

IoT como apoio a métricas e monitoramento de temperatura em refrigeradores para pequenas empresas

Thais Pinheiro da Cruz Vieira, Filipe Costa Ule,
Carlos Magnus Carlson Filho, Mário Henrique de Souza Pardo

thaispinheiro364@gmail.com; filipeule@gmail.com;
carlos.carlson@fatec.sp.gov.br; mario.pardo@fatec.sp.gov.br

Faculdade de Tecnologia de São José do Rio Preto, 2024

Resumo: Este artigo explora a implementação da Internet das Coisas (IoT) como uma ferramenta de suporte para monitoramento e controle de temperatura em sistemas de refrigeração. Essa solução é versátil e aplicável a vários segmentos empresariais, especialmente pequenos negócios, pelo baixo custo financeiro e facilidade de implantação que o dispositivo propõe alcançar. A proposta envolve o uso de sensores IoT para monitorar as condições térmicas dos refrigeradores, com a capacidade de enviar alertas por e-mail, SMS e *push notifications* caso ocorram alterações indesejadas. A escolha do protocolo *Message Queuing Telemetry Transport* (MQTT) é justificada, destacando sua eficiência frente a outros protocolos existentes. O objetivo dessa abordagem é garantir a integridade dos produtos sensíveis à temperatura, ao mesmo tempo em que proporciona uma gestão mais eficiente e otimizada dos recursos disponíveis nas pequenas empresas.

Palavras-chave: IOT, MQTT, monitoramento, temperatura, refrigeradores, pequenas empresas.

Abstract: *This article explores the implementation of the Internet of Things (IoT) as a support tool for temperature monitoring and control in refrigeration systems. This solution is versatile and applicable to various business segments, especially small businesses, due to the low financial cost and ease of implementation that the device aims to achieve. The proposal involves IoT sensors to monitor the thermal conditions of refrigerators, with the ability to send alerts via email, SMS and push notifications if unwanted changes occur. The choice of the Message Queuing Telemetry Transport (MQTT) protocol is justified, highlighting its efficiency compared to other existing protocols. The objective of this approach is to guarantee the integrity of temperature-sensitive products, while providing more efficient and optimized management of the resources available in small businesses.*

Keywords: *IoT, MQTT, monitoring, temperature, cold chain, refrigerators, Small business.*

1. INTRODUÇÃO

Durante a pandemia de Covid-19, tornou-se evidente a urgência de promover a transformação digital em empresas, independentemente de sua área de atuação. De acordo com Nexxto (2021), surgiram numerosas indagações sobre como otimizar processos, aprimorar a gestão de recursos e, simultaneamente, modernizar e humanizar os serviços prestados aos clientes. Nesse contexto, este artigo se propõe a explorar o monitoramento de temperatura em refrigeradores, com foco na redução de desperdícios e no aumento da eficiência operacional.

Conforme dados da Organização Mundial de Saúde (OMS) de 2020, pelo menos 25% de todas as vacinas foram entregues degradadas aos cidadãos, e 50% das vacinas distribuídas podem ter perdido sua eficácia devido ao armazenamento inadequado e falhas de refrigeração, entre outros problemas de logística. A OMS destaca que a manutenção de temperaturas ideais de conservação é crucial para a eficácia das vacinas, e qualquer falha na cadeia de frio pode resultar na deterioração dos imunizantes e, conseqüentemente, no descarte. Essas falhas podem ocorrer durante o transporte, armazenamento ou distribuição.

A Agência Nacional de Vigilância Sanitária (ANVISA, 2020) publicou uma nova resolução sobre boas práticas de distribuição, armazenamento e transporte de medicamentos. A

Resolução da Diretoria Colegiada (RDC) 430/2020 estabelece que os medicamentos termolábeis devem ser armazenados conforme as recomendações do fabricante, em ambientes que possam ser termicamente qualificados. Além disso, exige que os locais de armazenamento e transporte desses medicamentos estejam equipados com dispositivos para controle e monitoramento contínuo de temperatura e umidade.

De acordo com a Universidade Federal do Vale do São Francisco (UNIVASF, 2018), e segundo a Organização das Nações Unidas para Alimentação e Agricultura (FAO), aproximadamente um terço de todos os alimentos produzidos para consumo humano é perdido ou desperdiçado anualmente. No Brasil, 10% do desperdício ocorre no campo, 30% na distribuição e armazenamento e 50% no transporte, com os consumidores sendo responsáveis por 10% do desperdício total. Freire Junior (2021), pesquisador da Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA) explica que o frio desempenha um papel crucial na preservação da qualidade dos alimentos, retardando processos como a atividade enzimática e a oxidação, além de inibir o crescimento bacteriano e prolongar a vida útil dos produtos.

De acordo com a REVISTA DO FRIO & AR-CONDICIONADO (2023), um número significativo de empresas enfrenta desafios relacionados à conservação de produtos sensíveis à temperatura, como alimentos, medicamentos ou produtos químicos. Uma parcela considerável desses produtos é perdida devido a condições inadequadas de armazenamento, resultando em prejuízos financeiros e comprometendo a qualidade dos produtos.

Silva, Almeida e Elleres (2023) pontuam que a implementação de sistemas de monitoramento de temperatura em refrigeradores torna-se fundamental para mitigar esses problemas. Por meio da tecnologia IoT, os empresários podem monitorar de forma contínua e remota as condições de temperatura dentro dos refrigeradores, recebendo alertas instantâneos em caso de desvios ou variações não autorizadas. Isso não apenas ajuda a garantir a qualidade e a segurança dos produtos armazenados, mas também permite uma resposta rápida a quaisquer problemas que possam surgir.

2. JUSTIFICATIVA

A falta de monitoramento adequado da temperatura em ambientes de armazenamento refrigerado é uma questão crítica enfrentada por muitas empresas, resultando em perdas significativas de produtos e recursos, como citado pela REVISTA DO FRIO & AR-CONDICIONADO (2023). A implementação de um sistema de monitoramento de temperatura eficaz é fundamental para garantir a qualidade e a segurança dos produtos refrigerados, bem como para evitar desperdícios financeiros associados à deterioração dos materiais.

Com o objetivo de proporcionar praticidade na implementação e precisão nos alertas, este projeto foi desenvolvido com um custo reduzido em comparação a outras ferramentas de monitoramento disponíveis no mercado. Sua intenção é fornecer resultados rapidamente após a instalação e com um excelente custo-benefício. Este sistema emerge como uma solução prática e viável especialmente destinada a empresas de pequeno e médio porte, que possuem recursos orçamentários limitados para investir em um sistema de monitoramento de temperatura.

3. OBJETIVOS

3.1 Objetivo geral

Desenvolver uma aplicação IoT dedicada ao monitoramento e alerta de temperatura em refrigeradores, com a finalidade de atender às necessidades de diversas áreas de negócios que possuam produtos sensíveis à variação de temperatura e carecem de um monitoramento constante, estável e de qualidade.

3.2 Objetivos específicos

São os seguintes:

- a) Criar um protótipo físico com o microcontrolador ESP32 e sensores que irão monitorar e controlar o ambiente/câmara fria;
- b) Aplicar a comunicação do protótipo com a plataforma Tago.io através do protocolo MQTT para salvar, apresentar, notificar e controlar dos dados;
- c) Aplicar comunicação do protótipo com a plataforma Telegram para envio de notificações urgentes e emergenciais.

4. REVISÃO DA LITERATURA

4.1 Fundamentação Teórica

4.1.1 Monitoramento inteligente de temperatura de refrigeradores: Uma abordagem com IoT

De acordo com Marsh (2022), a utilização da Internet das Coisas no monitoramento de temperatura em refrigeradores oferece diversas vantagens significativas. Os sensores inteligentes de temperatura surgem como uma solução essencial neste contexto, realizando monitoramentos contínuos e precisos das condições térmicas nos refrigeradores. Esses dispositivos desempenham um papel crucial na garantia da qualidade e segurança dos produtos armazenados.

Marsh também pontua que, ao fornecer dados em tempo real sobre as variações de temperatura, os sensores IoT permitem uma resposta imediata a quaisquer desvios dos parâmetros preestabelecidos. Isso não apenas ajuda a evitar a deterioração dos produtos devido a condições inadequadas de refrigeração, mas também possibilita uma intervenção rápida em caso de falhas nos equipamentos de refrigeração, reduzindo os custos associados a perdas de produtos e fortalecendo a garantia de qualidade e segurança dos produtos ao longo de toda a cadeia de suprimentos.

4.1.2 *Message Queuing Telemetry Transport (MQTT)*

É um protocolo de comunicação *Machine-to-Machine (M2M)* com foco em IoT que funciona em cima do protocolo *Transmission Control Protocol (TCP)*. Sua comunicação se baseia entre o cliente que pode realizar tanto postagens quanto captação de informação e o *broker* que administra os dados a serem recebidos e enviados. Para essa comunicação, usa-se um paradigma chamado *publish-subscribe* (AMAZON, [s.d]). A Figura 1 resume a ideia.

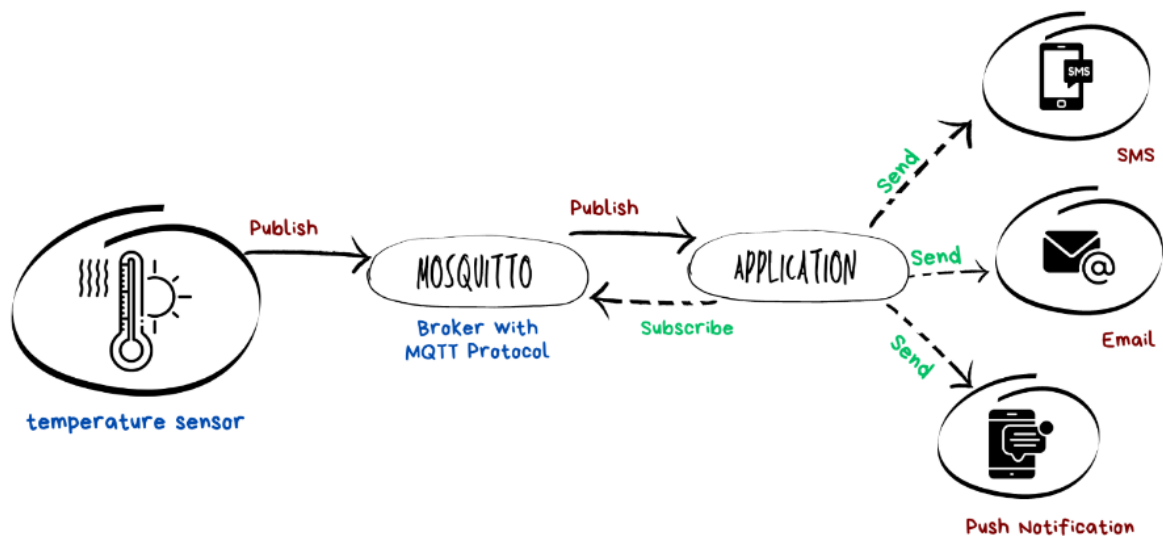


Figura 1 - Modelo de arquitetura MQTT.
 Fonte: ULE e VIEIRA (2023)

Por que MQTT e não outros protocolos?

O MQTT foi criado em 1999 pela IBM para ser usado especificamente no setor de petróleo e gás, com o intuito de monitorar oleodutos remotamente via satélite. O MQTT surgiu a partir da necessidade de um protocolo que operasse com largura de banda reduzida e consumo mínimo de energia, sendo, dessa forma, um recurso simples, de baixo consumo de dados e que possibilita uma comunicação bilateral (AMAZON, [s.d]).

Por que uma mensageria e não *Hypertext Transfer Protocol (HTTP)*?

Para abordar esta questão é preciso iniciar apontando suas diferenças de arquitetura. Angel (2020) explica que o HTTP funciona com um modelo cliente/servidor ou *request-response*, como exemplifica a Figura 2, enquanto o MQTT funciona por meio de publicações e assinaturas de um tópico (*publish-subscriber*).

No MQTT a conexão é mantida e *pings* podem ser compartilhados para mantê-la aberta, já o HTTP cria uma conexão apenas quando uma solicitação necessita ser enviada.

E, por fim, é válido acrescentar que o HTTP mantém uma conexão TCP *half-duplex* (transmissões são realizadas nos dois sentidos da comunicação, mas alternadamente, nunca simultâneas) enquanto o MQTT mantém uma *full-duplex* (as transmissões ocorrem de forma simultâneas, um dispositivo pode transmitir informação ao mesmo tempo que a recebe), permitindo comunicações assíncronas entre os serviços para que os mesmos não esperem a resposta do serviço de recebimento, sendo um modelo necessário para mensageria.

Além das diferenças apontadas acima, a mensageria torna a aplicação escalável, visto que o modelo *publish-subscriber* oferece suporte à mudança de números de serviços.

Após essa breve explicação, Angel (2020) conclui que o MQTT é muito mais otimizado para esse cenário em comparação ao HTTP em taxa de transferência, consumo de bateria e largura de banda, tornando-se uma melhor opção para a maioria das aplicações IoT e sendo reconhecido como o padrão de mercado de mensageria nesse contexto.

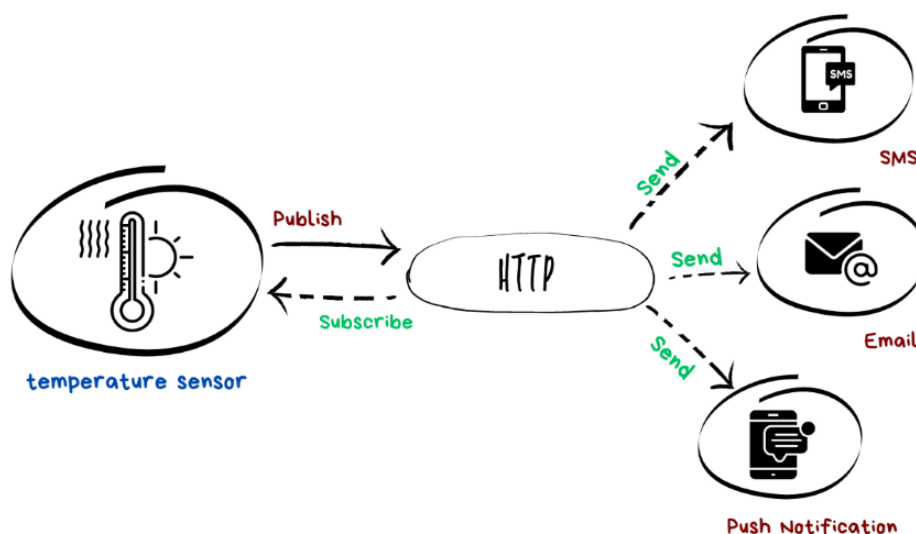


Figura 2 - Modelo de arquitetura HTTP.

Fonte: ULE e VIEIRA (2023)

4.2 Trabalhos Similares

De acordo com Pinheiro (2023), a integração da IoT na gestão da cadeia fria de produtos farmacêuticos e vacinas proporciona um controle preciso da temperatura em laboratórios. Sensores IoT permitem o monitoramento em tempo real, alertando sobre variações de temperatura através de diversos canais de comunicação. O uso do protocolo MQTT facilita a transmissão desses alertas. Além disso, análises automatizadas dos dados coletados permitem ajustes preditivos de temperatura e identificação ágil de falhas nos equipamentos.

Amaral e Maffei (2022), apresentam a construção de um protótipo de uma composteira e o desenvolvimento de sua automação com a finalidade de monitoramento da temperatura e umidade, necessário para garantir a qualidade do fertilizante orgânico produzido. Concluem que “ao adotar a automação nesse processo, é possível otimizar o controle e monitoramento da compostagem” o que contribui para a produção eficiente de composto orgânico e para a prática de uma compostagem sustentável.

Outro estudo a ser considerado é o de Moreira e Donaire (2018), que discute o desenvolvimento de um protótipo de câmara automatizada de maturação de embutidos, com monitoramento interno de temperatura e umidade. Os autores concluem que, apesar dos desvios durante o desenvolvimento, a automação resultou em maior confiabilidade das informações da câmara, facilitando seu gerenciamento. Além disso, destacam que o baixo custo do protótipo oferece uma comodidade após a inicialização, tornando-o viável principalmente para pequenos produtores.

A manutenção da integridade dos produtos farmacêuticos durante todo o processo de distribuição é crucial para garantir sua eficácia e segurança. A cadeia de frio, regulamentada pela ANVISA, desempenha um papel fundamental nesse contexto, exigindo controle preciso da temperatura desde a produção até o destino. Rodrigues (2022) propõe um sistema de monitoramento que utiliza tecnologias como Arduino, GPS e GSM GPRS para acompanhar a temperatura e umidade das caixas térmicas durante o transporte. De acordo com ele, a integração dessas tecnologias visa assegurar a qualidade dos medicamentos e imunizantes, fornecendo alertas em tempo real e facilitando a tomada de decisões para evitar danos aos produtos sensíveis à temperatura.

Ao analisar os trabalhos mencionados, é possível consolidar os fundamentos do projeto proposto, fortalecendo e ampliando a relevância da automação e seus benefícios para o setor de refrigeração e controle de câmaras frias.

5. METODOLOGIA

5.1 Tipo de pesquisa

Esse projeto está categorizado como uma pesquisa aplicada, onde se consegue gerar conhecimento para aplicação prática, visando solucionar problemas reais da sociedade por meio da busca da melhor abordagem para resolvê-los.

5.2 Ferramentas e tecnologias

Neste projeto, foi utilizada a linguagem de programação C para sistemas embarcados. Para o armazenamento, gerenciamento, criação de *dashboards* e o envio de mensagens via SMS, *push notification* e e-mail, contou-se com a plataforma Tago.io. Porém, para mensagens emergenciais, optou-se pelo uso do Telegram.

5.3 Recursos materiais

Para o desenvolvimento do protótipo, foram necessários equipamentos eletrônicos e uma caixa de isopor, simulando uma câmara de refrigeração. Os materiais utilizados são descritos abaixo:

- a) **Microcontrolador ESP32:** Escolhido pela sua praticidade e facilidade de conexão externa, pois suporta conexões *WI-FI* e *Bluetooth*.
- b) **Sensor de temperatura DS18B20:** Utilizado pela sua alta capacidade de registrar valores distintos e por ser blindado, facilitando o uso em refrigeradores e ambientes úmidos.
- c) **Protoboard 400 pontos:** Escolhida por ter uma quantidade razoável de pontos necessários para montar o projeto.
- d) **Fonte de alimentação chaveada 9V 1A:** Utilizada para alimentação do ESP32 e o sistema.
- e) **Fios de jumpers e resistores:** Utilizados para conexão do sensor de temperatura ao ESP32.

Para o desenvolvimento do código e dos *dashboards* do projeto, foi utilizado um *notebook* para escrever os códigos necessários e acessar a plataforma Tago.io via web, além de um celular para acessar a plataforma via aplicativo móvel e verificar o recebimento de mensagens e notificações disparadas.

5.4 População e amostra de dados

Os dados utilizados no projeto são obtidos através do monitoramento do refrigerador. É coletada a temperatura e armazenada em tempo real junto a data e hora da coleta, as quais são determinadas quando os dados são inseridos na plataforma.

5.5 Forma de coleta de dados

Os dados serão coletados de duas formas: A temperatura será coletada através do sensor de temperatura acoplado no refrigerador, enquanto a data e hora serão registradas na plataforma de gerenciamento dos dados no momento que esses dados forem enviados para ela.

5.6 Tratamento e análise de dados

Quando os dados são lidos do sensor de temperatura, essas informações são convertidas para uma estrutura no formato *JavaScript Object Notation* (JSON) pelo código do ESP32 e enviados para a plataforma através do MQTT. Posteriormente, esses dados são convertidos novamente em valores e são armazenados no *bucket* do dispositivo, facilitando a leitura e a análise realizada pelo usuário.

5.7 Métodos a serem utilizados para o desenvolvimento do trabalho

No projeto, foi utilizada uma combinação entre os algoritmos desenvolvidos pelos autores do projeto junto a bibliotecas externas da linguagem C para sistemas embarcados com o intuito de facilitar a coleta de dados do sensor e a sua transmissão por meio do MQTT. Também foram utilizadas estruturas de dados da linguagem, como *hashmap*, para a estruturação dos dados antes de serem convertidos para JSON.

5.8 Instrumentos e métricas

Os dados coletados pelo sensor de temperatura por meio do ESP32 são definidos em um intervalo de tempo. No momento que a aplicação é iniciada, é contabilizado um intervalo de 10 segundos para cada coleta de temperatura do refrigerador.

5.9 Cronograma de atividades

A Tabela 1 representa o cronograma definido para o desenvolvimento do projeto.

Tabela 1 - Cronograma de atividades do projeto

Cronograma de Atividades							
Atividade\ Mês	Dez.	Jan.	Fev.	Mar.	Abr.	Mai.	Jun.
Definição de escopo	X						
Escrever Fase 1 do artigo	X						
Orçamento e aquisição dos componentes		X	X				
Montagem do circuito			X	X			
Integração com Tago.io				X			
Integração com o Telegram					X		
Escrever Fase 2 do artigo						X	
Montagem do protótipo						X	
Testes e ajustes						X	X
Apresentação							X

Fonte: ULE e VIEIRA (2023)

6. DESENVOLVIMENTO

Para o desenvolvimento do protótipo deste projeto, houve uma divisão em cinco marcos:

- a) Pesquisa e definição de requisitos
- b) Projeto e montagem do circuito eletrônico
- c) Conexões do circuito com uma rede *WI-FI* e MQTT
- d) Conexões do circuito com as plataformas Tago.io e Telegram
- e) Configuração do disparo de mensagens e desenvolvimento de *dashboard* pela plataforma Tago.io

6.1 Pesquisa e definição de requisitos

Para a criação do circuito que irá coletar as informações das câmaras frias, foram pesquisados todos os equipamentos necessários de acordo com as necessidades do projeto. Para isso, foi utilizado o microcontrolador ESP32 modelo 30 barramentos e o sensor de temperatura DS18B20. O ESP32 foi escolhido por sua praticidade em se conectar a uma rede *WI-FI*, pois ele possui módulo para conexão embutido. O sensor DS18B20 foi escolhido pela sua precisão nas coletas da temperatura e por ser um sensor blindado, sendo à prova d'água, excelente para atuar em ambientes de umidade.

6.2 Projeto e montagem do circuito eletrônico

Para modelagem do circuito eletrônico, optou-se por utilizar o site WOKWI. A plataforma disponibiliza componentes semelhantes àqueles utilizados no projeto, além de permitir a simulação do código em um ambiente virtual, que facilita os testes da implementação física. O protótipo do modelo desenvolvido na plataforma é apresentado na Figura 3, demonstrando a viabilidade do circuito projetado.

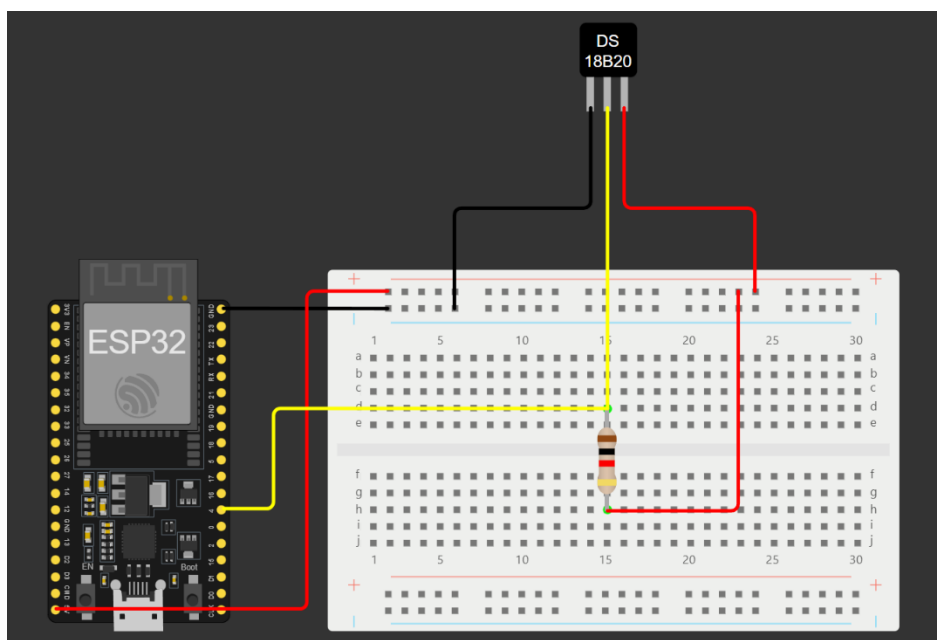


Figura 3 - Foto do projeto modelado na plataforma WOKWI.

Fonte: ULE e VIEIRA (2023)

O projeto foi iniciado a partir da montagem do ESP32, instalando-o na sua placa de expansão, que auxilia no desenvolvimento do projeto, pois facilita o acesso aos terminais dos pinos e a conexão de fios de *jumpers* aos mesmos. A placa de expansão também auxilia na alimentação do ESP32 e dos componentes do projeto, pois possui um *bourne* de alimentação, onde é possível conectar uma fonte externa à placa, que distribui a tensão para o ESP32 e para os pinos auxiliares de VCC e GND.

Após a montagem do ESP32, é feita a montagem do DS18B20 na *protoboard* acoplada ao projeto (Figura 4). O DS18B20 possui 3 fios de conexões: dois fios de alimentação do sensor (vermelho e preto) e um fio de dados (amarelo). Então, é alimentado o sensor com 3.3V, conectando-o no pino de barramento de VCC e GND da *protoboard*, e conectando o fio de dados no pino digital GPIO 4 do ESP32. Também é conectado um resistor *pullup* de 10kOhms no fio de dados do sensor, pois, de acordo com e *LAST MINUTE ENGINEERS* (2018) e Senirio (2021), o sensor funciona com o protocolo *1-wire*, necessitando manter-se em um estado de nível lógico alto para seu correto funcionamento.

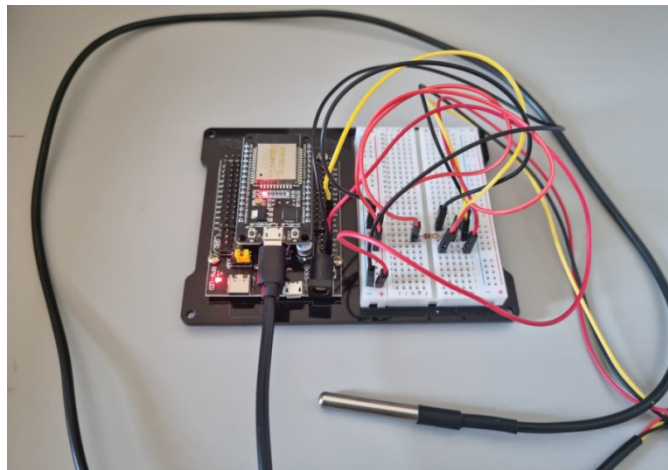


Figura 4 - Foto do ESP32 e do sensor DS18B20 montados na *protoboard*.

Fonte: ULE e VIEIRA (2023)

6.3 Conexão do circuito com uma rede WI-FI e MQTT

Após a montagem do ESP32 e do sensor DS18B20, deu-se início ao desenvolvimento do código do projeto, utilizando a linguagem C para Embarcados na programação do ESP32.

No código, a biblioteca nativa do ESP32 para conexão *WI-FI* foi utilizada, a `<wi-fi.h>`. Na função *setup*, é configurado o modo de acesso e as credenciais, realizando uma verificação enquanto o dispositivo não se conecta à rede *WI-FI* (Figura 5).

```

void setup() {
  Serial.begin(115200);
  delay(1000);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  wifiClient.setCACert(TELEGRAM_CERTIFICATE_ROOT);

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi..");
  }

  sensors.begin();

  Serial.println(WiFi.localIP());
}

```

Figura 5 - Trecho de código com configuração da conexão com rede *WI-FI*.
 Fonte: ULE e VIEIRA (2023)

Para a conexão com o MQTT, é utilizada a biblioteca `<EspMQTTClient.h>`. Nela, é possível configurar o objeto `EspMQTTClient` (Figura 6), definindo a identificação da rede *WI-FI* e seu *password*, o domínio do *Broker* MQTT padrão da Tago.io, o *username* padrão de acesso, o *token* gerado na plataforma, o nome que identifica o dispositivo e a porta do MQTT. Neste caso, optou-se pela porta 1883 do protocolo (padrão).

```

EspMQTTClient client(
  ssid,
  password,
  "mqtt.tago.io",
  "Default",
  "1b548e2f-0ea0-48e2-9bf0-5f5bf134fcc2",
  "ThermControlClient",
  1883
);

```

Figura 6 - Trecho do código com objeto `EspMQTTClient` configurado para acesso.
 Fonte: ULE e VIEIRA (2023)

6.4 Coleta da temperatura e conexões do circuito com as plataformas Tago.io e Telegram

Com a conexão *Wi-Fi* e o código MQTT configurado, o foco foi direcionado para a parte do código que será executada em *loop* (Figura 7). Nessa função, é coletado o valor atual do sensor de temperatura, o qual é armazenado em uma estrutura de chave-valor, com as propriedades *variable*, *value* e *unit*. Após o armazenamento, é feita a conversão da estrutura para o formato JSON, estrutura aceita pela plataforma Tago.io.

Feita a conversão dos dados para JSON, é possível abrir a conexão com o MQTT da Tago.io através do objeto *client* definido anteriormente, e enviar os dados através do método *publish*, onde é necessário especificar qual tópico do MQTT *Broker* essas informações serão enviadas, no caso, será o tópico *temperature*.

```
void loop() {
  sensors.requestTemperatures();
  float temperature = sensors.getTempCByIndex(0);

  StaticJsonDocument<300> TEMP;

  TEMP["variable"] = "temperature";
  TEMP["value"] = temperature;
  TEMP["unit"] = "C";

  serializeJson(TEMP, temp);

  float maxTemp = 5;
  float minTemp = -5;

  if (temperature >= maxTemp || temperature <= minTemp) {
    String mensagem = "\xE2\x9AxA0 Temperatura está fora dos parâmetros desejados \xE2\x9AxA0\n\n";
    mensagem += "\xF0\x9F\x8CxA1 Temperatura atual: ";
    mensagem += temperature;
    mensagem += " °C";

    bot.sendMessage(CHAT_ID, mensagem, "");
  }

  client.publish("temperature", temp);

  client.loop();

  delay(10000);
}
```

Figura 7 - Trecho do código com a função *loop*, coleta do sensor e envio para MQTT e Telegram.
Fonte: ULE e VIEIRA (2023)

Para enviar as mensagens ao Telegram, é definido o objeto *UniversalTelegramBot*, atribuindo no objeto o *token* do *bot* que enviará a mensagem e o ID do grupo que a mensagem será enviada. Com isso definido, é possível formatar a mensagem de destino com a temperatura atual e enviá-la para o grupo através do método *sendMessage*. O resultado é apresentado na Figura 8.



Figura 8 - Resultado do envio de mensagem através do Telegram. Visualização via *mobile*.
Fonte: ULE e VIEIRA (2023)

6.5 Configuração do disparo de mensagens e desenvolvimento de dashboard pela plataforma Tago.io

Após a coleta de dados do sensor e as conexões com o MQTT e Telegram prontas, é possível atuar na configuração da plataforma Tago.io (Figura 9). O primeiro passo é a adição e configuração do dispositivo ESP32, onde é gerado o *token* necessário para o dispositivo receber as informações do MQTT e criar o *bucket* do dispositivo, onde será armazenado os valores de temperatura.

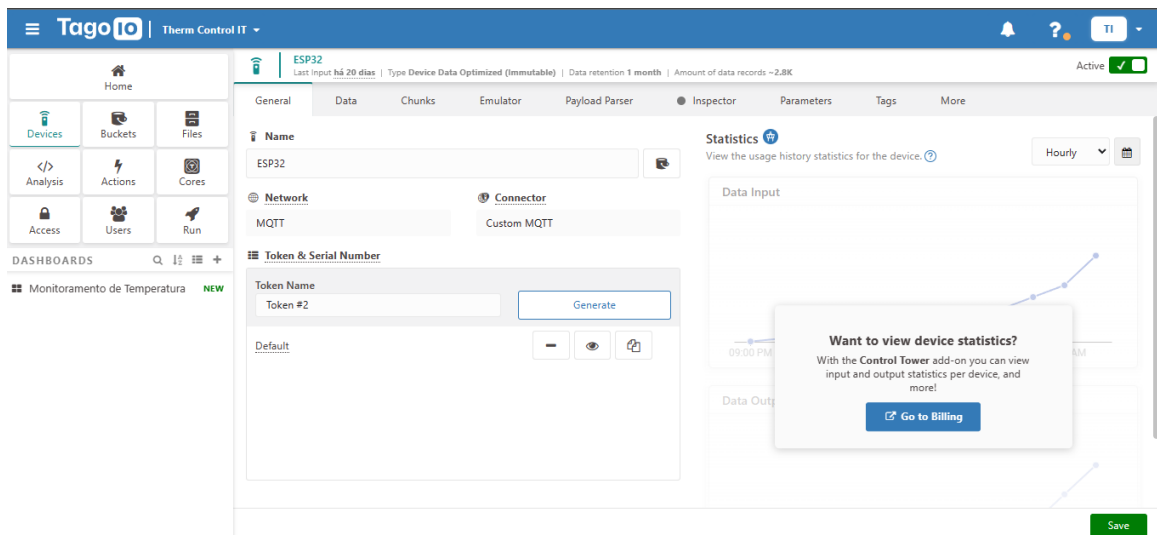
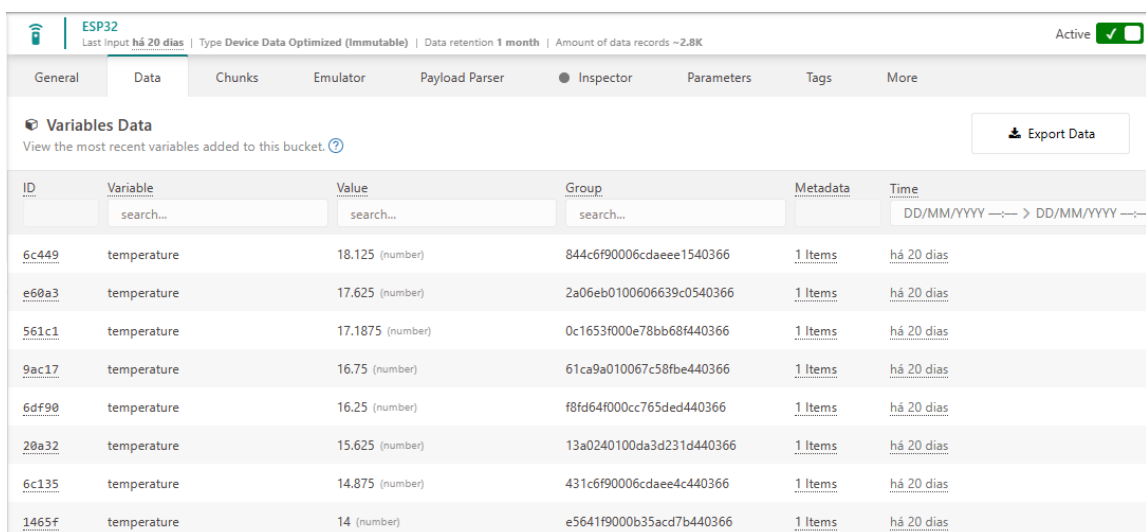


Figura 9 - Configurações gerais do dispositivo na plataforma Tago.io.
Fonte: ULE e VIEIRA (2023)

Quando os dados do sensor chegam ao *device* do ESP32 na plataforma, o dispositivo faz a conversão dos dados, que estão em JSON, para um objeto válido dentro da plataforma. Desta forma, os dados são armazenados no *bucket* do dispositivo e ficarão disponíveis para uso (Figura 10).



ID	Variable	Value	Group	Metadata	Time
6c449	temperature	18.125 (number)	844c6f90006cd4eee1540366	1 Items	há 20 dias
e60a3	temperature	17.625 (number)	2a06eb0100606639c0540366	1 Items	há 20 dias
561c1	temperature	17.1875 (number)	0c1653f000e78bb68f440366	1 Items	há 20 dias
9ac17	temperature	16.75 (number)	61ca9a010067c58fbc440366	1 Items	há 20 dias
6df90	temperature	16.25 (number)	f8fd64f000cc765ded440366	1 Items	há 20 dias
20a32	temperature	15.625 (number)	13a0240100da3d231d440366	1 Items	há 20 dias
6c135	temperature	14.875 (number)	431c6f90006cd4ee4c440366	1 Items	há 20 dias
1465f	temperature	14 (number)	e5641f9000b35acd7b440366	1 Items	há 20 dias

Figura 10 - Valores de temperatura coletados pelo sensor e armazenados no *bucket* do dispositivo.
Fonte: ULE e VIEIRA (2023)

Com os valores sendo armazenados conforme são enviados para a plataforma, é possível avançar para o desenvolvimento de *actions* e *dashboards*. As *actions* são ações dentro da plataforma que irão ocorrer caso o valor de uma variável armazenada em um *bucket* do dispositivo alcance algum limite definido, que são chamados de *thresholds*. São desenvolvidas, então, *actions* para o envio de *email*, SMS e *push notifications* caso o valor da variável de temperatura ultrapasse um certo valor de *threshold* definido. Também é definida uma *action* que dispara um e-mail caso o ESP32 perca conexão MQTT com a plataforma Tago.io, indicando uma falha de conexão com a *internet* ou uma falha elétrica do dispositivo (Figura 11).

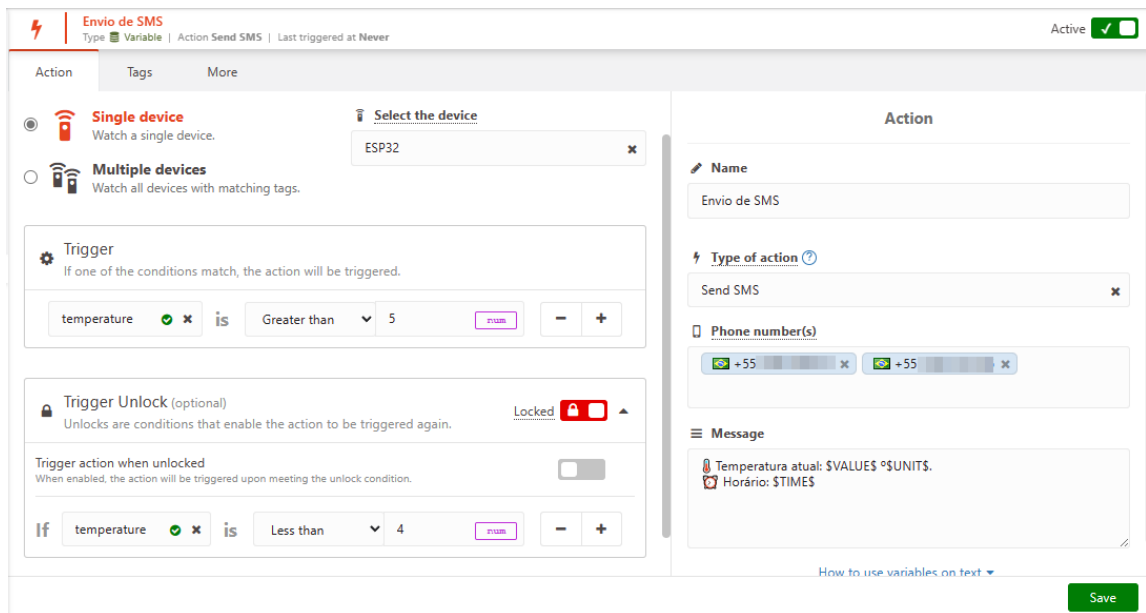


Figura 11 - Exemplo de *action* de envio de SMS na plataforma Tago.io.
Fonte: ULE e VIEIRA (2023)

Com as *actions* habilitadas, caso o dispositivo detecte um valor de temperatura acima ou abaixo do especificado na *action*, a plataforma Tago.io se encarregará de enviar a notificação pelo canal que foi designado.

A Figura 12 apresenta um exemplo de notificação por e-mail.

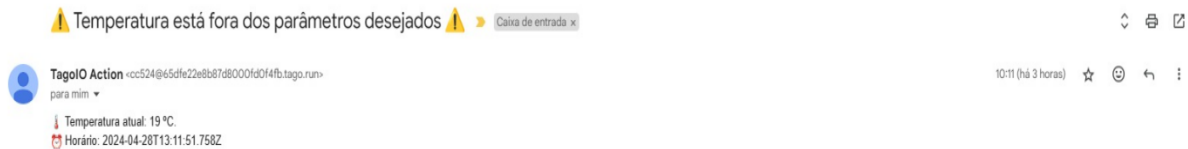


Figura 12 - Exemplo de e-mail de notificação de temperatura enviado para o cliente.
Fonte: ULE e VIEIRA (2023)

A Figura 13 apresenta um exemplo de notificação por SMS.



Figura 13 - Exemplo de SMS de notificação de temperatura enviado para o cliente.
Fonte: ULE e VIEIRA (2023)

A Figura 14 apresenta um exemplo de notificação via *push notification*.

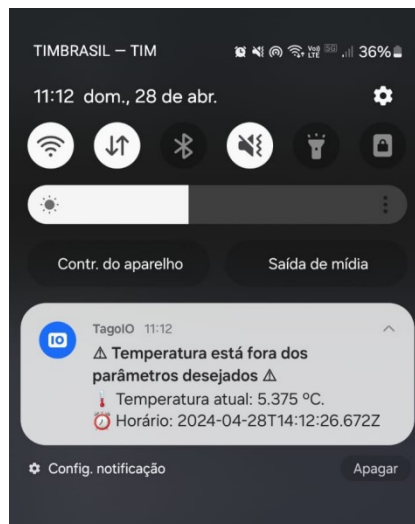


Figura 14 - Exemplo de notificação via *push notification* enviada para o cliente.
Fonte: ULE e VIEIRA (2023)

A Figura 15 apresenta um exemplo de notificação por e-mail no caso em que a plataforma deixa de receber dados do MQTT.

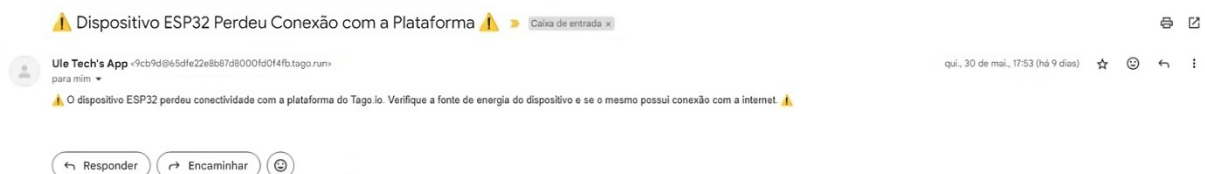


Figura 15 - Exemplo de e-mail de notificação quando a plataforma não recebe dados do MQTT.
Fonte: ULE e VIEIRA (2023)

A plataforma Tago.io também oferece a capacidade de criar *dashboards* com os dados armazenados no *bucket* do dispositivo. Nesse sentido, foi desenvolvida uma *dashboard* para visualização dos dados de temperatura coletados pelo sensor. Esta *dashboard* contém dois painéis principais: o primeiro evidencia o valor da temperatura atual, enquanto o segundo exibe um *timeseries*, com os valores coletados ao longo de um período definido na *dashboard* (Figura 16).



Figura 16 - Exemplo de *dashboard* com temperatura atual e *timeseries* da temperatura.
Fonte: ULE e VIEIRA (2023)

7. RESULTADOS E DISCUSSÕES

7.1 Discussões

Em um primeiro momento, seriam utilizados no projeto três componentes específicos: o Arduino R3, o sensor de temperatura DHT11 e o módulo WI-FI ESP8266. O Arduino R3 não possui módulo *WI-FI* embutido, então para a conexão com a plataforma Tago.io foi preciso utilizar o módulo ESP8266. Este, porém, não se mostrou compatível com as bibliotecas padrões do Arduino R3, e não foi possível adaptar as bibliotecas do ESP8266 no Arduino por falta de compatibilidade de modelos. O sensor de temperatura DHT11 também foi descartado pela falta de precisão apresentada, coletando os valores muitas vezes de forma lenta e inexata, além de não ser um sensor blindado, que poderia queimar ou ter um mau funcionamento em uma câmara fria com alta umidade.

7.2 Investimentos

Durante o processo de desenvolvimento desse projeto, foi investido um total de R\$ 197,02. Os detalhes de preço de cada componente podem ser visualizados na Tabela 2.

Tabela 2 - Valores investidos no desenvolvimento do projeto

Lista de materiais			
Qty.	Descrição	Valor Unitário	Fornecedor
1	Kit Placa Esp32 Doit Devkit 30 pinos + Placa de expansão Esp32 30 pinos	105,50	Mercado Livre
1	Protoboard 400 Pontos	7,50	Robocore
1	Resistor 10k Ohm	0,07	Robocore
1	Fonte chaveada 9V 1A	12,90	Robocore
1	Sensor de Temperatura DS18B20 blindado	19,90	Componel
1	Kit 100 fios de jumpers macho/fêmea colorido	17,00	Componel
1	Caixa Plastica Patola Pb114/2 Preta	34,15	Mercado Livre
	TOTAL	197,02	

Fonte: ULE e VIEIRA (2023)

Durante essa fase, surgiu um imprevisto relacionado à parceria estratégica. Houve uma mudança na política de uso do protocolo MQTT para contas gratuitas, conforme apresentado na Figura 17.

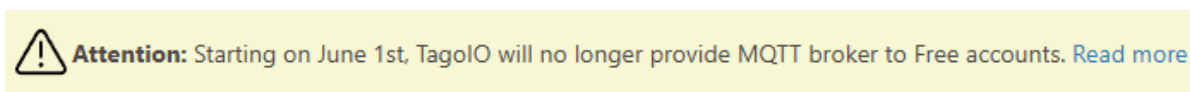


Figura 17 – Alerta de restrição de uso do MQTT *broker* para contas gratuitas.
Fonte: GUTIERRE (2024)

Como resultado, a plataforma passou a solicitar um investimento em uma das assinaturas disponíveis: *Starter* ou *Scale*, com custos a partir de U\$49,00 por mês. A Figura 18 exibe uma tabela retirada do próprio site da plataforma Tago.io, detalhando os valores e planos oferecidos.

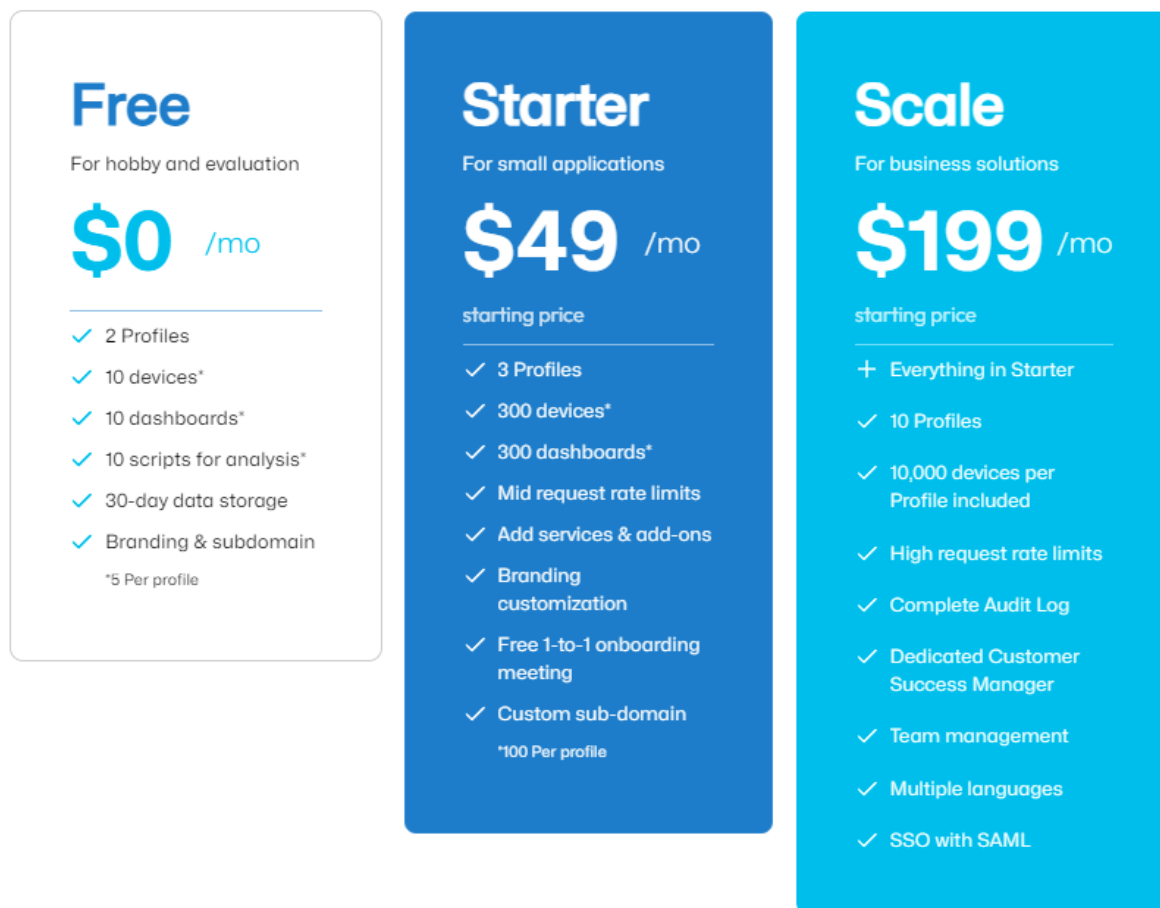


Figura 18 – Tabela de preços dos planos da Tago.io.
Fonte: Tago.io (2024)

7.3 Projetos Futuros

Hoje a aplicação está vinculada a um parceiro estratégico, a Tago.io. No entanto, foi identificado um ponto de melhoria que pode ser explorado em trabalhos futuros: a remoção dessa dependência, desenvolvendo uma plataforma própria. Isso proporcionaria um controle total sobre todos os aspectos, desde infraestrutura até a segurança de dados, permitindo uma personalização dos requisitos de acordo com necessidades específicas. Além disso, reduziria significativamente o risco de bloqueio pela Tago.io, permitindo maior liberdade para, se necessário, alterar ou migrar a solução no futuro.

A implementação de técnicas de automação baseadas no aprendizado de máquina se torna algo viável para projetos futuros. Propõe-se investigar a adoção de um modelo de *Machine Learning* para análises e tomadas de decisões a partir das coletas de temperatura. Com a aplicação dessas técnicas de automação, é possível desenvolver modelos que prevejam variações de temperatura e acionem o sistema de refrigeração realizando as alterações

necessárias, principalmente em momentos críticos. Contando com uma base de dados, tanto meteorológicos quanto do próprio refrigerador ao longo do seu ciclo de vida, o sistema de automação poderá ser capaz de identificar quando deve (e em quanto) ajustar a temperatura. Além disto, por meio de análises preditivas, terá dados suficientes para prever possíveis falhas técnicas do refrigerador ou do próprio sistema de monitoramento, promovendo ainda maior otimização e eficiência em processos manuais, aumentando a segurança do produto armazenado.

8. CONCLUSÃO

Ao concluir todas as análises e testes, fica evidente que todos os objetivos estabelecidos foram alcançados. O principal objetivo deste projeto era desenvolver uma aplicação IoT dedicada ao monitoramento e alerta de temperatura em refrigeradores, visando atender às necessidades de diversas áreas de negócios. Entende-se que esta demanda foi satisfeita com sucesso.

Todos os objetivos específicos delineados no início do desenvolvimento foram alcançados. Um protótipo físico foi criado, utilizando o microcontrolador ESP32 e sensores que irão monitorar e controlar o ambiente da câmara fria. A comunicação com a plataforma Tago.io por meio do protocolo MQTT permite armazenar, apresentar, notificar e controlar os dados, bem como comunicar-se com a plataforma Telegram para envio de notificações emergentes.

A aplicação oferece monitoramento *online* de fácil uso, com notificações de temperatura emergenciais ou não, disponíveis tanto em modo *web* quanto *mobile*. Isso permite acesso às leituras do sensor a qualquer momento e por diversos meios. O dispositivo é acessível para pequenas empresas de diversos setores devido ao seu baixo custo. Além disso, sua utilização pode ser expandida para pessoas físicas, como o uso para o monitoramento de medicamentos críticos, que exigem resfriamento constante, facilitando a vida de idosos e de cuidadores. O dispositivo é acessível, simples de configurar e direto em seu objetivo, atraindo esse público-alvo específico.

O trabalho apresenta uma solução de fácil usabilidade, comprometida com a precisão dos dados coletados, oferecendo um excelente custo-benefício e potencial de aplicação em diversos setores. Com alta possibilidade de expansão do seu uso, sendo uma ótima opção para pequenas empresas e pessoas que demandam esse tipo de solução, desde equipamentos residenciais até freezers comerciais e refrigeradores de farmácia e laboratórios, o dispositivo pode ser facilmente adaptado para monitorar condições de armazenamento, mostrando-se uma solução versátil e eficiente para uma ampla gama de necessidades.

AGRADECIMENTOS

Agradecemos aos nossos familiares e amigos pelo apoio inabalável durante o período do curso. Queremos estender nossa sincera gratidão ao nosso orientador, Carlos Magnus Carlson Filho e coorientador, Mário Henrique de Souza Pardo, por toda ajuda, conversa e paciência durante o desenvolvimento deste trabalho. Por fim, queremos reconhecer e agradecer ao Luan Thiago Maciel e ao Marildo Domingues da Silva por suas contribuições à banca fornecendo *insights* e sugestões que ajudaram a enriquecer nosso trabalho.

REFERÊNCIAS

ALMEIDA, Lucas. **5 razões do porquê medicamentos termolábeis precisam de monitoramento online**. 2020. Disponível em: <<https://nexxto.com/medicamentos-termolabeis-monitoramento-online/>>. Acesso em: 05 abr. 2024>.

ALMEIDA, Lucas. **RDC 304: tudo sobre as boas práticas de distribuição, armazenagem e transporte de medicamentos**. 2020. Disponível em: <<https://nexxto.com/rdc-304-boas-praticas-de-distribuicao-armazenagem-e-transporte-de-medicamentos/>>. Acesso em: 05 abr. 2024.

ANGEL, H. **MQTT vs HTTP: ¿qué protocolo es mejor para IoT?** 2020. Disponível em: <<https://borrowbits.com/2020/04/mqtt-vs-http-que-protocolo-es-mejor-para-iot/>>. Acesso em: 07 abr. 2024.

ANVISA (org.). **Certificado de Boas Práticas de Distribuição e Armazenagem (CBPDA)**. 2020. Disponível em: <https://www.gov.br/anvisa/pt-br/acesoainformacao/perguntasfrequentes/administrativo/certificados-de-boas-praticas/cbpda>. Acesso em: 25 maio 2024.

AMAZON. **O que é MQTT?** Disponível em: <<https://aws.amazon.com/pt/what-is/mqtt/>>. Acesso em: 07 abr. 2024.

AMAZON. **O que é IoT (Internet das Coisas)?** Disponível em: <[https://aws.amazon.com/pt/what-is/iot/#:~:text=o%20AWS%20IoT-,O%20que%20%C3%A9%20a%20Internet%20das%20Coisas%20\(IoT\)%3F,como%20entre%20os%20pr%C3%B3rios%20dispositivos](https://aws.amazon.com/pt/what-is/iot/#:~:text=o%20AWS%20IoT-,O%20que%20%C3%A9%20a%20Internet%20das%20Coisas%20(IoT)%3F,como%20entre%20os%20pr%C3%B3rios%20dispositivos)>. Acesso em: 07 abr. 2024.

AMARAL, Pedro Henrique Costa Lima do; MAFFEI, Rafael Luiz Zara. **DESENVOLVIMENTO DE UM PROTÓTIPO DE COMPOSTEIRA AUTOMATIZADA**. Fatec Rio Preto, São José do Rio Preto, v. 1, n. 1, p. 1-8, nov. 2022.

CORREIA, Wagner. **Conectando o ESP32 à plataforma TagoIO**. 2023. Disponível em: <<https://medium.com/@wagner.correia/conectando-o-esp32-%C3%A0-plataforma-tagoio-120b3852d601>>. Acesso em: 05 abr. 2024.

ELBER MEDICAL. **Termolábeis - Aprenda como fazer a gestão da cadeia do frio**. 2022. Disponível em: <<https://elbermedical.com.br/gestao/gestao-da-cadeia-do-frio/>>. Acesso em: 05 abr. 2024.

FREIRE JUNIOR, Murillo. **Conservação a frio**. 2021. Disponível em: <<https://www.embrapa.br/agencia-de-informacao-tecnologica/tematicas/tecnologia-de-alimentos/processos/tipos-de-processos/conservacao-a-frio#:~:text=O%20frio%2C%20de%20acordo%20com,mantida%20ao%20longo%20do%20processamento>>. Acesso em: 25 maio 2024.

GUTIERRE, Phil. **Changes to our MQTT Broker service availability**. 2024. Disponível em: <https://help.tago.io/portal/en/community/topic/changes-to-our-mqtt-broker-service-availability>. Acesso em: 26 maio 2024.

IFOPE EDUCACIONAL. **Técnicas de conservação de alimentos: tudo o que você precisa saber.** 2020. Disponível em: <<https://blog.ifoep.com.br/metodos-conservacao-dos-alimentos/>>. Acesso em: 05 abr. 2024.

KNIGHT, Nicola J. et al. **Talk2Lab: The Smart Lab of the Future.** 2020. Disponível em: <<https://ieeexplore.ieee.org/abstract/document/9094640>>. Acesso em: 07 abr. 2024.

LAST MINUTE ENGINEERS. **Interfacing DS18B20 1-Wire Digital Temperature Sensor with Arduino.** 2018. Disponível em: <<https://lastminuteengineers.com/ds18b20-arduino-tutorial/>>. Acesso em: 05 abr. 2024.

MAGALHÃES, Raquel Maria de. **Desperdício de alimentos, impactos na sociedade e alternativas: uma revisão.** 2022. 71 f. TCC (Graduação) - Curso de Engenharia de Alimentos, Universidade Federal de Uberlândia, Patos de Minas - Mg, 2022. Disponível em: <<https://repositorio.ufu.br/bitstream/123456789/36399/1/Desperd%C3%ADcioAlimentosImpactos.pdf>>. Acesso em: 05 abr. 2024.

MARSH, Jane. **The IoT Is the Best Solution for Cold Chain Management Problems.** 2022. Disponível em: <<https://www.allthingsupplychain.com/the-iot-is-the-best-solution-for-cold-chain-management-problems/>>. Acesso em: 07 abr. 2024.

MORO, Raabe. **O MONITORAMENTO DE TEMPERATURA NOS HOSPITAIS DO BRASIL.** 2022. Disponível em: <<https://sensorweb.com.br/o-monitoramento-de-temperaturas-nos-hospitais-do-brasil/>>. Acesso em: 05 abr. 2024.

MQTT TagoIO. [S.I]: Albacore Tecnologia, 2022. (8 min.), son., color. Disponível em: <<https://www.youtube.com/watch?v=a6oKPeYzp1Y>>. Acesso em: 05 abr. 2024.

MOREIRA, Alef Clinton Sebastião; DONAIRE, Gabriel Felix Porto; PARDO, Mario Henrique Souza. **Desenvolvimento de Protótipo de Câmara de Maturação de Embutidos Automatizada com Monitoramento Remoto.** Fatec Rio Preto, São José do Rio Preto, v. 1, n. 1, p. 1-21, nov. 2018.

NEXXTO. **Desperdício de medicamentos: como evitar perdas.** 2022. Disponível em: <https://www.linkedin.com/pulse/desperd%C3%ADcio-de-medicamentos-como-evitar-perdas-nexxto/?trk=organization-update-content_share-article%3E>. Acesso em: 07 abr. 2024.

POONGOTHAI, M. **Implementation of IoT based Smart Laboratory.** 2018. Disponível em: <https://www.researchgate.net/publication/327723745_Implementation_of_IoT_based_Smart_Laboratory>. Acesso em: 07 abr. 2024.

PERNAMBUCO. UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO. (org.). **Opinião: por que devemos falar sobre o desperdício de alimento?** 2018. Disponível em: <<https://portais.univasf.edu.br/sustentabilidade/noticias-sustentaveis/opiniaoporque-devemos-falar-sobre-o-desperdicio-de-alimento>>. Acesso em: 25 maio 2024.

PINHEIRO, Thais. **TECNOLOGIAS INTERSYSTEMS E IOT COMO APOIO A MÉTRICAS E MONITORAMENTO DE TEMPERATURA EM REFRIGERADORES DE LABORATÓRIOS DE SAÚDE.** 2023. Disponível em:

<<https://pt.community.intersystems.com/post/tecnologias-intersystems-e-iot-como-apoio-m%C3%A9tricas-e-monitoramento-de-temperatura-em>>. Acesso em: 26 maio 2024.

RANDON NERD TUTORIALS. **Telegram: Control ESP32/ESP8266 Outputs (Arduino IDE)**. 2020. Disponível em: <<https://randomnerdtutorials.com/telegram-control-esp32-esp8266-nodemcu-outputs/>>. Acesso em: 05 abr. 2024.

REDAÇÃO NEXXTO. **Erros no armazenamento: como evitar a perda dos insumos de saúde**. 2021. Disponível em: <<https://nexxto.com/erros-no-armazenamento-como-evitar-a-perda-dos-insumos-de-saude/>>. Acesso em: 05 abr. 2024.

REDHAT. **O que é um servidor Linux?** 2022. Disponível em: <<https://www.redhat.com/pt-br/topics/linux/linux-server>>. Acesso em: 11 dez. 2023

REVISTA DO FRIO & AR CONDICIONADO. **Cadeia do frio minimiza desperdício de alimentos**. 2023. Disponível em: <<https://revistadofrio.com.br/cadeia-do-frio-minimiza-desperdicio-de-alimentos/>>. Acesso em: 05 abr. 2024.

SENIRIO, João. **Entendendo os Resistores de Pull-Up e Pull-Down**. 2021. Disponível em: <https://blog.eletragate.com/entendendo-os-resistores-de-pull-up-e-pull-down/>. Acesso em: 19 maio 2024.

SILVA, Vitor Albarado da; ALMEIDA, Amon Naasson de Jesus da Silva; ELLERES, Pablo Augusto da Paz. **CONTROLE DE TEMPERATURA COM SENSORES E IOT APLICADA À GESTÃO DA QUALIDADE NA INDÚSTRIA**. 2023. Disponível em: <<https://revistaft.com.br/controle-de-temperatura-com-sensores-e-iot-aplicada-a-gestao-da-qualidade-na-industria/>>. Acesso em: 05 abr. 2024.

SOLA, Rajesh. **MQTT: Get started with IoT protocols**. 2016. Disponível em: <<https://www.opensourceforu.com/2016/11/mqtt-get-started-iot-protocols/>>. Acesso em: 07 abr. 2024.

TAGO.IO (org.). **MQTT - Process data, Publish it and Subscribe to a topic**. Disponível em: <https://help.tago.io/portal/en/kb/articles/12-mqtt-process-data-publish-it-and-subscribe-to-a-topic#Connecting_to_TagoIO_MQTT_Broker>. Acesso em: 05 abr. 2024.

TORRES, Alcides; ZENATTI, Julia. **A cultura do desperdício**. 2022. Disponível em: <<https://www.scotconsultoria.com.br/imprimir/noticias/55560>>. Acesso em: 05 abr. 2024.