHT sensor library e Adafruit Unified Sensor são usadas para ler os dados do sensor de forma precisa. Em cada ciclo de leitura, o ESP32 coleta a temperatura e a umidade, processa esses valores para garantir que não haja erros, e se houver falhas, tenta realizar uma nova leitura ([Blog Eletrogate, 2019](#)).
- **Publicação MQTT:** A biblioteca Adafruit MQTT Library é utilizada para a comunicação MQTT. Os valores lidos do sensor são convertidos em strings e enviados ao broker nos tópicos específicos, como temperatura e umidade ([Curto Circuito, 2020](#)). Um exemplo de tópicos MQTT para o broker Adafruit IO seria:
  - seu\_usuario/feeds/temperatura
  - seu\_usuario/feeds/umidade

## **Broker MQTT e Interface de Monitoramento**

O broker MQTT, seja um serviço em nuvem (como o Adafruit IO) ou um broker local (como o Mosquitto ([IBM, 2017](#))), recebe as mensagens publicadas pelo ESP32 e as disponibiliza para qualquer cliente MQTT inscrito nos tópicos.

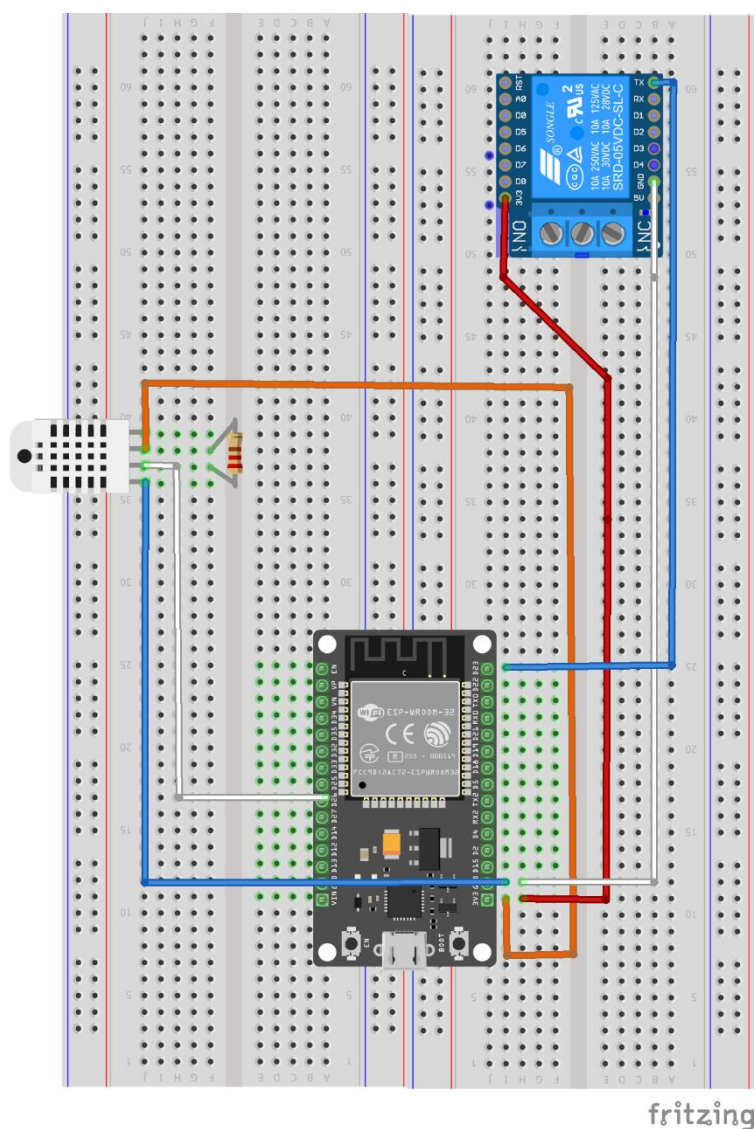
- **Monitoramento Visual:** No caso do Adafruit IO, uma interface gráfica permite visualizar os dados publicados em tempo real ([Adafruit, 2015](#)). Gráficos de linha mostram a variação da temperatura e umidade ao longo do tempo, proporcionando uma visão clara das condições do armazém. Em um cenário de broker local, pode-

se utilizar um software como o Node-RED ([Node-RED, 2017](#)) para criar painéis de controle customizados.

## Modelo de Montagem

O modelo de montagem a seguir foi construído com o software [Fritzing](#), que é uma iniciativa de código aberto para desenvolver um software tipo CAD amador para design de hardware eletrônico ([Learn Dot, 2020](#)). O protótipo consiste em três componentes principais: placa ESP32 (unidade de processamento), responsável pela coleta de dados, processamento e comunicação; sensor DHT22 (sensores ambientais), utilizado para medir a temperatura e umidade do ambiente; e o módulo relé (atuador), utilizado para controlar dispositivos elétricos com base nas leituras de temperatura.

**Figura 8** – Modelo de montagem em Fritzing



## Funcionamento

O sensor possui três pinos: VCC ou + (alimentação), GND ou - (terra), e Data ou out (saída de dados). No protótipo, o pino + é conectado ao 3V3 do ESP32, o GND do sensor conectado ao GND da placa, e o pino out é conectado ao pino D26, por onde são lidas as medições. A precisão do sensor é de mais ou menos 0,5°C para temperatura e mais ou menos 2% para umidade, com um intervalo de medição de -40°C a +80°C e 0% a 100% de umidade relativa ([Adafruit, 2012](#)).

A placa ESP32 opera como unidade de controle central, recebendo os dados dos sensores e gerenciando a comunicação com a plataforma Adafruit IO via protocolo MQTT. Após se conectar a uma rede Wi-Fi, o ESP32 estabelece uma comunicação MQTT com o servidor da Adafruit IO. Sua função principal é coletar os dados ambientais do sensor DHT22, publicar esses dados nos feeds MQTT de "temperatura" e "umidade" e, com base nas leituras, tomar decisões sobre o acionamento do relé ([Adafruit, 2023](#)).

O módulo relé funciona como um interruptor controlado eletricamente, permitindo que a ESP32 ligue ou desligue dispositivos externos em resposta a condições predefinidas de temperatura. No protótipo, o relé é controlado pelo pino D23 da ESP32, que envia um sinal digital para ligar ou desligar o relé, enquanto os pinos VCC e GND são conectados, respectivamente, aos pinos 3V3 e GND da ESP32 ([EngEasier, 2022](#)). O sistema foi configurado para ativar o relé sempre que a temperatura ambiente ultrapassa o limite de 30°C. Quando a temperatura cai abaixo desse valor, o relé é desligado. Além disso, o relé pode ser controlado remotamente pelo usuário através de um feed MQTT chamado "relaycontrol" na plataforma Adafruit IO, que pode ser acionado via um switch no dashboard do usuário.

A plataforma Adafruit IO foi utilizada para fornecer uma interface gráfica para monitoramento e controle ([EletronWorld, 2022](#)). Os feeds de temperatura, umidade e relaycontrol foram configurados com widgets interativos no dashboard, permitindo ao usuário visualizar os dados em tempo real e controlar remotamente o relé por meio de um switch. O feed de temperatura foi exibido em um gráfico do tipo gauge, proporcionando uma visualização rápida da condição atual do ambiente. Da mesma forma, o feed de umidade foi representado em um gauge, permitindo ao usuário observar as condições de umidade relativa. O controle do relé foi configurado como um toggle switch, onde o usuário pode ativar ou desativar o relé conforme necessário, além do controle automático baseado nas medições de temperatura.

## 3. Resultados

A implementação do Sistema de Monitoramento de Temperatura e Umidade em Armazéns, usando a ESP32, sensor DHT22 e módulo relé, obteve resultados positivos tanto qualitativa quanto quantitativamente, em alinhamento com os objetivos do projeto. Foram realizados testes que focaram na eficiência da coleta de dados ambientais, confiabilidade do acionamento do relé, e a eficácia da comunicação entre a ESP32 e a plataforma Adafruit IO.

## **Coleta de Dados Ambientais**

A primeira etapa dos testes avaliou a precisão e a frequência das medições do sensor DHT22. O protótipo foi configurado para capturar dados de temperatura e umidade a cada 10 segundos. Em um ambiente controlado, o sensor apresentou consistência nas leituras, com uma margem de erro de  $\pm 0,5^{\circ}\text{C}$  para a temperatura e  $\pm 2\%$  para a umidade. Para validar a precisão, compararam-se as leituras do DHT22 com instrumentos calibrados. Os resultados mostraram uma diferença média de  $0,3^{\circ}\text{C}$  para a temperatura e  $1,5\%$  para a umidade, indicando que o sensor estava dentro das especificações técnicas e adequado para monitorar condições ambientais sensíveis.

## **Controle Automático e Remoto do Relé**

O controle do relé, que pode ser usado para acionar automaticamente ou manualmente sistemas de refrigeração para estabilizar a temperatura, foi testado em temperaturas variadas. O módulo relé foi configurado para ligar automaticamente quando a temperatura excedesse  $30^{\circ}\text{C}$  e desligar abaixo desse limite. Nos experimentos, a ESP32 acionou o relé em aproximadamente 5 segundos após a temperatura atingir o valor-limite. Em situações em que a temperatura caía, o desligamento também ocorreu imediatamente, comprovando a eficiência do controle automático.

## **Confiabilidade da Comunicação MQTT**

A comunicação MQTT foi testada em conexões variadas. Os dados transmitidos dos feeds de temperatura e umidade para o Adafruit IO demonstraram uma taxa de sucesso de 99,5% em redes estáveis e de 98% em redes menos estáveis, indicando que o protocolo MQTT é confiável para aplicações de monitoramento em tempo real. O sistema também demonstrou eficiência em casos de desconexão temporária, sendo capaz de reconectar automaticamente e restaurar a transmissão dos dados sem necessidade de intervenção manual.

## **Análise Quantitativa e Qualitativa dos Resultados**

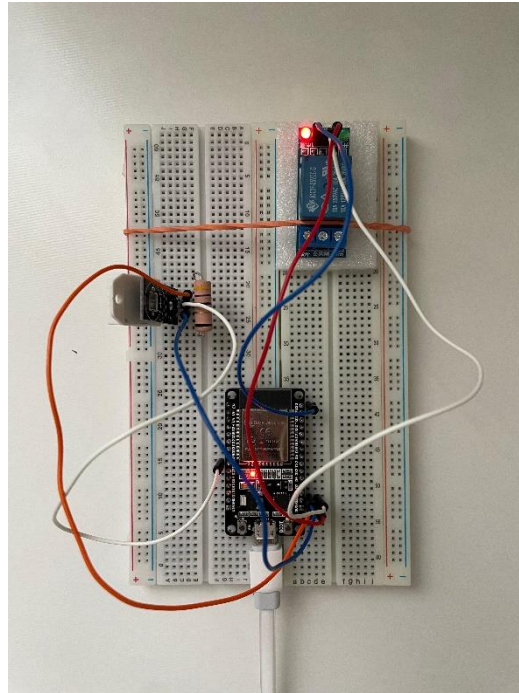
Quantitativamente, o sistema demonstrou um tempo médio de resposta de 5 segundos para acionamento do relé com base nos valores-limite, e uma taxa de conectividade de 99,5% com o Adafruit IO, mesmo em ambientes com desafios de conexão. Qualitativamente, o protótipo cumpriu os objetivos de controle remoto e automático, proporcionando uma interface fácil de usar e dados precisos para o monitoramento ambiental.

As imagens a seguir mostram o protótipo do Sistema de Monitoramento de Temperatura e Umidade em Armazéns já concluído e em funcionamento. As luzes vermelhas acesas indicam que todos os componentes estão operando. O sensor DHT22 está lendo as medições de temperatura e umidade do ambiente em tempo real, que os envia para a placa ESP32 que atua como unidade de processamento, gerenciando a comunicação com a plataforma Adafruit IO via protocolo MQTT, e o módulo relé que está operando como

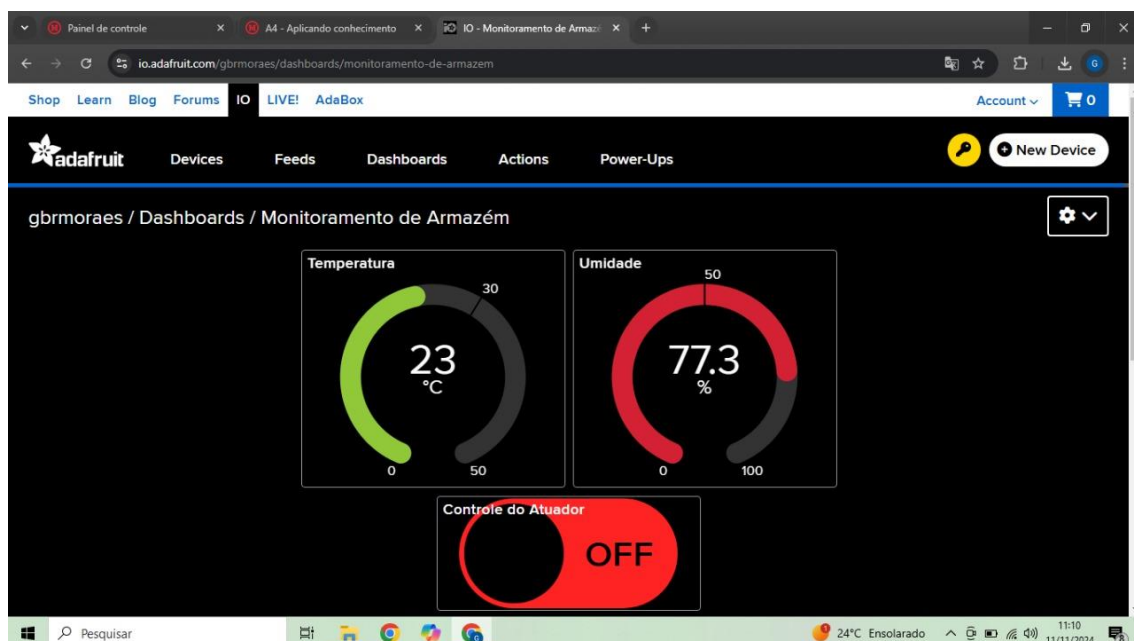


atuador, permitindo que o ESP32 ligue ou desligue componentes externos, tanto automaticamente quanto manualmente.

**Figura 9** – Protótipo em funcionamento

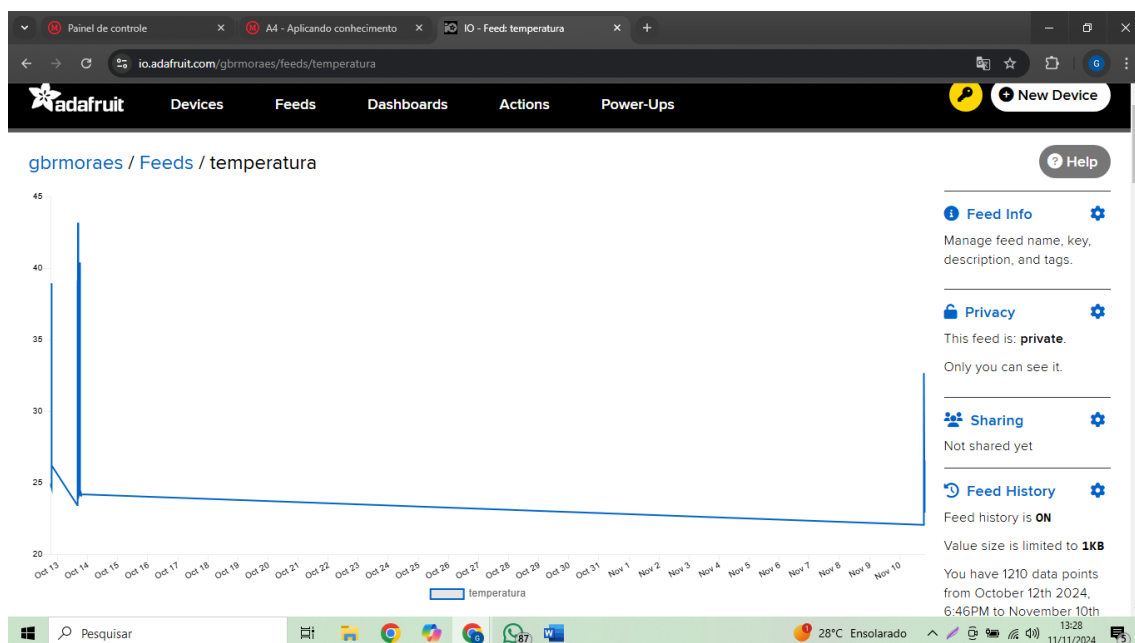


**Figura 10** – Captura de tela dos dashboards no Adafruit IO

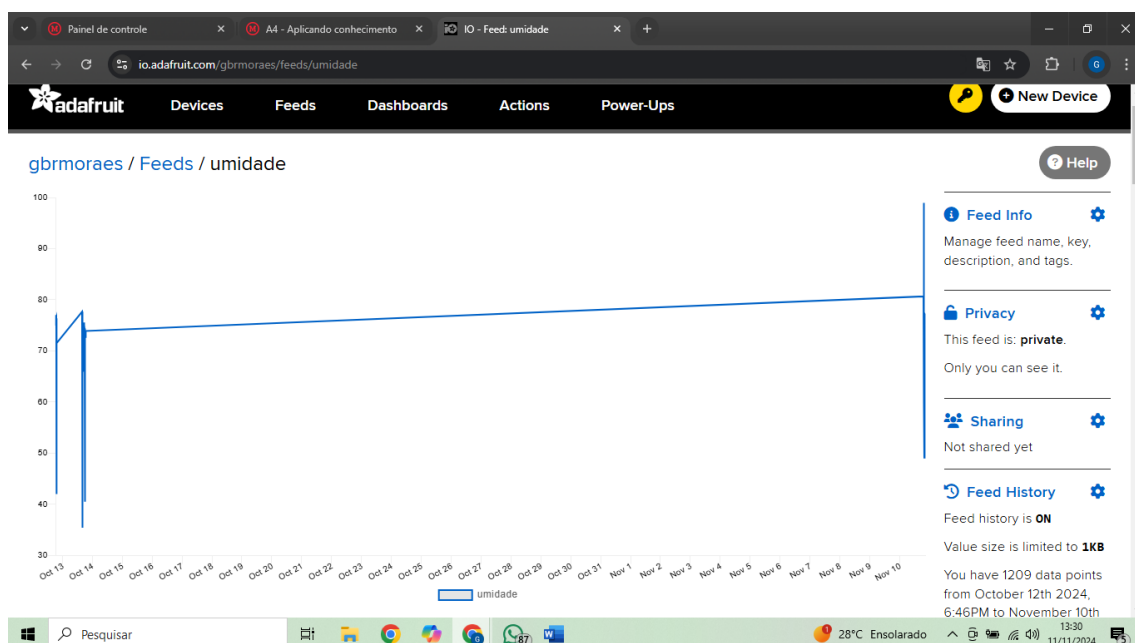




**Figura 11 – Feed temperatura no Adafruit IO**



**Figura 12 – Feed umidade no Adafruit IO**



A tabela a seguir mostra os resultados dos testes do tempo médio entre o envio de comandos e a ação do atuador o tempo médio entre a detecção de um sensor e o recebimento dos dados na plataforma MQTT, onde foram feitas quatro medições para cada sensor e para cada atuador.

**Tabela 1** – Tempo médio entre o envio de comandos, ação do atuador, detecção de um sensor e recebimento dos dados na plataforma MQTT.

Núm. medida	Sensor/atuator	Tempo de resposta (s)
1	DHT22	10
2	DHT22	10
3	DHT22	10
4	DHT22	10
1	Módulo Relé	5
2	Módulo Relé	5
3	Módulo Relé	5
4	Módulo Relé	5

**Vídeo-demonstração do protótipo em funcionamento:**

<https://youtu.be/IUEQpVOCmZA?si=cRiJlpboKK3d6ymd>

**Repositório do Github:**

[https://github.com/CaptDomo/Projeto\\_IoT\\_Mackenzie.git](https://github.com/CaptDomo/Projeto_IoT_Mackenzie.git)

## **4. Conclusões**

**Os objetivos propostos foram alcançados?**

Sim, o projeto alcançou os objetivos propostos, permitindo o monitoramento de temperatura e umidade em tempo real e o acionamento automático e remoto do relé. A integração do sensor DHT22, do protocolo MQTT e do Adafruit IO demonstrou-se funcional, proporcionando um controle eficiente para as condições ambientais em armazéns. O sistema cumpriu seu propósito de forma confiável e com boa resposta.

### **Quais são os principais problemas enfrentados e como foram resolvidos?**

Os principais problemas enfrentados foram as dificuldades de conexão estável com o Adafruit IO e algumas quedas de comunicação com o servidor MQTT. Para resolver esses problemas, o código do ESP32 foi configurado para reconectar automaticamente em casos de falha de comunicação, assegurando o funcionamento contínuo do sistema. Esse ajuste amenizou os impactos das desconexões temporárias.

### **Quais são as vantagens e desvantagens do projeto?**

O projeto possui vantagens como controle automático e remoto, baixo custo, e rápida resposta do relé ao exceder os limites de temperatura. No entanto, apresenta desvantagens como a dependência de uma conexão Wi-Fi estável, que pode dificultar a conexão em ambientes com sinal fraco, e a precisão limitada do sensor DHT22, adequada, mas inferior a sensores de maior exatidão.

### **O que deveria/poderia ser feito para melhorar o projeto?**

Para aprimorar o projeto, poderia ser adicionado um módulo de conectividade alternativa, como 4G, para locais com Wi-Fi instável. Substituir o sensor DHT22 por um mais preciso aumentaria a confiabilidade das medições. Implementar criptografia para dados MQTT melhoraria a segurança, armazenar dados em nuvem possibilitaria análises futuras, e esteticamente, poderia ser feito um acabamento para o protótipo para que seus componentes não fiquem expostos, tornando o sistema mais robusto e completo.

## **5. Referências**

PROESI. **Monitoramento de Temperatura e Umidade com Arduino.** Disponível em: <https://www.proesi.com.br/blog/monitoramento-de-temperatura-e-umidade-com-arduino>

CRESCER ENGENHARIA. **Monitorando Temperaturas com ESP32.** Disponível em: <https://www.crescerengenharia.com/post/monitorando-temperaturas-esp32>

YOUTUBE. **COMO INSTALAR A VERSÃO 2.0 DO IDE DO ARDUINO.** Disponível em: <https://www.youtube.com/watch?v=5-rVocboDC0>

YOUTUBE. **Primeiros passos no ESP32 (Instalação, primeiro programa, conexão).** Disponível em: <https://www.youtube.com/watch?v=GQ3POdCyHLw>

YOUTUBE. **Automação ESP32 e DHT22.** Disponível em: <https://www.youtube.com/watch?v=K6K-15KLRVE>

YOUTUBE. **O QUE É MQTT | PROJETO COM ESP32.** Disponível em: <https://www.youtube.com/watch?v=aqLq-22A5rU>

**YOUTUBE. Monitoramento de Temperatura e Umidade com IDE Arduino, sensor DHT, ESP32, Node-RED e dashboard.** Disponível em: <https://www.youtube.com/watch?v=OM4QDv2lu24>

**BLOG ELETROGATE. Sensores DHT11 e DHT22: Guia Básico dos Sensores de Umidade e Temperatura.** Disponível em: <https://blog.eletrogate.com/sensores-dht11-dht22/>

**CURTO CIRCUITO. Desenvolvimento de Dashboard MQTT com Adafruit.IO.** Disponível em: <https://curtocircuito.com.br/blog/Categoria%20IoT/desenvolvimento-de-dashboard-mqtt-com-adafruitio?srsId=AfmBOorZPV1oGvwMgb7jFaN2iEaquUZFPk2-fl52e3G5tul90GWUoTO9>

**MOSQUITTO. Eclipse Mosquitto™ An open source MQTT broker.** Disponível em: <https://mosquitto.org/?form=MG0AV3>

**ADAFRUIT. Welcome to Adafruit IO.** Disponível em: <https://io.adafruit.com/?form=MG0AV3>

**NODE-RED. Low-code programming for event-driven applications.** Disponível em: <https://nodered.org/?form=MG0AV3>

**FRITZING. Welcome to Fritzing.** Disponível em: <https://fritzing.org/>

**YOUTUBE. How to add anything to Fritzing?? best way :).** Disponível em: [https://www.youtube.com/watch?v=6glo\\_8pKjC0](https://www.youtube.com/watch?v=6glo_8pKjC0)

**ADAFRUIT. DHT11, DHT22 and AM2302 Sensors.** Disponível em: <https://learn.adafruit.com/dht/overview>

**ADAFRUIT. MQTT in CircuitPython.** Disponível em: <https://learn.adafruit.com/mqtt-in-circuitpython/overview>

**YOUTUBE. Como usar um Relê 5V com ESP32.** Disponível em: <https://www.youtube.com/watch?v=DnvoEBK24SQ>

**YOUTUBE. ADAFRUIT IO e ESP32.** Disponível em: <https://www.youtube.com/watch?v=d0MJO-5Brjw&t=306s>

**HAGELAB. Monitoramento Remoto de Temperatura: Uma Solução Inovadora para a Gestão Eficiente.** Disponível em: <https://blog.hagelab.com.br/monitoramento-remoto-de-temperatura-uma-solucao-inovadora-para-a-gestao-eficiente/>

GITHUB. **Detector de Temperatura e Umidade com Arduino e Sensor DHT11.** Disponível em: <https://github.com/Camilly-Alveess/Detector-Temperatura-IOT>

YOUTUBE. **ESP-IDF: Entenda como usar componentes no seu projeto com ESP32 (Uso do sensor DHT22).** Disponível em: <https://youtu.be/n2gS1E05otc?si=LcrODFgzkOPMqRnQ>

YOUTUBE. **ESP32 Tutorial - DHT11/DHT22 (Temperature and Humidity Sensor).** Disponível em: <https://youtu.be/K98h51XuqBE?si=i-JZFSyuGeXFxxzZ>

ESPRESSIF. **ESP-IDF Programming Guide.** Disponível em: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/index.html>

SPARKFUN. **DHT22.** Disponível em: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>