

CONTROLE E MONITORAMENTO DE TEMPERATURA PARA PRODUTOS TERMOLÁBEIS

Matheus Da Silva Schmidt¹, Valter Blauth Junior²

Resumo: Produtos sensíveis à temperatura devem ter monitoramento constante para evitar perdas na eficácia e conseqüentemente o descarte destes materiais tão valiosos à saúde humana. Esta pesquisa teve como objetivo aplicar conceitos de Internet das Coisas na integração de um refrigerador com um servidor (computador central que processa e gerencia os dados) e interface para controle e monitoramento da temperatura interna. A partir de valores estabelecidos pelo usuário, o refrigerador automatizado controla a temperatura e extrai os dados para a emissão de um relatório. No controle de temperatura, foi utilizado lógica Fuzzy para lidar com conjuntos criados dinamicamente. Os resultados mostraram um correto funcionamento em controle de temperaturas entre 10 °C e 30 °C com uma taxa de erro absoluta de 1,5%.

Palavras-chave: Temperatura; Sistemas de controle; Lógica Fuzzy;

ABSTRACT: Temperature-sensitive products require constant monitoring to prevent loss of efficacy and the consequent disposal of these valuable materials for human health. This research aimed to apply Internet of Things (IoT) concepts to integrate a refrigerator with a server (a central computer that processes and manages data) and an interface for controlling and monitoring the internal temperature. Based on user-defined values, the automated refrigerator controls the temperature and extracts data for report generation. Fuzzy logic was used for temperature control to handle dynamically created sets. The results showed correct functioning in controlling temperatures between 10 °C and 30 °C, with an absolute error rate of 1.5%.

Keywords: Temperature; Control systems; Fuzzy logic.

¹ Curso de Ciência da Computação Universidade do Extremo Sul Catarinense (Unesc), Criciúma - Santa Catarina - Brasil. matheus.schmidt@outlook.com.

² Orientador, Curso de Ciência da Computação, Grupo de Pesquisa em Inteligência Artificial Aplicada, Universidade do Extremo Sul Catarinense (Unesc), Criciúma - Santa Catarina - Brasil. valterblauth@unesc.net.

1. INTRODUÇÃO

As vacinas são um dos principais instrumentos da saúde pública, contribuindo para a prevenção de doenças infecciosas e a proteção da saúde da população. Segundo Britto (2021):

A importância da vacinação não está somente na proteção individual, mas porque ela evita a propagação em massa de doenças que podem levar à morte ou a sequelas graves, comprometendo a qualidade de vida e saúde das pessoas vitimizadas.

Um dos principais obstáculos no armazenamento de vacinas reside na natureza delicada dos componentes biológicos presentes nas formulações. As vacinas são produtos termolábeis, o que significa que devem ser armazenadas em temperaturas específicas para manter sua eficácia. Segundo Inframetro (2021) em Minas Gerais, 12 mil doses de vacina contra a covid-19 foram perdidas por razões como falta de energia elétrica, falhas em equipamentos e no transporte.

O presente trabalho visa propor um sistema de monitoramento e controle de temperatura para armazenamento de vacinas baseado em microcontroladores. O sistema utiliza um sensor de temperatura para monitorar a temperatura interna do recipiente térmico, e um controlador de temperatura baseado em microcontrolador para manter a temperatura interna dentro da faixa especificada. O sistema também inclui um sistema de monitoramento remoto que permite visualizar os dados de temperatura.

O objetivo do estudo é avaliar a viabilidade e a eficiência do sistema proposto. Para isso, serão realizados testes para avaliar a eficiência do controlador de temperatura e também a funcionalidade do sistema de monitoramento remoto integrado.

Essa abordagem oferece uma solução eficiente e confiável em comparação com métodos tradicionais de controle manual, reduzindo o risco de perdas e garantindo a qualidade das vacinas transportadas.

Além disso, a adoção desta tecnologia traz vantagens significativas. A capacidade de monitorar e controlar remotamente a temperatura interna da caixa térmica em tempo real proporciona maior agilidade na detecção de variações indesejáveis, permitindo a tomada de medidas imediatas para corrigir qualquer desvio.

Isso contribui para a otimização da cadeia de frio, evitando danos aos produtos e garantindo a eficácia das vacinas no momento da administração (NOLETO, 2022).

Os objetivos específicos deste trabalho consistem em investigar as principais tecnologias para monitoramento de temperatura e estabelecer um dispositivo para ser utilizado no projeto; avaliar as técnicas e tecnologias utilizadas atualmente para o controle de temperatura com o intuito de apontar suas vantagens e limitações para uso neste cenário; gerar um protótipo viável e funcional para armazenamento de produtos sensíveis à temperatura; e, por fim, extrair os dados gerados de forma fácil e clara para posterior avaliação da viabilidade do protótipo.

2. TRABALHOS CORRELATOS

Na literatura, existem trabalhos diversos relacionados ao controle e monitoramento de produtos sensíveis à temperatura que permeiam o desenvolvimento desta pesquisa.

O projeto de Pinto (2023), consiste em um hardware que monitora a temperatura das doses da vacina contra Covid-19 durante o armazenamento. O hardware inclui o sensor DS18B20, microcontrolador ESP32, módulo GPS NEO-6M e antena LoRa na frequência de 915 MHz. O software utilizado é a solução ProLoT, que oferece serviços em nuvem e interface do sistema.

Como não há um sistema de controle, os testes foram divididos em três etapas: medição de temperatura, recebimento de dados GPS e teste de alcance da transmissão LoRa. O teste de temperatura durou 43 minutos para refrigerar de +2°C a +8°C, com tempo máximo de armazenamento de 27 horas e 56 minutos. O teste de GPS assegurou a localização da caixa térmica durante o transporte, enquanto o teste de alcance LoRa estimou um alcance de 12 km. Concluiu-se que a tecnologia LoRaWAN da ATC em Manaus permite o funcionamento pleno do monitoramento de temperatura dentro dos valores obtidos.

A pesquisa de Araújo (2023), teve como objetivo construir um dispositivo de monitoramento que captura dados do sensor de temperatura BMP280 e do GPS, armazenando-os em um cartão SD e enviando alertas SMS de temperaturas não ideais. O dispositivo consiste em um módulo GSM/GPRS SIM800L, placa ESP-WROOM-32, módulo GPS, módulo de cartão SD e sensor BMP280.

Os testes envolveram a medição direta da temperatura com um multímetro digital com termopar e análise dos dados enviados pelo ESP via GSM em dois cenários de teste: um estático e outro com trajeto de 8 km.

Os resultados mostraram um erro absoluto máximo de temperatura de 2,32 °C e 2,06 °C nos dois cenários, respectivamente. Os módulos funcionaram conforme o esperado, embora com atrasos ao entrar em pleno funcionamento em ambientes internos de edifícios devido à perda de potência do sinal e antenas de baixa sensibilidade.

O protótipo desenvolvido por Ferreira et al (2023), tem como objetivo o estudo e desenvolvimento de um monitoramento de baixo custo e fácil acesso, utilizando o NodeMCU e um sensor de temperatura. O trabalho adota uma abordagem qualitativa experimental, utilizando componentes eletrônicos para coleta e processamento de dados.

O hardware inclui um microcontrolador ESP8266, um display LCD e um sensor de temperatura MAX6675, enquanto o software utiliza a aplicação de nuvem IoT Thingspeak, integrada à plataforma X para envio de alertas de temperatura. O protótipo permite identificar mudanças na temperatura e enviar notificações, oferecendo uma solução econômica.

O artigo elaborado Lima et al (2021), tem como objetivo construir um protótipo de monitoramento de temperatura para o armazenamento de vacinas, utilizando LED verde e vermelho, e um *buzzer* para indicar se a temperatura está acima ou abaixo do especificado.

O protótipo utiliza a plataforma Arduino IDE para programação do código-fonte e é composto por um sensor de temperatura DS18B20, plataforma de programação Arduino UNO R3 e um LCD para notificações. Embora simples, o projeto monitora com precisão o conteúdo da caixa de isopor, gerando notificações visuais e sonoras em caso de necessidade de correção da temperatura.

3. MATERIAIS E MÉTODOS

A presente pesquisa possui uma abordagem qualitativa aplicada, de base tecnológica, voltando-se ao uso de técnicas de controle para automação em um cenário de armazenamento de produtos termolábeis. Com a prerrogativa de construir um protótipo eficiente e seguro para o correto armazenamento de produtos sensíveis à temperatura, o desenvolvimento desta aplicação é idealizado para manter a temperatura de uma caixa térmica através de um sensor de temperatura e uma pastilha de Peltier para o correto resfriamento.

O projeto passou por diferentes etapas para sua construção, tendo em vista que o mesmo abrange áreas diversas da eletrônica e computação. Podemos separar entre a montagem do hardware para atender as necessidades físicas de controle de temperatura do projeto e dos componentes eletrônicos, o desenvolvimento do servidor responsável pelo processamento da lógica de controle de temperatura e armazenamento dos registros e a construção da interface para visualização dos dados e interação entre o usuário e sistema.

3.1. MONTAGEM DO HARDWARE

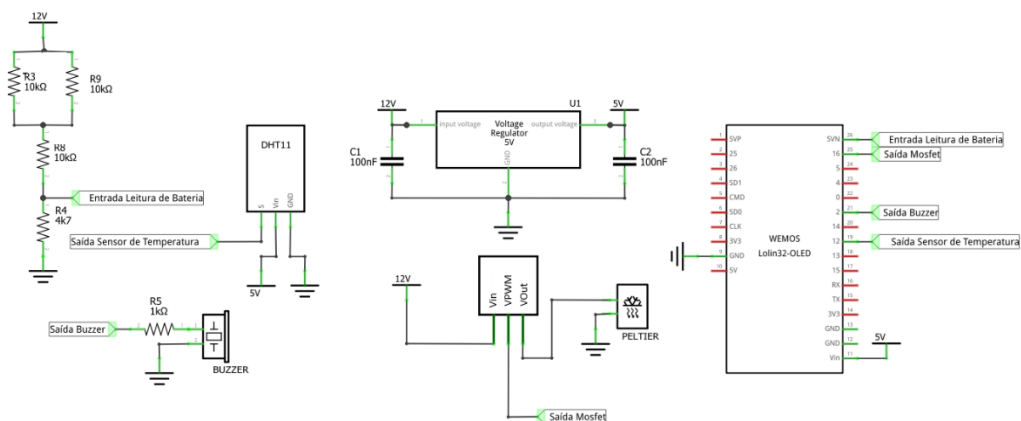
O hardware do projeto é fundamental para o correto funcionamento do sistema como um todo. Os componentes aplicados foram cuidadosamente selecionados para garantir o correto funcionamento do mesmo.

3.1.1. Montagem do circuito

O circuito é composto primordialmente de uma fonte chaveada do tipo colmeia que retifica e baixa a tensão de 220VCA para 12VDC, um regulador de tensão 7805, um módulo para conversão PWM 3.3V para 12V, uma placa controladora Wemos D1 Mini Nodemcu Esp32 com chip Wi-Fi e Bluetooth nativos e display OLED, um sensor de temperatura DHT11, uma pastilha peltier Tec1-12706, três resistores de 10k ohms (1 Watt), um resistor de 4.7k ohms (1 Watt), um resistor de 1k ohms (1 Watt), um cooler 12V e um mini refrigerador.

A disposição dos componentes pode ser compreendida no esquemático eletrônico abaixo:

Figura 1 – Esquemático eletrônico



Fonte: Do autor (2024).

A fonte colmeia recebe a tensão vinda da rede no formato 220 V e corrente alternada, fazendo a retificação para 220 V corrente contínua e regulando para 12 V. O 12 V fornecido pela fonte chaveada será novamente regulado, desta vez para 5 V, alimentando a placa ESP32. A tensão entregue pela fonte também serve de alimentação do módulo PWM, onde será futuramente convertida para alimentar a pastilha de Peltier e o cooler.

A placa ESP32 foi fixada em uma placa de prototipagem, que facilita o manuseio dos componentes eletrônicos pois possui furos e conexões internas, dispensando a necessidade de solda.

O ESP32 é responsável por receber os valores do sensor de temperatura e umidade DHT11, e envia o valor de PWM para o módulo correspondente, que recebe o 12 V fornecido pela fonte chaveada e divide por pulsos conforme recebido pelo ESP32, enviando o valor de 12 V modularizado para o cooler e pastilha de peltier.

O mini refrigerador utilizado tem a estrutura térmica necessária para efetuar os testes idealizados neste projeto, embora já seja um refrigerador funcional foram retiradas todas as peças referentes a refrigeração e ao controle para construir novamente estes sistemas baseados nos requisitos do projeto.

Figura 2 - Refrigerador

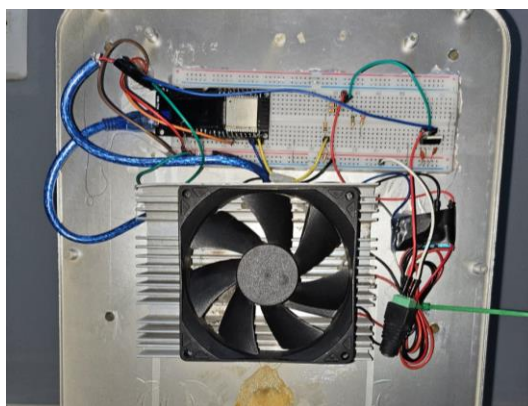


Fonte: Do autor (2024).

Embora não seja a premissa principal do projeto, o mesmo possui suporte para funcionamento a bateria, tendo acoplado um circuito de leitura do percentual de carga da bateria, que funciona baseado no princípio de divisão de tensão para converter a tensão da bateria, que pode variar de 8 V até 14,8 V, para uma tensão correspondente de 0 a 3.3V e um buzzer para indicação sonora do percentual de carga da bateria, que será melhor desenvolvido no tópico da escrita do código do microcontrolador.

A montagem final e fixação do sistema no refrigerador pode ser visualizada na figura abaixo:

Figura 3 – Montagem final



Fonte: Do autor (2024).

3.2. ESCRITA DO CODIGO DO MICROCONTROLADOR

No desenvolvimento do software foi usado o Arduino IDE em sua versão 2.3.2, com a linguagem de programação C++. Para auxiliar no desenvolvimento da aplicação foram implementadas sete bibliotecas, são elas: ArduinoHTTPClient versão 0.6, Arduino_JSON versão 0.2, Adafruit GFX Library versão 1.11.9, Adafruit SSD1306 versão 2.5.10, DHT sensor library versão 1.4.6. Além destas foram utilizadas também as bibliotecas WiFi.h e Wire.h, fornecidas pela Arduino IDE e sem controle de versão aparente. O restante das bibliotecas foi fornecido diretamente na aba “extensões” da Arduino IDE tendo em vista que são de terceiros.

A biblioteca Wire.h é utilizada para estabelecer comunicação I2C entre o esp32 e o display OLED acoplado. Usou-se também a biblioteca Adafruit_SSD1306 em conjunto com a Adafruit_GFX para escrita no display e atualização do mesmo. A biblioteca WiFi.h tem a função de estabelecer conexão a uma rede WiFi, e a ArduinoHTTPClient para conectar-se ao servidor e enviar/receber dados a partir de um pacote JSON, configurado pela biblioteca Arduino_JSON. Por fim, a biblioteca DHT tem a função de estabelecer o método de leitura e umidade do sensor DHT11 pelo ESP32.

O código do sistema pode ser separado em duas etapas, são elas: Configuração (setup) e Execução (loop). Durante a etapa de configuração é feita a inicialização de todos os pinos utilizados no ESP32 e sua respectiva função (entrada ou saída), inicialização da leitura do sensor DHT e do display OLED. Ainda na etapa de configuração é estabelecida a conexão a uma rede Wi-Fi e definição do canal do PWM a ser utilizado, sua frequência e a resolução de funcionamento.

A etapa de execução é voltada ao envio e recebimento dos dados e também a exibição de informações no display OLED. O loop começa verificando se houve uma conexão WiFi, que é primordial para o correto funcionamento do sistema, caso não esteja conectado é feito mais dez tentativas num intervalo de um segundo cada para garantir que haja conexão. Passado este período, é assumido que não haverá conexão e é estabelecido o valor de 102 na variável PWM, para manter a temperatura interna.

Caso haja conexão após as instruções do código acima, é chamado a função getPwmValue(), que busca o valor de temperatura e o concatena na URL para enviar numa solicitação do tipo GET ao servidor, que retorna o valor numérico correspondente entre 0 e 255. Caso não haja conexão com o servidor, o valor é estabelecido em 102 para manter temperatura interna.

Após o retorno da função descrita no parágrafo anterior, é definido que o valor recebido será enviado no canal zero, que foi estabelecido na etapa de configuração. Também na execução é feito a leitura do valor de tensão recebido pela divisão de tensão, caso este valor seja maior que zero (para verificar se uma bateria esteja conectada, já que o sistema também funciona pela rede), é efetuado a conversão do valor de tensão em porcentagem, e conforme o valor de porcentagem é emitido um sinal sonoro através do buzzer para indicar se a bateria precisa ser recarregada.

Por fim, é chamado a função `displayData()`, que atualiza no display OLED os valores de temperatura lida pelo sensor, PWM enviado para a pastilha de Peltier e cooler, e status da conexão Wi-Fi. Caso o funcionamento do sistema seja por meio de bateria (identificado conforme descrição no parágrafo anterior) ele alterna a exibição do status do Wi-Fi e porcentagem de bateria.

Figura 4 – Exibição dos dados no display OLED



Fonte: Do autor (2024).

3.3. ESCRITA DO CÓDIGO DO SERVIDOR

O servidor foi desenvolvido em linguagem TypeScript (TS) através do framework NestJS, no ambiente de programação Visual Studio Code, versão 1.86.1. Para executar o código JS em backend foi usado o Node.js, versão 21.6.1 e foram usadas as bibliotecas *typeorm*, *dotenv*, *moment*, *cors*, *class-validator* e *nest*. As bibliotecas foram disponibilizadas de forma gratuita pelo gerenciador de pacotes Node Package Manager (NPM), versão 10.2.4.

A biblioteca Dotenv permitiu que fossem configuradas algumas variáveis de ambiente para facilitar o desenvolvimento do código. A Moment disponibilizou

várias funções que permitiram manipular variáveis vinculadas as datas. A conexão e a inserção de registros no banco de dados foram feitas pela TypeORM, a biblioteca CORS é utilizada para controlar o acesso a recursos entre diferentes origens, a class-validator ajuda na validação de dados usando decorators em classes e, por fim, a biblioteca nest possibilita o uso do framework no backend.

O módulo servidor possui duas principais responsabilidades, que são elas: receber o valor de temperatura lido pelo hardware, processar o dado através de lógica fuzzy e responder com um valor PWM correspondente; e gerenciar o armazenamento e envio dos dados de configuração e registros para posterior análise.

3.3.1. Lógica de controle

Para fazer o controle de temperatura, o servidor é acionado a partir de uma URL que é configurada pela concatenação de: “http://”, a porta utilizada pelo servidor, “/fuzzy/”, e o valor de temperatura lido.

A partir desta URL, o servidor acessa o serviço de controle de temperatura entendendo o valor de temperatura atual. Logo após a inicialização, é definido uma classe interna chamada “FuzzySet”, que molda um conjunto fuzzy. Cada conjunto é caracterizado por uma função de pertinência que atribui a cada elemento um grau de pertinência entre 0 e 1. As propriedades desta classe são: name, min, midMin, midMax e max, definindo que cada conjunto terá uma nomenclatura e quatro valores para representa-lo.

Logo após a definição da classe interna, é trazido do banco de dados duas variáveis, chamadas de “temperature” e “offset”. Estas variáveis correspondem ao valor ideal definido pelo usuário e sua margem de erro de operação. Estes valores moldam os conjuntos de temperatura que serão definidos de maneira dinâmica futuramente. A configuração dos conjuntos fuzzy de temperatura é feita criando cinco conjuntos fuzzy (“very low”, “low”, “medium”, “hot”, “very hot”), cada um com limites específicos baseados no valor de temperatura ideal e o offset calculado, permitindo que a faixa de temperatura seja dividida em intervalos para melhor controle.

Em seguida, os conjuntos fuzzy de PWM são definidos de maneira estática, representando as diferentes velocidades do ventilador (“fan off”, “low speed”, “medium speed”, “fast speed”, “fan on max”). Estes conjuntos são então utilizados para mapear as temperaturas correspondentes às velocidades do ventilador. Quando o valor de

temperatura lido é recebido, ele é comparado com os conjuntos fuzzy de temperatura para identificar em qual conjunto ele se encaixa.

Uma vez identificado o conjunto correspondente, o conjunto de PWM equivalente é selecionado, e o valor de PWM é calculado como a média aritmética dos valores dos limites do conjunto. Este valor de PWM é então arredondado e retornado como a resposta final do método, que determina a velocidade do ventilador necessária para manter a temperatura dentro da faixa desejada.

3.3.2. Gerenciamento dos registros e configuração

O servidor além de controlar a temperatura do sistema, também tem a função de armazenar a configuração de temperatura ideal e offset, e guardar e armazenar os registros para análise e para exibição na interface do sistema.

Os valores de temperatura ideal e offset são passados por uma URL composta pela concatenação de: "http://", a porta utilizada pelo servidor, "/configs/1". No corpo da requisição é passado um objeto JSON chamado updateConfigDTO com os atributos temperature e offset, que se configura como um objeto com as informações necessárias para atualizar o registro de configuração. Este registro é acessado pela função de controle de temperatura e também pela função que armazena os registros.

Já os registros de temperatura são enviados pelo hardware, e acessam a URL que se constitui por: "http://", a porta utilizada pelo servidor, "/registries". No corpo da requisição é passado um objeto JSON chamado createRegistryDTO com os atributos temperature, humidity e createdAt. Estas variáveis correspondem a temperatura e umidade no momento da leitura dos mesmos, bem como a data e hora deste registro. Esta função tem a única responsabilidade de armazenar o registro no banco de dados, para futuras consultas pela interface.

Figura 5 – Objetos de envio para o servidor

```
{
  "createRegistryDto": {
    "temperature": 23,
    "humidity": 5,
    "created_at": "2024-04-21T16:29:48.217Z"
  }
}
{
  "updateConfigDto": {
    "temperature": 15,
    "offset": 2
  }
}
```

Fonte: Do autor (2024).

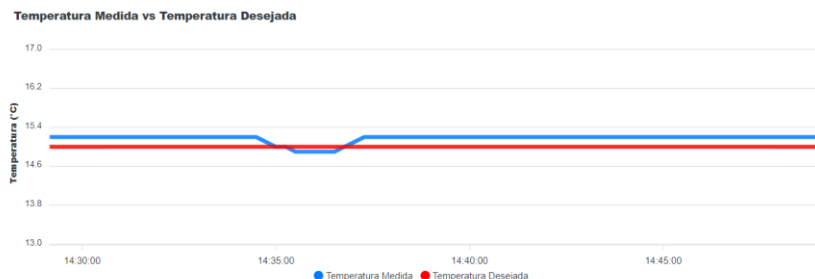
3.4. ESCRITA DO CÓDIGO DA INTERFACE

Para uma melhor interação do usuário com o sistema de controle de temperatura, foi desenvolvido uma interface em linguagem TypeScript (TS) através do framework Vue3, no ambiente de programação Visual Studio Code, versão 1.86.1. Para auxiliar no desenvolvimento foram utilizadas as bibliotecas apexcharts na versão 3.49.1, axios na versão 1.6.8, vue-datepicker na versão 8.6.0 e a biblioteca vue na versão 3.4.21.

A biblioteca vue permite o funcionamento do framework no projeto, datepicker cria um componente de seleção de datas, axios faz as requisições HTTP no servidor e a biblioteca apexcharts cria e renderiza gráficos com atualização contínua para exibição na tela.

A tela possui uma única página, com título e subtítulo identificando que a mesma se trata do controle e monitoramento da temperatura interna do refrigerador, no canto superior direito possuem dois botões, intitulados “Editar temperatura ideal” e “Emissão de relatórios”.

Figura 6 – Interface para monitoramento



© Projeto desenvolvido para a disciplina de TCC no curso de Ciência da Computação da UNESC.

Fonte: Do autor (2024).

O botão “Editar temperatura ideal” abre um pequeno formulário onde pode ser inserido o valor da temperatura ideal e a margem de erro, ambos em graus celsius.

Logo ao lado, o botão “Emissão de relatórios” abre um componente do tipo datepicker, um calendário que possibilita ao usuário escolher duas datas. Ao clicar no botão Gerar, é feito o download de um arquivo do tipo .xlsx com os registros de data, temperatura, umidade, e temperatura ideal dos registros no intervalo solicitado pelo usuário.

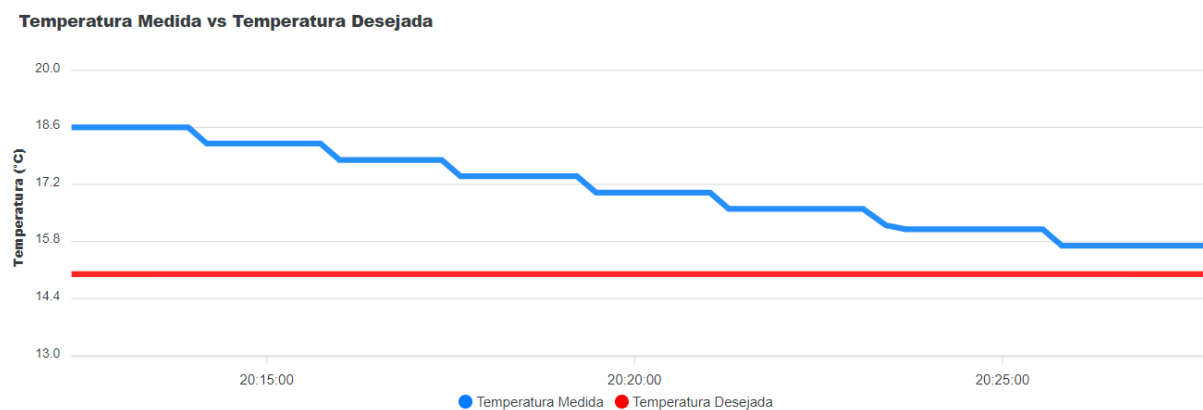
Centralizado na tela, está o gráfico intitulado “Temperatura Medida x Temperatura Desejada” que traça duas linhas trazendo os valores de temperatura através de uma chamada GET dos últimos 50 registros, e o valor de temperatura ideal configurado.

No eixo das ordenadas, estão os valores de temperatura, com uma margem entre o maior e menor valor de temperatura recebidos, e no eixo das abscissas se encontra a data dos registros, sendo que se os valores correspondem a todos no mesmo dia, se informa apenas horas e minutos dos registros.

4. RESULTADOS E DISCUSSÕES

O sistema por completo demonstrou ser eficaz quanto ao monitoramento e controle da temperatura. Hardware, servidor e interface interagiram dentro dos aspectos especificados e foi possível não só controlar a temperatura e deixá-la dentro da margem de erro como também extrair os dados para análise do comportamento do sistema em diferentes situações.

Figura 7 – Queda de temperatura visualizada na Interface



Fonte: Do autor (2024).

A ideia inicial era de se fazer o controle via lógica Fuzzy diretamente no ESP32, deixando o servidor apenas com a função de gerenciar os registros e a configuração de temperatura ideal. Porém a biblioteca eFLL encontrada para fazer este processo não se comportava corretamente, e não havia documentação suficiente para entender o seu funcionamento e efetuar a correção dos problemas.

Outro problema encontrado foi que, com a biblioteca eFLL, a memória do ESP estava em 95% de sua capacidade apenas com as variáveis globais de funcionamento, não tendo margem para armazenar nenhum registro em caso de perda de conexão com o servidor. Alternativamente foi passado a lógica de controle para o servidor, que por ter sua base em JavaScript, tem maior documentação disponível para elaboração e construção da função desejada.

Um ponto a ser considerado é a interferência da temperatura externa no projeto, embora a estrutura física do refrigerador retenha de maneira eficaz o frio em sua parte interna, não há vedações que garantem total isolamento da temperatura interna em relação a temperatura ambiente. Nos testes aferidos de funcionamento da pastilha de Peltier, em uma temperatura ambiente de 20,2 graus celsius, a temperatura interna do refrigerador atingiu o mínimo de 10,1 graus celsius.

Em dias onde a temperatura externa passou dos 30 graus celsius, o refrigerador não conseguiu manter temperaturas abaixo dos 15 graus celsius. Em cenários com temperaturas mais amenas do que a citada acima, havia apenas um maior período de tempo para se atingir a temperatura ideal.

Uma última observação referente ao projeto inicial é a mobilidade, a idealização do mesmo permitia não só o armazenamento dos produtos termolábeis, mas também a possibilidade de utiliza-lo com uma bateria para transportar os materiais de um ponto ao outro. Agora utilizando o servidor para obtenção do valor de PWM, é necessário conexão constante com o servidor para o seu correto funcionamento.

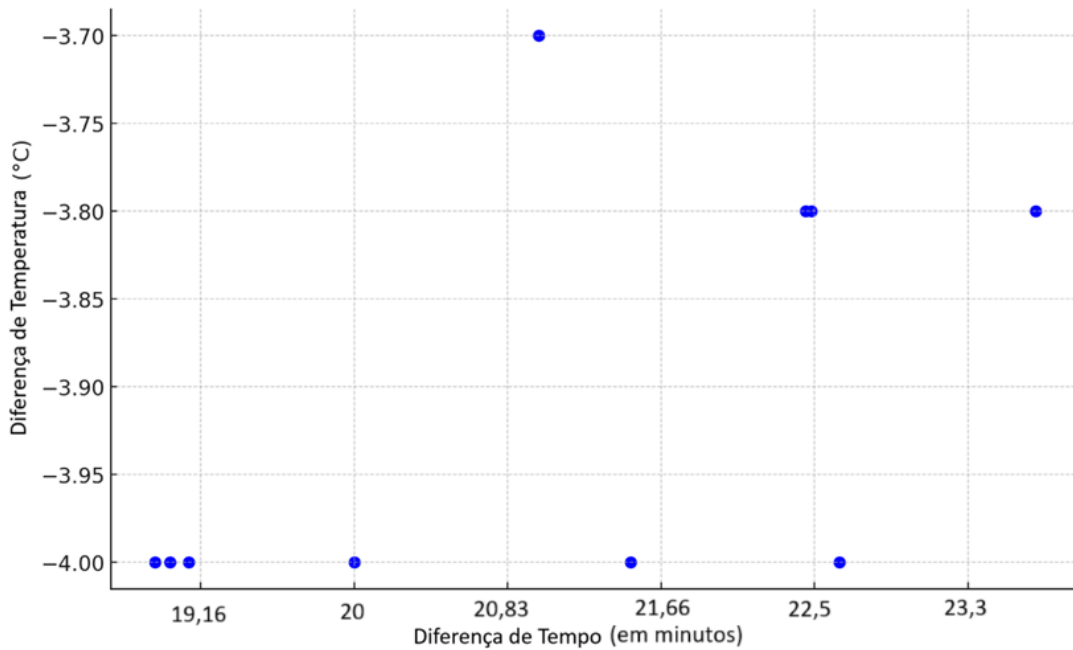
Caso haja a intenção de uma melhoria no sistema para permitir a mobilidade do mesmo, é necessário implementar no hardware um suporte para chip GSM, para que se possa ter conexão constante com a internet mesmo sem Wi-Fi conectado.

Tabela 1 – Queda de temperatura

| Temperatura (°C) | | Tempo (hh:mm:ss) | | Queda de Temperatura (°C) | Duração (minutos) | Queda (°C/minuto) |
|------------------|-------|------------------|----------|---------------------------|-------------------|-------------------|
| Inicial | Final | Inicial | Final | | | |
| 23.8 | 19.8 | 19:10:44 | 19:29:39 | 4 | 18.92 | 0.211 |
| 19 | 15.2 | 22:09:30 | 22:31:59 | 3.8 | 22.48 | 0.169 |
| 22.5 | 18.5 | 20:00:00 | 20:19:00 | 4 | 19 | 0.211 |
| 18.2 | 14.5 | 23:00:00 | 23:21:00 | 3.7 | 21 | 0.176 |
| 24 | 20 | 21:00:00 | 21:20:00 | 4 | 20 | 0.2 |
| 20 | 16 | 18:15:52 | 18:37:22 | 4 | 21.5 | 0.186 |
| 23.5 | 19.5 | 18:29:26 | 18:48:32 | 4 | 19.1 | 0.209 |
| 19.5 | 15.7 | 21:43:39 | 22:07:21 | 3.8 | 23.7 | 0.16 |
| 23 | 19 | 20:28:43 | 20:51:21 | 4 | 22.63 | 0.177 |
| 18.8 | 15 | 23:22:35 | 23:45:02 | 3.8 | 22.45 | 0.169 |

Fonte: Do autor (2024).

Figura 8 – Queda de Temperatura X Tempo



Fonte: Do autor (2024).

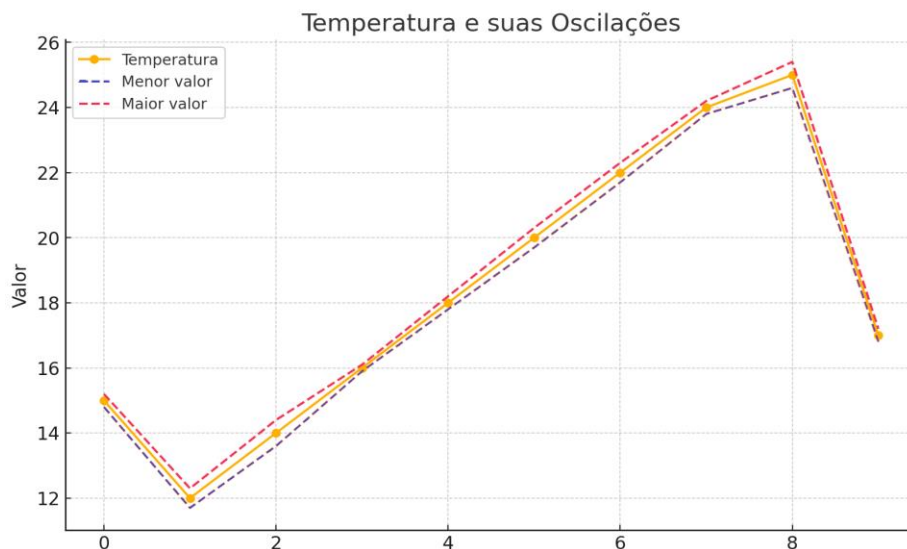
Conforme observado na tabela acima, o sistema possui uma capacidade de resfriamento média de aproximadamente 0.187 graus por minuto, ou um grau celsius a cada 5 minutos e 21 segundos. O hardware envia cerca de cinco registros por minuto ao servidor, utilizando uma média de 107 registros por teste para obter os valores registrados.

Tabela 2 – Variação de temperatura

| Temperatura (°C) | Offset (°C) | Menor Valor (°C) | Maior Valor (°C) |
|------------------|-------------|------------------|------------------|
| 15 | 2 | 14.8 | 15.2 |
| 12 | 3 | 11.7 | 12.3 |
| 14 | 4 | 13.6 | 14.4 |
| 16 | 1 | 15.9 | 16.1 |
| 18 | 2 | 17.8 | 18.2 |
| 20 | 3 | 19.7 | 20.3 |
| 22 | 3 | 21.7 | 22.3 |
| 24 | 2 | 23.8 | 24.2 |
| 25 | 4 | 24.6 | 25.4 |
| 17 | 2 | 16.8 | 17.2 |

Fonte: Do autor (2024).

Figura 9 – Variação de temperatura



Fonte: Do autor (2024).

A tabela acima mostra as variações dos valores de temperatura ao atingirem a estabilização na temperatura ideal, demonstrando uma precisão fixa de 10% em relação ao offset, e aproximadamente 1.5% de taxa de erro em relação a temperatura em si.

5. CONCLUSÃO

Este trabalho aplicou os conceitos de Internet das Coisas em um protótipo de refrigerador que possibilita gerenciar o controle e monitoramento do mesmo via internet, sem a necessidade direta da presença do usuário no mesmo ambiente do refrigerador. Um sistema automatizado de controle e monitoramento de temperatura garante a eficácia dos produtos armazenados, e reduz a perda de produtos valiosos para a saúde humana. A aplicação demonstrou ser um sistema flexível e fácil de usar, bastando apenas de uma rede Wi-Fi para conectar-se ao servidor.

Os resultados alcançados mostraram também a aplicabilidade da lógica Fuzzy em sistemas de controle refinados, demonstrando a exatidão do método mesmo com valores imprecisos, trazendo mais flexibilidade em relação a modelos de lógica booleana. A lógica Fuzzy permite que termos vagos e subjetivos sejam usados para descrever as condições do sistema. Por exemplo, em vez de "alto" e "baixo" para temperatura, podemos usar termos como "quente", "morno" e "frio", que são mais

próximos da linguagem humana e permitem uma modelagem mais intuitiva do comportamento do sistema.

O algoritmo empregado demonstrou capacidade de monitorar e controlar a temperatura, porém o hardware implementado limitou o uso do sistema em locais com a disponibilidade de conexão Wi-Fi e também com uma temperatura ambiente relativamente baixa, o que pode ser um fator impeditivo para operações em regiões mais quentes e/ou isoladas.

Embora os testes realizados tenham sido positivos, para alcançar um objetivo de comercialização ou implementação em unidades de saúde, devem ser avaliadas melhorias no hardware, como a implementação de uma melhor vedação e adição de mais uma pastilha de Peltier para poder diminuir ainda mais os valores de temperatura atingidos

Com base nos conhecimentos adquiridos, bem como nos resultados obtidos, propõe-se para futuros trabalhos: Adicionar uma pastilha de Peltier ao sistema, melhorar a vedação do refrigerador, utilizar um microcontrolador Raspberry que possibilita a escrita em outras linguagens de programação, possibilitando a inserção da lógica de controle diretamente no microcontrolador.

REFERÊNCIAS

ARAÚJO, Thierry-Yves Alves - **DESENVOLVIMENTO DE PROTÓTIPO DE DISPOSITIVO DE MONITORAMENTO DE TEMPERATURA E GEOLOCALIZAÇÃO DE VACINAS DURANTE O TRANSPORTE E EM UNIDADES BÁSICAS DE SAÚDE POR MEIO DE TELEMETRIA PARA DISPOSITIVOS MÓVEIS**. 2023. 67 f. TCC (Graduação) - Curso de Engenharia Eletrônica, Universidade do Estado do Amazonas, Manaus, 2023. Disponível em: <http://177.66.14.82/handle/riuea/4776>. Acesso em: 04 out. 2023.

BRITTO, José Augusto Alves de - **A importância da vacinação não está somente na proteção individual, mas porque ela evita a propagação em massa de doenças que podem levar à morte ou a sequelas graves**. 2021. Disponível em: <https://www.incqs.fiocruz.br/index.php?option=com_content&view=article&id=1721:a-importancia-da-vacinacao-nao-esta-somente-na-protacao-individual-mas-porque-ela-evita-a-propagacao-em-massa-de-doencas-que-podem-levar-a-morte-ou-a-sequelas-graves&catid=114&Itemid=166#:~:text=na%20pessoa%20acometida,-,A%20import%C3%A2ncia%20da%20vacina%C3%A7%C3%A3o%20n%C3%A3o%20est%C3%A1%20somente%20na%20prote%C3%A7%C3%A3o%20individual,e%20sa%C3%BAde%20das%20pessoas%20vitimizadas>. Acesso em: 23 set. 2023.

FERREIRA, D. et al. - **PROTÓTIPO PARA MONITORAMENTO DE REFRIGERAÇÃO DE VACINAS BASEADO NA INTERNET DAS COISAS**. 15ª JORNADA CIENTÍFICA

E TECNOLÓGICA DO IFSULDEMINAS, 23 out. 2023. Disponível em: <<https://josif.ifsuldeminas.edu.br/ojs/index.php/anais/article/view/873>>. Acesso em: 11 de out. 2023.

INFRAMETRO - **Desperdício de vacinas no Brasil: Saúde indo para o lixo**. 2022. Disponível em: <<https://inframetro.com.br/desperdicio-de-vacinas-no-brasil-saude-indo-para-o-lixo/>>. Acesso em: 18 set. 2023.

LIMA, M. et al. - **MONITORAMENTO SEM FIO DE AMBIENTES CLIMATIZADOS**. 2021. Disponível em: <https://repositorio.animaeducacao.com.br/handle/ANIMA/35454>. Acesso em 14 set. 2023

NOLETTO, Amanda Chini - **DESENVOLVIMENTO DE UM SISTEMA DE AQUECIMENTO CONTROLADO: Uma aplicação da pastilha de Peltier**. 2022. 58 f. TCC (Graduação) - Curso de Engenharia Eletrônica, Universidade Tecnológica Federal do Paraná, Toledo, 2022. Disponível em: <https://riut.utfpr.edu.br/jspui/bitstream/1/31428/1/desenvolvementosistemaaquecimento.pdf>. Acesso em: 14 set. 2023.

PINTO, Jones Castro - **SISTEMA DE MONITORAMENTO SEM FIO DE TEMPERATURA APLICADO A DOSES DA VACINA CONTRA COVID-19**. 2021. 57 f. TCC (Graduação) - Curso de Engenharia de Telecomunicações, Universidade Federal do Amazonas, Manaus, 2021. Disponível em: https://riu.ufam.edu.br/bitstream/prefix/6018/6/TCC_JonesPinto.pdf. Acesso em: 25 set. 2023.