

**SITUAÇÃO DE APRENDIZAGEM – DOCUMENTO TÉCNICO**

**Desenvolvimento de Sistemas**

**KeroFrio**

**Integrantes:**

Adilson António Coragem

Edilene Nunes Santos Souto

Eugénio Tchanjongombe Evaristo Mateus

Luara Santos Souto

**Professores:**

Eduardo Rahal Martins Barbosa

Fernando Costenaro Silva

Florianópolis

Junho de 2023

**Histórico de alteração do projeto**

Durante a elaboração deste projeto passou-se por diversas fases, questionamentos e até mesmo contradições no seu todo. Primeiramente, a ideia inicial do projeto seria construir um sistema que lê e, ao mesmo tempo, manipula e controla a temperatura do ar condicionado de forma remota a partir de qualquer lugar onde quer que o usuário estivesse. Quando se trata de manipulação da temperatura neste trabalho quer com isso dizer que o usuário teria a possibilidade de ligar, desligar, controlar e mudar remotamente a temperatura do ar condicionado a partir de um dispositivo web e/ou móvel.

Todavia, ao longo da produção e desenvolvimento do projeto deu a entender que a opção de controle e manipulação do ar condicionado por parte do usuário não seria viável devido a vários fatores, cita-se principalmente fatores que envolvem a fraca disponibilidade de materiais e equipamentos eletrônicos que a presente instituição (SENAI) fornece. Outrossim, destaca-se o curto prazo de tempo que temos para fazer a pesquisa, desenvolvimento e apresentação do projeto no seu todo.

No entanto, dada a complexidade de funcionalidade do sistema em que inicialmente se deu a proposta do projeto demandaria mais tempo para sua execução. Dessa forma, após algumas análises e conversas com o professor Fernando, que leciona a disciplina *“Internet das Coisas”*, entendeu-se que seria necessário repensar a complexidade do funcionamento e desenvolver algo que seria mais executável tendo em conta o prazo de entrega e a escassez dos materiais e equipamentos eletrônicos disponibilizados pela instituição.

Diante do ocorrido, chegou-se à conclusão de propor um sistema que tem a funcionalidade de ligar e desligar o ar condicionado através de um controle de forma remota, do mesmo modo, tem a capacidade de monitorar a temperatura e ligar automaticamente caso a temperatura for maior ou igual a 24°C, caso o contrário ela não será ligada, só poderá ser ligada caso o usuário decidir ligar manualmente.

| **Data** | **Versão** | **Descrição** | **Autor** |
| --- | --- | --- | --- |
| 20/04/2023 | 1.0 | Versão Inicial | Grupo geral |
| 06/06/2023 | 1.1 | Adição do botão de controle do grau de temperatura. | Luara Souto |

**Sumário**

[**1. Introdução** 4](#_heading=h.2et92p0)

[**2. Objetivos** 4](#_heading=h.tyjcwt)

[**2.1 Objetivo Geral** 4](#_heading=h.3dy6vkm)

[**2.2 Objetivos Específicos** 4](#_heading=h.1t3h5sf)

[**3. Descrição geral do sistema** 5](#_heading=h.4d34og8)

[**3.1. Descrição do problema** 5](#_heading=h.2s8eyo1)

[**3.2. Principais Envolvidos e suas Características** 5](#_heading=h.17dp8vu)

[**3.2.1. Usuários do Sistema** 5](#_heading=h.3rdcrjn)

[**3.2.2. Desenvolvedores do Sistema** 6](#_heading=h.26in1rg)

[**4. Especificações de requisitos do sistema** 6](#_heading=h.lnxbz9)

[**4.1 Requisitos funcionais** 6](#_heading=h.35nkun2)

[**4.2 Requisitos Não-funcionais** 6](#_heading=h.1ksv4uv)

[**5. Diagramas de casos de uso** 7](#_heading=h.44sinio)

[**5.1. Detalhamento** 7](#_heading=h.2jxsxqh)

[**6. Diagramas extras** 7](#_heading=h.z337ya)

[**7. Protótipos** 8](#_heading=h.3j2qqm3)

[**8. Tecnologias** 10](#_heading=h.1y810tw)

[**8.1 Sensores, atuadores, configurações e programação do sistema** 10](#_heading=h.4i7ojhp)

[**8.2 Protocolo, plataforma e suas configurações** 12](#_heading=h.2xcytpi)

[**8.3. Integração, conexão e envio de dados** 13](#_heading=h.1ci93xb)

[**8.4. Dashboard e informações coletadas** 13](#_heading=h.3whwml4)

[**9. Evolução do sistema** 13](#_heading=h.2bn6wsx)

[**10. Conclusão** 14](#_heading=h.qsh70q)

[**11. Referências e outras fontes** 15](#_heading=h.3as4poj)

# **1. Introdução**

O presente projeto faz parte da avaliação final do 3º período do curso Técnico em Desenvolvimento de Sistemas e tem como objetivo apresentar de forma prática e teórica o sistema “*KeroFrio*” a partir de suas funcionalidades e comunicação com a parte eletrônica.

Nestes moldes, para realização deste projeto se fez o uso das seguintes ferramentas: Placa ESP32, Arduino, Sensor de Temperatura, Relé, TagoIO, Ar Condicionado (que durante a nossa apresentação possivelmente irá ser substituída ou adaptada por uma Lâmpada), Resistor, Fios e códigos de programação em C++.

Entretanto, em primeira instância procurou-se fazer uma descrição geral do sistema KeroFrio, dando ênfases na sua finalidade e objetivo principal. Igualmente, detalhou-se o problema que o presente sistema se propõe a dar garantias de uma solução viável. Em seguida, abordou-se sobre as características dos envolvidos do sistema, destacando principalmente os usuários e desenvolvedores do sistema.

Ademais, com o mesmo projeto também se propõe fazer uma análise sobre os requisitos do sistema, dentre os mesmos, detalhou-se de forma minuciosa os requisitos funcionais e não funcionais que se enquadram nas características e funcionalidades do sistema keroFrio.

Por último, mas não menos importante, apresenta-se os diagramas de modelagem de presente sistema, assim sendo, selecionou-se o diagrama de caso de uso - onde se fez o detalhamento da utilização do usuário com o aparelho/sistema; diagrama de atividade - deu-se ênfase sobre o procedimento de atividades relacional entre o cliente e o sistema; e diagrama de estado de máquina - neste se desdobrou desde a sua interação com o TagoIO.

# **2. Objetivos**

## **2.1 Objetivo Geral**

* Apresentar o funcionamento geral do sistema KeroFrio, considerando a sua integração com a parte eletrônica e remota;

## **2.2 Objetivos Específicos**

* Analisar as propostas de utilização do sistema KeroFrio e seus possíveis impactos na sociedade;
* Compreender os requisitos de funcionalidade, de não funcionalidade e o detalhamento do nível de monitoramento do ar condicionado no sistema KeroFrio;
* Descrever a praticidade do usuário em relação ao manuseio da temperatura do seu lar em torná-lo num ambiente mais agradável.

# **3. Descrição geral do sistema**

O projeto KeroFrio surgiu para trazer conforto e praticidade aos usuários, proporcionando comodidade em sua utilização pois mesmo a distância o usuário poderá acioná-lo e tornar o ambiente mais agradável.

KeroFrio tem como objetivo ligar e desligar um ar condicionado a distância e verificar a temperatura em tempo real, esta ação permitirá adentrar ao ambiente previamente climatizado trazendo a sensação de bem estar ao indivíduo.

## **3.1. Descrição do problema**

Normalmente, o ar-condicionado é configurado para ser acionado a partir de um controle remoto, com isso o usuário se limita a estar no ambiente de instalação para que o mesmo seja ligado e ajustado à temperatura desejada. Por conseguinte, percebe-se a necessidade de criar um sistema que permita ao usuário ter uma interação com o aparelho de forma remota com antecedência, isso fará com que o ambiente fique climatizado antes mesmo do usuário adentrar o local onde está instalado o aparelho.

Utilizando uma placa ESP32, uma protoboard, sensor de temperatura DHT11, um relé, a linguagem utilizada para programar foi a C++ e o TagoIO que é uma plataforma que controla o relé e recebe informações do sensor de temperatura através do protocolo MQTT.

## **3.2. Principais Envolvidos e suas Características**

### **3.2.1. Usuários do Sistema**

O sistema será direcionado ao usuário comum, que poderá adquirir o produto a ser instalado em sua residência ou ambiente de trabalho. Este projeto irá revolucionar a já existente era moderna viabilizando acesso a novas tecnologias e funcionalidades.

### **3.2.2. Desenvolvedores do Sistema**

O desenvolvimento do projeto foi realizado por quatro estudantes de desenvolvimento de sistemas do SENAI, onde receberam orientações de dois professores das disciplinas de Internet Das Coisas e Modelagem de Sistemas.

# **4. Especificações de requisitos do sistema**

A proposta deste projeto é criar uma solução para pessoas que apreciam um conforto a mais proporcionado por novas tecnologias. O projeto surgiu devido uma necessidade de desenvolvimento das disciplinas do 3° semestre do curso Técnico em Desenvolvimento de Sistemas. A ideia principal é ligar e desligar um ar condicionado através da plataforma TagoIO.

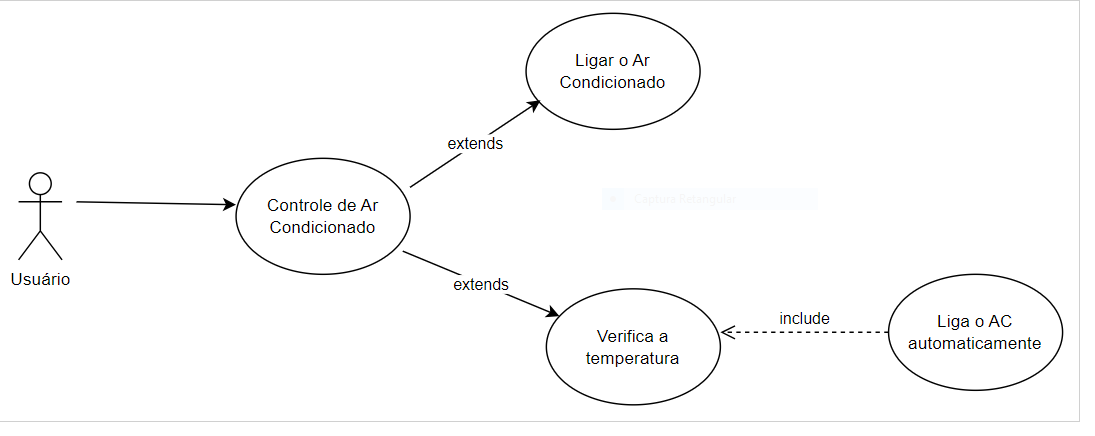
## **4.1 Requisitos funcionais**

1. **ligar ar condicionado [RF01]**
2. Ligar e desligar um ar condicionado à distância;
3. Ao acionar o ar condicionado, mostrará no dashboard o status ligado (ON);
4. Ao desligar o ar condicionado, mostrará no dashboard o status desligado (OFF);
5. **Temperatura de referência [RF02]**
   1. Permitirá o cadastro prévio de uma determinada temperatura de referência;
   2. O ar condicionado deverá ser acionado automaticamente quando a temperatura for maior ou igual a de referência;

## **4.2 Requisitos Não-funcionais**

1. **Histórico de temperaturas do ambiente [RNF 01]**
   1. No gráfico poderá verificar o histórico de temperaturas;
2. **Display de temperatura atual [RNF 02]**
   1. Deverá mostrar a temperatura atual do ambiente;

# **5. Diagramas de casos de uso**

****

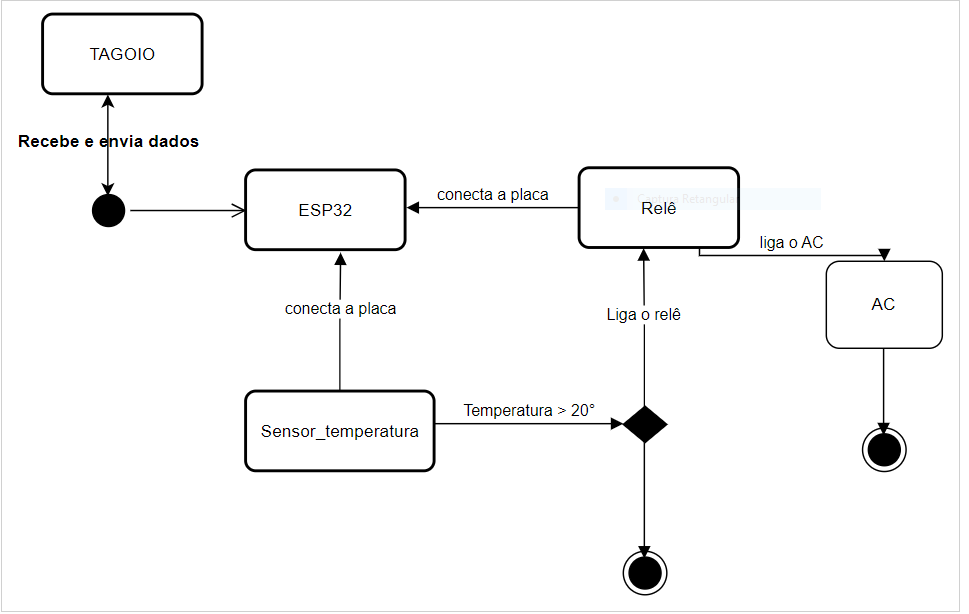
## **5.1. Detalhamento**

Dentro do diagrama de caso de uso o nosso sistema se desdobra no seguinte:

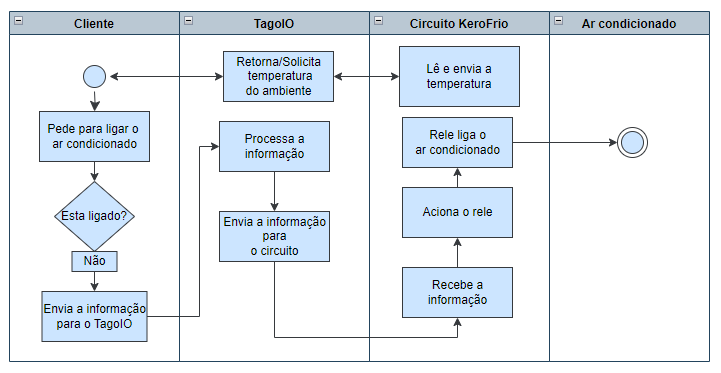
1. O **usuário** tem o domínio ou o controle de ligar e desligar o ar condicionado a partir de um dispositivo móvel, independente do espaço geográfico.
2. Quando o usuário acionar a opção ligar e/ou desligar, o aparelho de casa responde o comando e age segundo o comando.
3. O sistema comporta-se inteligentemente para verificar a temperatura do ambiente, caso esteja acima de 24ºC, ele por si só liga o ar condicionado; caso contrário, o dispositivo não liga.

# **6. Diagramas extras**

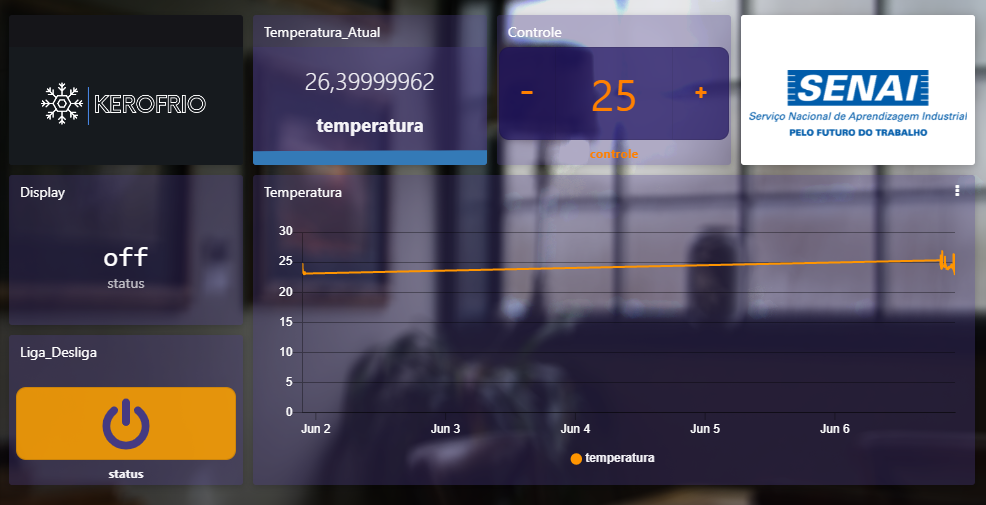
Dentre os diagramas escolhidos para referenciarmos, temos o diagrama de estado de máquina:

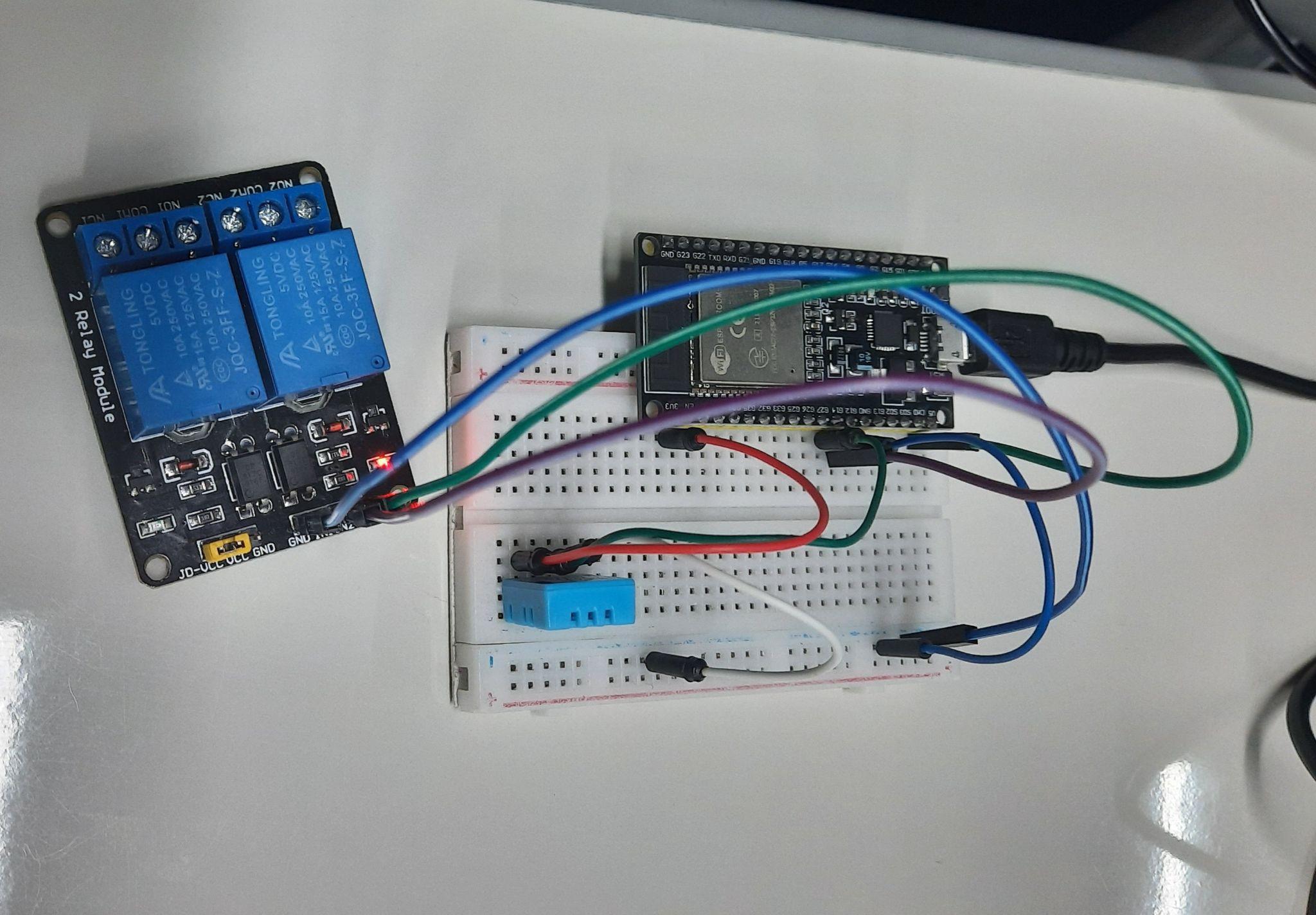
****

E também destaca-se o diagrama de atividade:



# **7. Protótipos**

****

****

# **8. Tecnologias**

Em conformidade com as tecnologias disponíveis a apresentadas durante o semestre. Não se utilizou tecnologias com licenças pagas de manutenção e atualização. Por conseguinte, foram utilizadas as seguintes ferramentas de desenvolvimento do projeto:

* C++
* TagoIO
* Placa ESP32
* Relé
* Sensor DHT11

## **8.1 Sensores, atuadores, configurações e programação do sistema**

O sensor de temperatura utilizado foi o modelo DHT11, como atuador utilizou-se o relé com dois canais. Para a instalação foi utilizado uma protoboard, fios de conexões Jumper e uma placa ESP32.

Para realizar as conexões deve-se colocar a placa ESP32 na base protoboard utilizando o lado da placa que possui o V5. Com a placa conectada, deve-se inserir o fio na protoboard em direção ao V5 e o outro lado do fio no VCC do relé, conecta-se outro fio no G12 e liga a outra ponta no IN2 no relé, outro fio no GND e no relé GND. Conecta o GND na linha negativa da protoboard. A conexão do sensor é feita ligando um fio no 3V3 da placa ESP32 e o outro lado no sensor no VCC, liga-se outro fio no G14 e no DATA ou saída, outro fio no GND e outro fio no GND a GND. Não se faz conexão no NC do sensor, este pino fica entre o GND e o DATA.

A programação foi realizada no Arduino.IDE utilizando a linguagem C++, para a construção do código é necessário adicionar as bibliotecas necessárias que foram "EspMQTTClient.h" e "DHTesp.h", em seguida a definição dos pinos de cada componente da placa que são DHT11 para o pino 14 e o relé para o pino 12.

As variáveis internas que são globais recebem o valor da temperatura conforme abaixo:

float valor\_temperatura = dht.getTemperature();

float valor\_temperatura\_ref;

O próximo passo do código deve-se adicionar o token que é gerado no TagoIo, nesse caso específico foi o seguinte:

EspMQTTClient client

(

"FIESC\_IOT", //nome da sua rede Wi-Fi

"C6qnM4ag81", //senha da sua rede Wi-Fi

"mqtt.tago.io", // MQTT Broker server ip padrão da tago

"Token", // username

**"219edbdd-05f9-4b20-87ef-47d829b6e19d", // Código do Token**

"TestClient", // Client name that uniquely identify your device

1883 // The MQTT port, default to 1883. this line can be omitted

);

Declara-se a função setup que configura os pinos de acordo com a necessidade, logo em seguida a função loop.

void setup() {

pinMode(rele, OUTPUT);

Serial.begin(9600);

dht.setup(DHT11\_PIN, DHTesp::DHT11);

}

void loop() {

delay(dht.getMinimumSamplingPeriod());

float umidade = dht.getHumidity();

float temperatura = dht.getTemperature();

Serial.print("Temperatura = ");

Serial.println(temperatura);

Serial.print("Umidade = ");

Serial.println(umidade);

//tago io

leitura\_sinais();

converte\_json();

envia\_msg();

delay(1000);

controla\_rele();

client.loop();

}

A conexão com o Tagoio é realizada através do protocolo MQTT, no código existe três funções responsáveis por enviar informações para a plataforma, função leitura\_sinais( ), converte\_json( ) e por último envia\_msg( ). A função leitura de sinais é responsável pela leitura da temperatura que será enviada para o TagoIo, a função converte\_json( ) faz a conversão da temperatura para o formato jason, a função envia\_msg( ) envia o json através do protocolo MQTT para o TagoIO.

Para receber as informações do Tago utilizou-se as seguintes funções processa\_msg que desconstrói a mensagem que vem em formato json do TagoIO e controla o relé pelo botão ligar e desligar do TagoIo, controla\_rele processa a mensagem através do controle de temperartura e onConnectionEstablished( ) estabelece uma conexão com o TagoIO para receber a mensagem dele.

**8.2 Protocolo, plataforma e suas configurações**

O protocolo utilizado foi o MQTT, é um protocolo de mensagens leves. É muito utilizado para comunicação M2M, (acrônimo para Machine-to-Machine), é um dos protocolos mais adequados atualmente para comunicação M2M em dispositivos embarcados de forma geral. Justamente por isso MQTT é popular quando se trata de internet das coisas.

A plataforma utilizada foi a TagoIO, é uma plataforma web avançada para monitorar o ambiente por meio de dispositivos IoT em rede. O TagoIO tem dois diferenciais: a velocidade com que uma solução IoT pode ser implementada no mercado e ser uma ferramenta poderosa que oferece diversas funcionalidades necessárias para tal implementação. A plataforma TagoIO permite edificar e desenvolver rapidamente soluções IoT.

O processo de construção de uma solução de IoT ocorre em quatro etapas. Primeiro, há Wi-Fi, LoRa, Sigfox, GPRS, LTE, BLE, ZigBee etc. para ler a mídia e enviar dados para a plataforma. Então você precisa escolher um modelo para edificar o aplicativo e integrar soluções, combinando dados com sistemas externos. Para concluir o processo usuários e dispositivos precisam criar e gerenciar aplicativos escalando rapidamente. Após esta configuração, você poderá monitorar o painel de instrumentos do seu sistema.

Os usuários poderão criar painéis. Com recursos para mapas e marcadores, links, imagens, formulários, gráficos, vídeos e outros widgets também podem ser adicionados mais adequadamente para a solução em tempo real.

A configuração do TagoIO se inicia criando um Device com o modelo MQTT, que será utilizado na conexão, em seguida cria-se as ações (Actions) que realiza o controle dos elementos do Dashboard, cada Actions tem a finalidade de enviar ou receber informações.

**8.3. Integração, conexão e envio de dados entre ESP32 e o tagoIO**

A conexão do ESP32 com o TagoIO ocorre como mencionado no subitem 8.1, adiciona ao código as bibliotecas EspMQTTClient e o ArduinoJson. A comunicação básica constitui apenas do EspMQTTClient para enviar e receber dados do TagoIO. O sensor envia o grau de uma determinada temperatura para a plataforma utilizada que, por sua vez, quando recebe redireciona o valor para o Dashboard.

**8.4. Dashboard e informações coletadas**

No Dashboard, ao receber as informações esta é apresentada em um display que mostra somente a temperatura atual, no gráfico é apresentado as temperaturas que foram registradas anteriormente. Possui a opção de ligar e desligar o ar condicionado e um controle para temperatura de referência que liga o ar condicionado automaticamente caso a temperatura esteja acima da que foi definida previamente. Estas informações podem ser conferidas no modelo acima no item 8. Protótipos.

# **9. Evolução do sistema**

No decorrer do desenvolvimento do projeto, observou-se outras necessidades as quais poderão ser implementadas mais a frente caso se julgue necessário. Por exemplo, alterar a temperatura para mais ou para menos de forma remota em tempo real através de uma página web ou dispositivo móvel.

Em síntese, sugere-se a criação de uma aplicação web ou aplicativo para dispositivos móveis que permita ao usuário a possibilidade de ter acesso e realizar qualquer alteração, verificar as informações de qualquer localização.

# **10. Conclusão**

Portanto, diante do exposto nos tópicos acima, percebe-se então que a limitação que se dá no controle e/ou monitoramento da temperatura de um ambiente através do ar-condicionado, se configurou como base para a construção do projeto.

Nestes moldes, o KeroFrio chega para amplificar as possibilidades do usuário de ar-condicionado, no sentido do mesmo já não se limitar dentro de um espaço geográfico para administrar o ambiente desejado.

Assim, dentro daquilo que é o seu objetivo geral, o projeto se desenha como eficiente na resolução de problemas apresentados durante o trabalho.

# **11. Referências e outras fontes**

BERTOLETTI, Pedro. **Comunicando-se via MQTT com o ESP32 (MIC404)**. Instituto NCB: Janeiro, 2020. Acesso aos 13 de junho de 2023 < disponível em: <https://www.newtoncbraga.com.br/index.php/microcontroladores/143-tecnologia/17117-comunicando-se-via-mqtt-com-o-esp32-mic404.html> >.

CASTRO, Jaelson Freire et al. Elaboração dos Modelos de Requisitos Funcionais e Não Funcionais do Sistema de Apoio às Atividades dos Laboratórios de Física. Recife: Centro de Informática, UFPE, 2016. Acesso em 16 de maio de 2023 < disponível em: <https://www.cin.ufpe.br/~if716/projetos/2016-1/ProjetoII/Equipe%207%20-%20Proj%202.pdf> >.

MELO. Ana Cristina. **Exercitando modelagem em UML** / Ana Cristina Melo. - Rio de Janeiro: Brasport, 2006. Acesso aos 27 de maio de 2023 < disponível em : <https://xdocz.com.br/doc/exercitando-modelagem-em-uml-328541lr5mox> >

RAMOS, Ricardo Argenton. **UML – Aula I Diagramas de Caso de Uso, Sequência e Colaboração.** Engenharia de Software II, 2013.1. < disponível em: <http://www.univasf.edu.br/~ricardo.aramos/disciplinas/ES_II_2013_1/UML_AulaI.pdf> > acessado aos 20 de maio de 2023.

SERRA, Ana Paula Gonçalves. Documentação de um Produto de Software. Versão 3.0. 2005, disponível em: < <http://www.waltenomartins.com.br/esof2_projeto_documento_de_sw.pdf> > acesso aos 14 de maio de 2023.

Felipe Rooke da Silva. **Documento de Requisitos do Sistema**. Módulo de Avaliação Acadêmica no SIGA Ensino, 2016. Acesso aos 20 de maio de 2023 < disponível em: <https://www2.ufjf.br/diavi//files/2016/07/DocumentosdeRequisitosdoSistema.pdf> >.

XAVIER, Raquel. Conheça a TagoIO, a primeira ferramenta cloud para desenvolvimento de solução IoT homologada Khomp. **Blog da Khomp,** 29 jan 2019. Disponível em: < <https://www.khomp.com/pt/tagoio-solucao-iot/>>. acesso em: 13 jun. 2023.