

Termostato de Controle e Monitoramento de Temperatura

Disciplina: Projeto de TCC I

Prof.^a. Helvio J. Junior

INSTITUTO FEDERAL DO RIO DE JANEIRO - CAMPUS NITERÓI
CURSO TÉCNICO DE INFORMÁTICA INTEGRADO AO ENSINO MÉDIO

GABRIEL SALES RODRIGUES
MICHAEL MONTEIRO DE LIMA
MIRYAM ESTÁBILA NOGUEIRA DE SOUZA

TERMOSTATO DE CONTROLE E MONITORAMENTO DE TEMPERATURA

Projeto de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia do Rio de Janeiro Campus Niterói como requisito parcial para obtenção de notas das disciplinas de Projeto Final de Curso (I e II) e para obtenção da colação de grau do Curso Técnico de Informática Integrado ao Ensino Médio.

Orientador: DSc. Luiz Felipe Silva Oliveira.

Mensões: Pedro Henrique Biega Lopes Bandeira e Miguel Ramos Gil.

NITERÓI/RJ

2022

Resumo

O trabalho de conclusão de curso, termostato de controle e monitoramento de temperatura tem como objetivo controlar e monitorar a temperatura das salas de aula ou laboratórios do campus. Seu objetivo é tornar prático o controle e monitoramento da temperatura, evitar eventuais problemas de saúde para frequentadores do campus, e incentivar as pessoas a prática e desenvolvimento da eletrônica, tecnologias IoT e a Pesquisa e Extensão.

Palavras-Chaves: Eletrônica, Prototipação e IoT.

Abstract

The course completion work, control thermostat and temperature monitoring aims to control and monitor the temperature of classrooms or campus laboratories. Its objective is to make temperature control and monitoring practical, to avoid possible health problems for campus visitors, and to encourage people to practice and develop electronics, IoT technologies and Research and Extension.

Keywords: Electronics, Prototyping and IoT.

LISTA DE FIGURAS

Figura 1 – Brainstorming	11
Figura 2 – Trello	11
Figura 3 – Modelo Conceitual do projeto	13
Figura 4 – Interface do Wokwi com o protótipo	16
Figura 5 – Bibliotecas utilizadas	16
Figura 6 – Código do segundo protótipo	17
Figura 7 – Interface de controle e monitoramento no MQTT dashboard	19
Figura 8 – Interface do aplicativo	20
Figura 9 – Modelo 3D no Tinkercad	21
Figura 10 – Circuitos Eletrônicos	22
Figura 11 – Primeiro protótipo físico do projeto	23

LISTA DE TABELAS

Tabela de funcionalidades previstas	14
Tabela de orçamento do projeto.	24

LISTA DE SIGLAS E ABREVIações

TCC	Trabalho de Conclusão de Curso
IDE	Integrated Development Environment
IOT	Internet of Things
MQTT	Message Queuing Telemetry Transport
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
ESP 32	Espressif 32
LoRa 32	Long Range 32
DHT	Detector for humidity and temperature
SCT	Sensor de corrente e tensão
OLED	Organic Light Emitting Diode
C++	Classes Plus Plus
Wi-Fi	Wireless Fidelity
REQ	Requisito

SUMÁRIO

1. INTRODUÇÃO	9
1.1 Contextualização e Motivação	9
1.2. Objetivos	10
1.2.1 Objetivo Geral	10
1.2.2. Objetivos Específicos	10
1.3 Metodologia	10
2. TÍTULO DO TEMA DO PROJETO	13
2.1. Descrição Geral da Proposta	13
2.2. Lista de Requisitos (Funcionalidades) previstas	14
2.3. Processo de Prototipagem	16
2.3.1 Primeiro protótipo do projeto	16
2.3.2 Protótipo versão MQTT(Segundo protótipo do projeto)	17
2.3.3 Configuração do MQTT Dashboard	19
2.3.4 Protótipo do gabinete impresso em 3D	20
2.3.5 Protótipo Físico	22
2.4. Orçamento do Projeto	24
3. REFERÊNCIAS BIBLIOGRÁFICAS	25
3.1. Fontes das referências em sites, fóruns e artigos na internet	25
3.2. Fontes das referências em vídeos na internet	25

1. INTRODUÇÃO

1.1 Contextualização e Motivação

O IFRJ campus Niterói possui vários laboratórios com equipamentos que têm que trabalhar e que precisam ser operados em condições climáticas de acordo com a especificação técnica. relatos da equipe técnica apontam que constantemente alguns equipamentos apresentam defeitos por não estarem operando em uma temperatura adequada.

Essa perda dos equipamentos por esse motivo bem como o monitoramento desses ambientes demanda tempo e custos para a instituição.

Visando solucionar esse problema, alguns alunos do semestre 2022.1 começaram a idealizar uma solução para tentar minimizar o problema reportado acima. Portanto, as seguintes propostas de projeto foram definidas: A prototipação de uma central de monitoramento e controle do ar-condicionado.

Portanto, o protótipo a ser desenvolvido é fazer um termostato na qual é possível monitorar a temperatura ambiente do ar condicionado (**Temperatura e Energia**) e também o controle do ar-condicionado(**Ligar/Desligar e Controle de Temperatura**). Esse controle e monitoramento será feito utilizando protocolo MQTT e o Aplicativo MQTT Dashboard.

Usaremos o Heltec Lora 32 que é uma versão de esp32 com o display OLED embutido junto de componentes eletrônicos como: Sensores de Temperatura (**DHT11 e DHT22**), Sensor de Corrente (**SCT-013**), e o Transmissor Infravermelho (**HX-53**). O circuito eletrônico deste projeto será feito com uma protoboard, resistores e jumps. Tudo isso dentro de um gabinete (uma caixa provavelmente) feita em impressão em 3D.

O público alvo do projeto é os discentes, docentes e equipe de manutenção para ajudar a manter o bem estar de todos no campus. Também há a questão de que se algum ar-condicionado estiver com problema, o indivíduo usuário do aplicativo saberá antes mesmo de entrar na sala, em caso de até mesmo de evitar problemas de saúde como um eventual choque térmico por exemplo.

A importância desse projeto é para facilitar o controle da temperatura dos ar-condicionado e monitorar a temperatura dos ambientes de sala em tempo real. A ideia é evitar que os alunos, docentes e servidores tenham problemas de saúde por conta de dias quentes e o mau uso do aparelho. Evitar o desperdício de energia do campus e desgastes do aparelho.

1.2. Objetivos

1.2.1 Objetivo Geral

A proposta deste trabalho é o desenvolvimento de um medidor de temperatura de ambientes para monitorar o funcionamento do ar-condicionado e a temperatura do ambiente. No IFRJ campus Niterói, existe a necessidade de controlar a temperatura das salas de aulas e também manter os alunos do campus em uma temperatura ideal, bem como prover meios de economizar a energia da escola.

1.2.2. Objetivos Específicos

Medidor de temperatura de sala, assim podendo saber se a temperatura, a temperatura da saída do ar-condicionado e saber se está ligado ou não. Isso será feito com um Heltec Lora 32 e os sensores DHT11, DHT22 e SCT-013. Também será feito o controle da temperatura do ar-condicionado usando protocolo MQTT, aplicativo MQTT Dashboard e o Transmissor infravermelho HX-53.

Os objetivos específicos que auxiliaram no atendimento do objetivo geral deste trabalho são os seguintes:

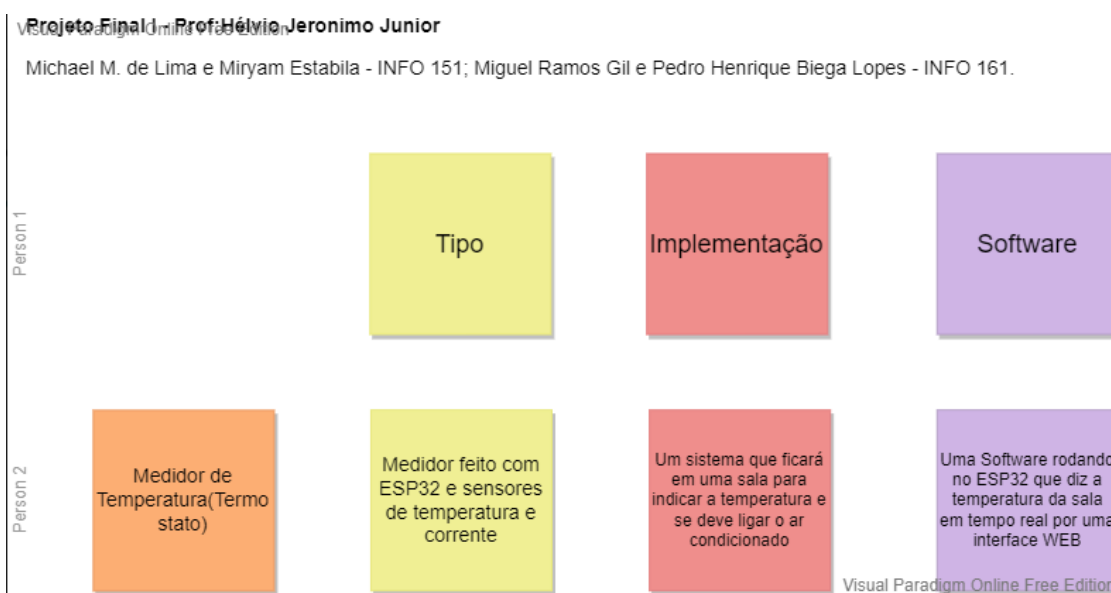
- Permitir a leitura (medidor) da temperatura das salas usando sensores DHT11, DHT22 e SCT-013
- Permitir o controle do ar condicionado através do envio de informações para executar ações de aumento, acionamento e diminuição usando um transmissor infravermelho HX-53.

1.3 Metodologia

Como esse projeto foi iniciado no período 2022.1 pelos alunos. Inicialmente o projeto era um “2 em 1” com uma Sirene e um Termostato para a entrega no final do curso. Porém com a reprovação de Michael no 5º período, houve uma divisão do projeto. A Sirene ficou com Pedro e Miguel e o termostato com Michael que se juntou a Miryam para realizá-lo. A metodologia aplicada é a de desenvolver a ideia dentro das aulas de projeto final I e assim desenvolver um protótipo de um produto junto de um documento. Os passos dados foram o desenvolvimento da ideia com o grupo; a pesquisa dos componentes a serem usados, com orientações do prof. Luiz Felipe e usando as aulas de Hardwares Programáveis I e II como base; escolha dos componentes; a compra dos componentes; prototipação.

1. Ideação: A Necessidade de monitorar e de controlar a temperatura das salas de aula em dias de calor para que o ar condicionado seja ligado ou não, evitando o desperdício de energia, desgaste do aparelho e problemas de saúde de todos do campus devido ao calor ou ao mau uso. Também com o objetivo de controlar a temperatura das salas de aula e manter os alunos do campus em uma temperatura ideal e também, economizar energia da escola, equilibrando entre o conforto dos que estão na aula e a economia de energia da escola. Essa temperatura definida pela escola é de em média 22 a 23 graus.

— Brainstorming



— Trello



2. Definição das funcionalidades a serem implementadas: As seguintes funcionalidades a serem aplicadas serão o monitoramento e controle. No caso, monitoramento da temperatura ambiente de uma sala ou laboratório no campus; monitoramento da temperatura de saída do ar-condicionado; monitoramento da corrente e de on/off do ar-condicionado; e controle de temperatura do ar-condicionado, além da função de ligar e desligar.

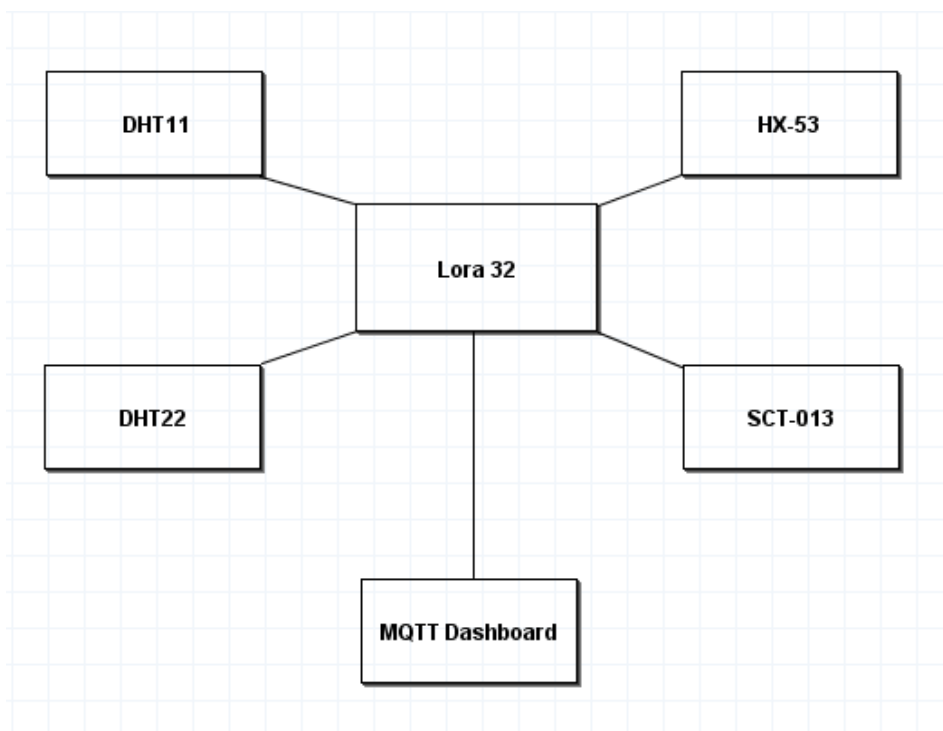
3. Prototipação: O protótipo trata-se de um conjunto dos componentes DHT11 para monitoramento da temperatura ambiente, DHT22 para monitoramento da temperatura do ar-condicionado, SCT-013 para monitoramento da corrente e de on/off do ar condicionado, um transmissor infravermelho HX-53 para controle do ar-condicionado, e um microcontrolador Heltec Lora 32 que tem uma tela onboard para mostrar as temperaturas monitoradas. A junção de todos esses componentes resulta em uma central de controle e monitoramento do ar-condicionado, a ideia é juntar todos esses componentes em uma case. Óbvio, nem todos os componentes será possível ficarem juntos, já que o sensor de temperatura que vai monitorar o ar-condicionado, não pode está junto do sensor de temperatura que irá monitorar a temperatura ambiente, porém o circuito que vai fazer a conexão de todos os componentes e os microcontrolador precisa necessariamente estarem juntos em uma central. Até para uma questão de organização.

4. Configuração do Ambiente de Desenvolvimento: Os ambientes de trabalho do projeto incluem alguns programas como o Arduino IDE, Wokwi, TinkerCad e o Ultimaker cura como plataformas de desenvolvimento do projeto. Basicamente o Arduino IDE é um ambiente e desenvolvimento para programação em linguagem C e C++ para placas de desenvolvimento como o arduino, esp8266, esp32, lora 32, etc. Nessa IDE possuem diversas bibliotecas com foco em inúmeras aplicações, mas no caso desse projeto usaremos bibliotecas com foco em IoT e Suporte a componentes. O Wokwi é um simulador online de diversas placas de desenvolvimento, incluindo o esp32 e que sua IDE é similar ao Arduino IDE. O TinkerCad além de ser um simulador online com foco em arduino, ele tem diversos recursos e entre eles um de modelagem 3D. De fato é básico, porém é o suficiente para o projeto. E o Ultimaker cura é um software de fatiamento 3D que serve para importar um projeto 3D e converter ou adaptar o projeto para impressão.

5. Implementação das Funcionalidades: Usaremos o protocolo MQTT para enviar as informações para um broker e usaremos o MQTT dashboard como interface de monitoramento e controle do produto. Teremos a relação de publisher e subscribe entre os sensores de monitoramento e controle que estarão ligados ao esp32 e que terão suas informações exibidas no MQTT dashboard. O controle será feito da mesma forma, só que o inverso, haverá botões na interface de usuário simulado os botões de um controle de ar-condicionado que quando acionado, enviará o comando para o transmissor infravermelho HX-53 para determinar alguma ação do ar-condicionado.

6. Implementação de Melhorias: As melhorias poderão ser broker MQTT, que podemos implementar um próprio do campus como proposto pelo orientador do projeto. Também podemos melhorar na questão de, invés de depender do conjunto MQTT como um todo (Protocolo MQTT e MQTT Dashboard), mas sim, usar uma interface web com HTML e CSS. É uma solução opcional caso não der certo o conjunto MQTT.

- Modelo Conceitual do projeto:



2. TÍTULO DO TEMA DO PROJETO

2.1. Descrição Geral da Proposta

A proposta do termostato é ser uma central de monitoramento e controle da temperatura de uma sala ou laboratório do campus. Esse monitoramento e controle será feito pela internet usando tecnologias IoT como o protocolo MQTT que é compatível com o Lora 32. A linguagem de programação a ser utilizada é o C++ usando uma IDE (Integrated Development Environment) chamada Arduino IDE, que dá suporte além dessa linguagem, dá suporte a bibliotecas que iremos utilizar como DHT sensor library, ESP8266 and ESP32 OLED driver for SSD1306 displays, WiFi, PubSubClient, IRremote e drivers como USB Silicon Labs CP210x para UART Bridge VCP que dão suporte ao Lora 32 e os todos os componentes que utilizaremos no protótipo como o DHT11, DHT22 e SCT-013, que são sensores de monitoramento. E o transmissor infravermelho HX-53, que transmite informações que

também podem ser comandos. Todos esses componentes serão interligados em uma protoboard usando jumpers e usaremos alguns resistores de 10kΩ para os sensores de monitoramento descritos acima e também no Transmissor infravermelho HX-53. Tudo isso estará dentro de um gabinete/case que será impresso em 3D usando tecnologias como o thingiverse, tinkercad e o ultimaker cura. O primeiro é um site de projetos em 3D para usarmos como base para a construção da case, o segundo é um software de modelagem 3D web para modelar o projeto, e o terceiro é um software de fatiamento 3D usando para imprimir de fato o projeto em uma impressora 3D.

A ideia é desenvolver uma interface no mqtt dashboard, usando os protocolos mqtt para se comunicar ao broker mqtt para podemos controlá-los remotamente usando o Lora 32 com os componentes descritos no último parágrafo. O layout do aplicativo será da seguinte forma, lá haverá 3 caixas de exibição que mostram: 1. Temperatura ambiente, 2. Temperatura do ar-condicionado, 3. Corrente do ar-condicionado. Depois, haverá 3 botões: 1. Liga e desliga, 2. Aumentar a temperatura, 3. Diminuir temperatura. Cada caixa de exibição e cada botão haverá um tópico definido dentro do código que será colocado no Lora 32 para que o aplicativo possa se comunicar com o Lora 32. Esses tópicos podemos chamar de protocolos MQTT, pois, através deles que a informação é enviada ao um broker, que é o chamado publisher. E também que o aplicativo possa receber essa informação que está no broker, que é chamado de subscriber. Também acontece o caminho contrário deste tráfego de informação, que aí no caso o publisher será o aplicativo e o subscriber o Lora 32, geralmente usado para comandos de controle e atribuídos a botões no aplicativo.

2.2. Lista de Requisitos (Funcionalidades) previstas

ID Requisito	Descrição
REQ1 - Medir a temperatura ambiente	A Temperatura ambiente será medida através de um Sensor DHT11 conectado ao Lora 32 por jumps junto de resistor de 10kΩ. O resistor e o Lora32 estarão conectados em uma protoboard
REQ2 - Medir a temperatura do	A temperatura do ar-condicionado vai ser medida através de um Sensor DHT22, que tem maior precisão se comparado ao DHT22, conectado ao Lora 32 por jumps junto de resistor de 10kΩ. O resistor e o Lora 32 estarão conectados em uma protoboard. O sensor estará posicionado na frente do

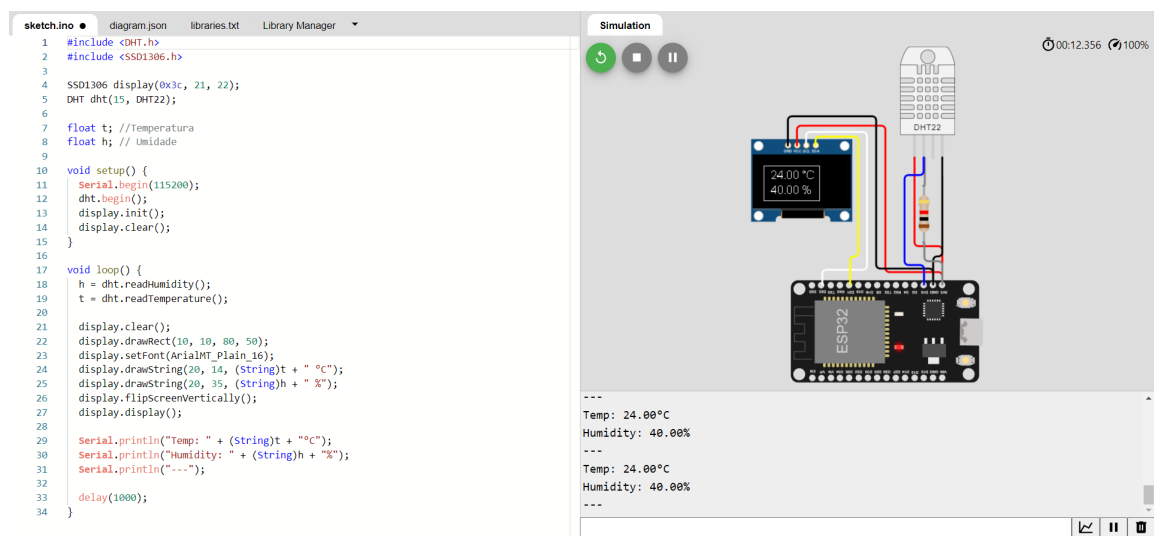
ar-condiciona do	duto de ar do ar-condicionado para que possa monitorar a temperatura do ar-condicionado.
REQ3 - Medir a Corrente do ar-condiciona do	A corrente do Ar-condicionado será medida através de um sensor chamado SCT-013 que é um sensor de corrente não invasivo, ou seja, ele vai monitorar a corrente que é consumida pelo ar-condicionado. Ele será ligado com resistores de 10kΩ (muito provavelmente) em uma protoboard e será conectado ao Lora 32 através da mesma.
REQ4 - Se conectar ao uma rede e enviar ass informações ao Broker MQTT	O Lora 32 vai se conectar a uma rede de internet e se conectar ao broker MQTT. Depois, ele pegará as informações coletadas pelos sensores descritos e enviará as informações para o Broker MQTT através de um tópico. No aplicativo MQTT Dashboard, ele pegará as informações que estão sendo enviadas para aquele tópico e vai exibir em uma interface. Nessa interface haverá informações de monitoramento como: temperatura ambiente, temperatura do ar-condicionado, corrente utilizada pela o ar-condicionado.
REQ5 - O controle do ar-condiciona do	Usando um transmissor infravermelho HX-53, serão transmitidos sinais em infravermelho que vai emular os botões do ar-condicionado. O comando será dado pelo aplicativo MQTT dashboard e esse comando será enviado a partir de um tópico MQTT para o Lora 32 que vai receber e executar o comando para o transmissor infravermelho HX-53.

2.3. Processo de Prototipagem

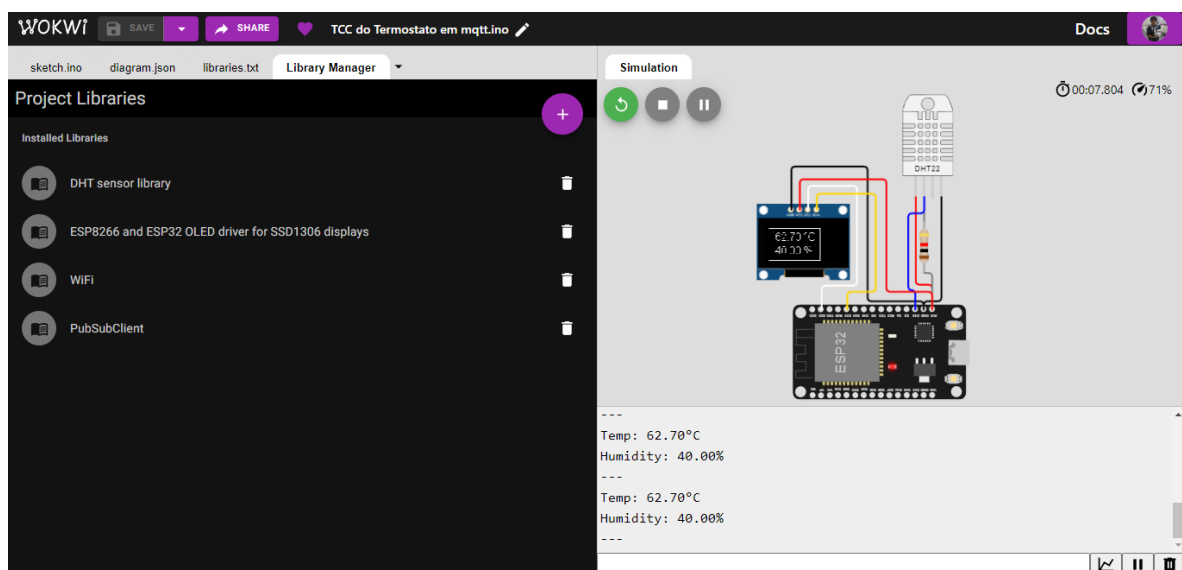
2.3.1 Primeiro protótipo do projeto:

Os testes foram feitos inicialmente pelo simulador Wokwi, como mostrado na imagem abaixo, esse teste abaixo é somente para a verificação e entendimento do funcionamento de uma parte do projeto. No caso um sensor dht22 mostrando a temperatura coletada do “ambiente”. A ideia é pegar a essa informação e jogar para o broker MQTT para ser exibido no MQTT dashboard, porém isso o que é demonstrado abaixo foi um primeiro “esboço” do projeto.

- Interface do Wokwi com o protótipo:



- Bibliotecas utilizadas:



2.3.2 Protótipo versão MQTT(Segundo protótipo do projeto):

Nesse protótipo, pretendemos ir mais além. No primeiro protótipo era relativamente simples, somente coletando as informações captadas pelo sensor dht22 e assim mostrando as informações de temperatura e umidade do ambiente no monitor serial e no display OLED. Usando como base o primeiro protótipo, foi desenvolvido o segundo protótipo com um código que se conecta a uma rede Wi-Fi com internet e que faz o envio das informações captadas pelo sensor dht22 para um broker MQTT, essa ação é chamada de publisher e assim podendo ser coletadas através de um subscriber. No código é mostrado que o envio das informações é feito através de um tópico MQTT e usando um aplicativo como o MQTT dashboard, criamos uma interface onde podemos colocar o tópico em uma caixa de exibição do aplicativo para justamente exibir as informações enviadas.

- Código do segundo protótipo:

```
1 //Bibliotecas
2 #include <WiFi.h>
3 #include <PubSubClient.h>
4 #include <ESPmDNS.h>
5 #include <DHT.h>
6 #include <SSD1306.h>
7
8 //Net Setup
9 #define NET_SSID "Wokwi-GUEST"
10 #define NET_PASSWORD ""
11
12 //MQTT Setup
13 #define MQTT_ID "tcctermostato-esp32-0800313131"
14 #define MQTT_BROKER "broker.hivemq.com"
15 #define MQTT_PORT 1883
16 #define MQTT_MILLIS_TOPIC "tcc_termostato_millis"
17 #define MQTT_DHT_TEMP_TOPIC "tcc_termostato_temp"
18 #define MQTT_DHT_HUMD_TOPIC "tcc_termostato_humd"
19
20 //Cliente de rede
21 WiFiClient espClient;
22
23 //Cliente MQTT
24 PubSubClient MQTT(espClient);
25
26 //Variáveis que armazenam as informações que são enviadas ao broker
27 char millis_str[10] = "";
28 char temp_str[10] = "";
29 char humd_str[10] = "";
30
31 //Display
32 SSD1306 display(0x3c, 21, 22);
33
34 //Sensor DHT22
35 DHT dht(15, DHT22);
36
37 //Temperatura
38 float t;
39
40 //Umidade
41 float h;
42
43
44 void setupWifi() {
45
46 //Configura a conexão à rede sem fio
47 if (WiFi.status() == WL_CONNECTED)
48 | return;
49
50 Serial.println();
51 Serial.print("Connecting to ");
52 Serial.println(NET_SSID);
53
54 WiFi.begin(NET_SSID, NET_PASSWORD);
55
56 while (WiFi.status() != WL_CONNECTED) {
57 | delay(500);
58 | Serial.print(".");
59 }
60 }
```

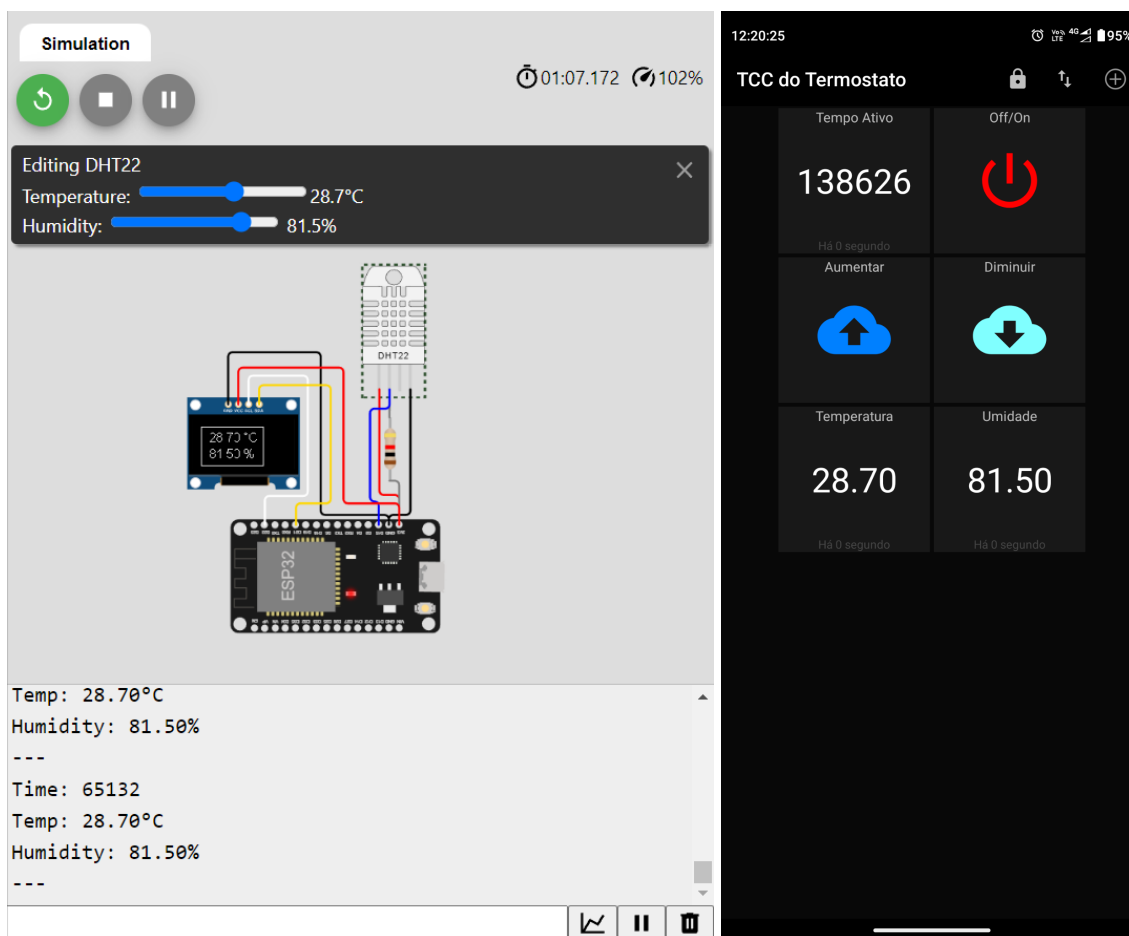
```

61   Serial.println("");
62   Serial.println("Wifi connected");
63   Serial.println("IP address: ");
64   Serial.println(WiFi.localIP());
65 }
66
67
68 void setupMQTT() {
69
70   //Informa qual broker e porta deve ser conectado
71   MQTT.setServer(MQTT_BROKER, MQTT_PORT);
72
73   while (!MQTT.connected())
74   {
75       Serial.print("**Tentando se conectar ao Broker MQTT: ");
76       Serial.println(MQTT_BROKER);
77       if (MQTT.connect(MQTT_ID))
78       {
79           Serial.println("Conectado com sucesso ao broker MQTT!");
80       }
81       else
82       {
83           Serial.println("Falha ao reconectar no broker.");
84           Serial.println("Havera nova tentativa de conexao em 2s");
85           delay(2000);
86       }
87   }
88 }

90 void setup(void) {
91
92   //Configura o baudrate da comunicação serial
93   Serial.begin(115200);
94
95   dht.begin();
96   display.init();
97   display.clear();
98
99   setupWifi();
100
101   setupMQTT();
102 }
103
104 void loop(void) {
105
106   //Strings de temperatura e umidade
107   h = dht.readHumidity();
108   t = dht.readTemperature();
109
110   //Configuração do Display
111   display.clear();
112   display.drawRect(10, 10, 80, 50);
113   display.setFont(ArialMT_Plain_16);
114   display.drawString(20, 14, (String)t+ " °C");
115   display.drawString(20, 35, (String)h+ " %");
116   display.flipScreenVertically();
117   display.display();
118
119   //Monitor Serial
120   Serial.print("Time: ");
121   Serial.println(millis_str);
122   Serial.println("Temp: " + (String)t+ "°C");
123   Serial.println("Humidity: " + (String)h+ "%");
124   Serial.println("---");
125
126   //Envia Segundos
127   sprintf(millis_str, "%d", millis());
128   MQTT.publish(MQTT_MILLIS_TOPIC, millis_str);
129
130   //Envia Temperatura
131   sprintf(temp_str, "%s", (String)t);
132   MQTT.publish(MQTT_DHT_TEMP_TOPIC, temp_str);
133
134   //Envia Umidade
135   sprintf(humd_str, "%s", (String)h);
136   MQTT.publish(MQTT_DHT_HUMD_TOPIC, humd_str);
137
138   //Configuração de Setup e Delay
139   setupWifi();
140   setupMQTT();
141   delay(1000);
142 }

```

- Interface de controle e monitoramento no MQTT dashboard:

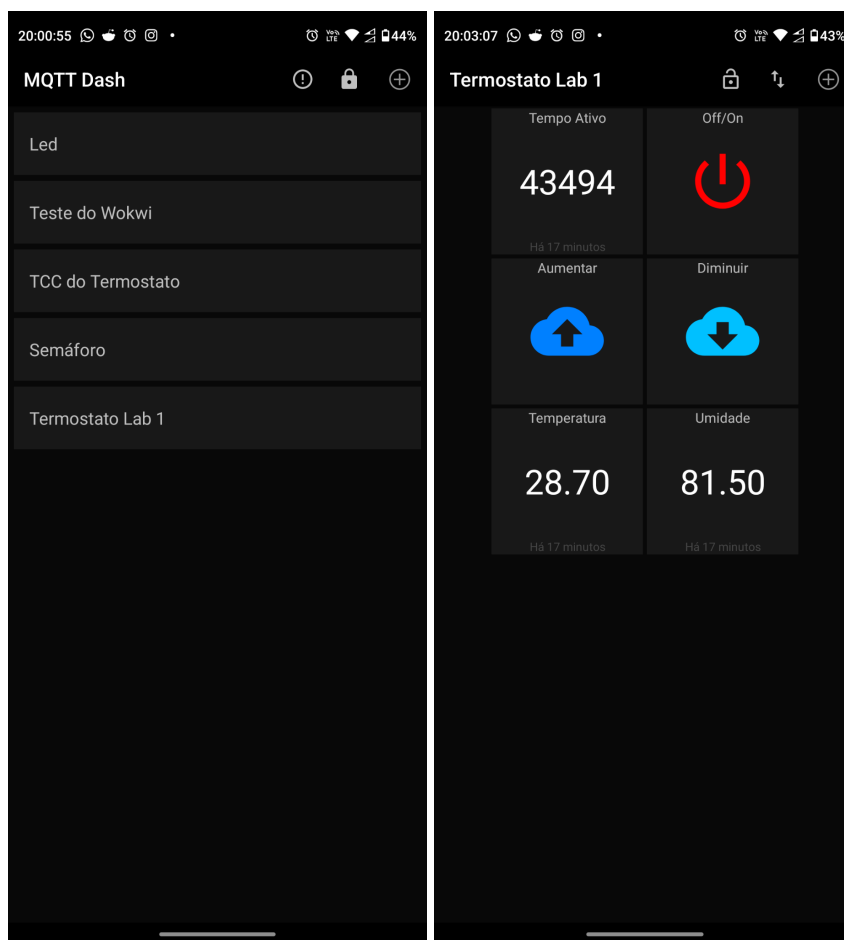


2.3.3 Configuração do MQTT Dashboard:

A configuração do MQTT Dashboard consiste em criar uma aba para onde ficará a sua interface e durante a criação dessa aba você deve colocar o endereço e a porta do broker MQTT que está sendo utilizado. No caso deste protótipo o endereço “broker.hivemq.com” com a porta “1883”. Depois disso deve-se criar botões que serão as caixas de exibição para cada tópico MQTT definido no código, nesse caso, um para o Tempo, um para a Temperatura e um para Umidade. Em breve a ideia é desenvolver mais 3 botões, um para ligar e desligar o ar-condicionado, um para aumentar a temperatura e um para diminuir. Tudo isso utilizando de tópicos MQTT que farão alguma ação definida no código que estará dentro do Lora 32 que irá comandar o HX-53 que é o transmissor infravermelho. Irei estar disponibilizando um documento com o passo-a-passo com imagens e instruções de configuração do aplicativo MQTT Dashboard. Abaixo segue o link desse documento e algumas imagens da interface desenvolvida.

Link do documento: <<Documento ainda a ser redigido e disponibilizado>>

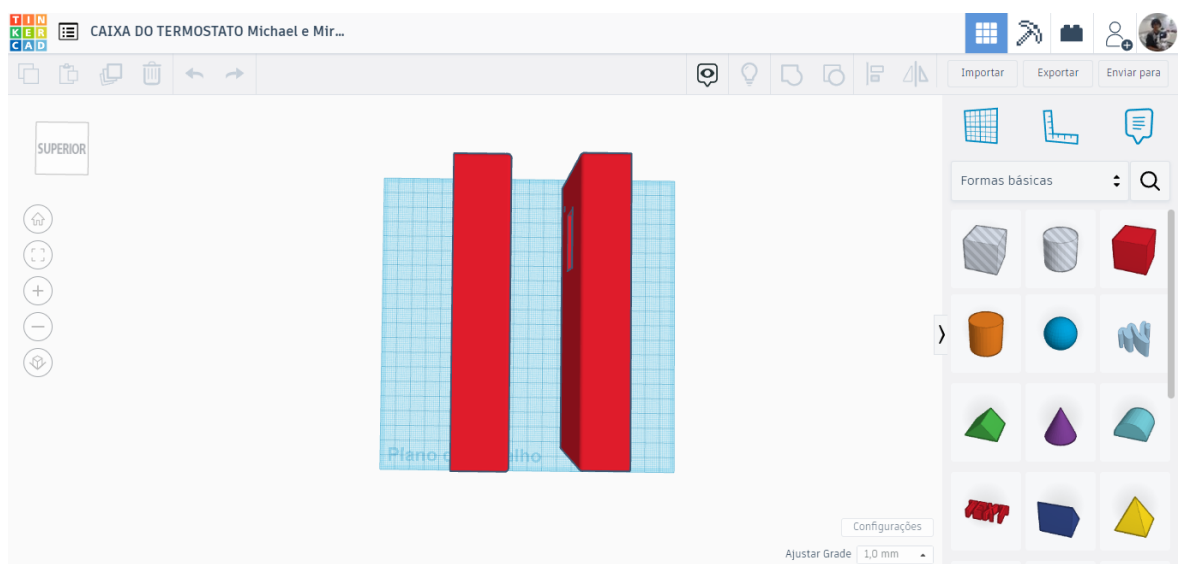
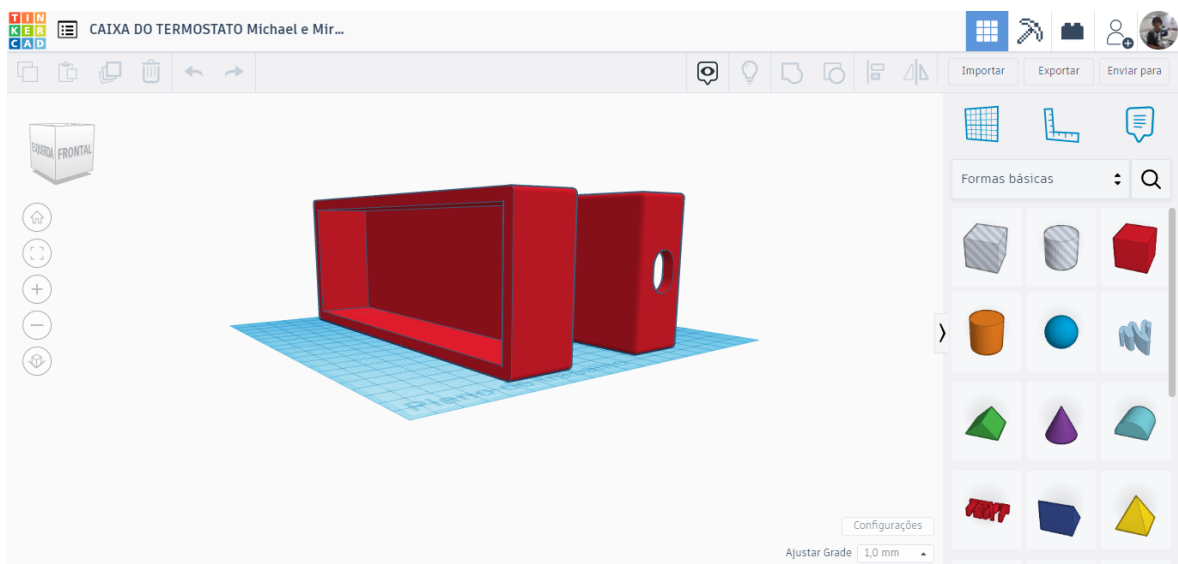
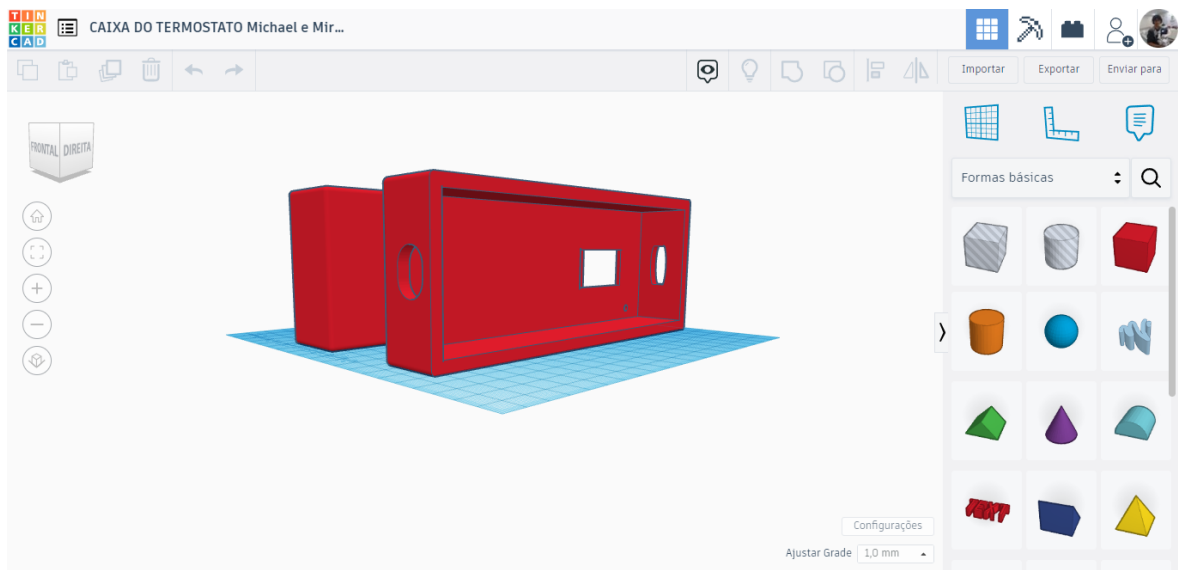
- Interface do aplicativo:



2.3.4 Protótipo do gabinete impresso em 3D:

O gabinete foi desenvolvido por Michael Lima e Pedro Henrique para comportar todos os componentes necessários. Que são a protoboard, o Lora 32, sensores dht11, dht22, sct-013 e HX-53. Ele foi desenvolvido em uma solução web chamada tinkercad em uma aba do programa específica para modelagem 3D. Suas respectivas medidas são: 17cm de largura, 6cm de profundidade e 6cm de altura. Ela se abrirá ao meio, onde estará comportando seus componentes de dará acesso aos mesmos. Abaixo estão algumas imagens do modelo em 3D.

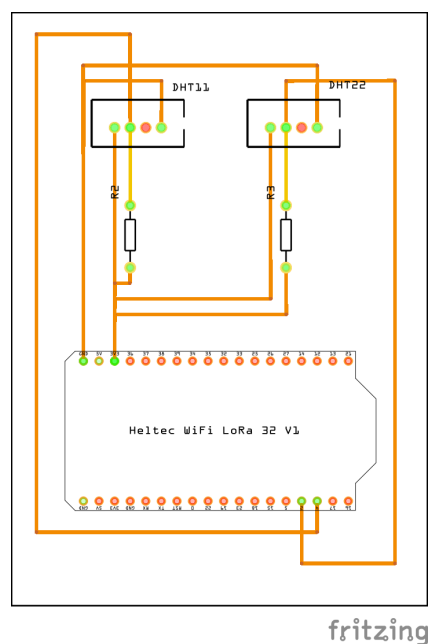
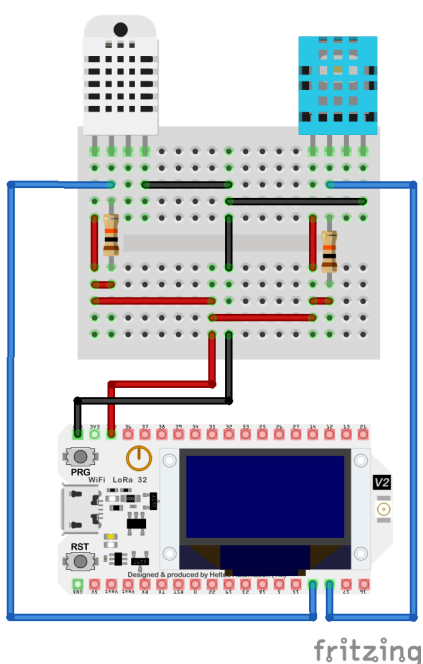
- Modelo 3D no Tinkercad:

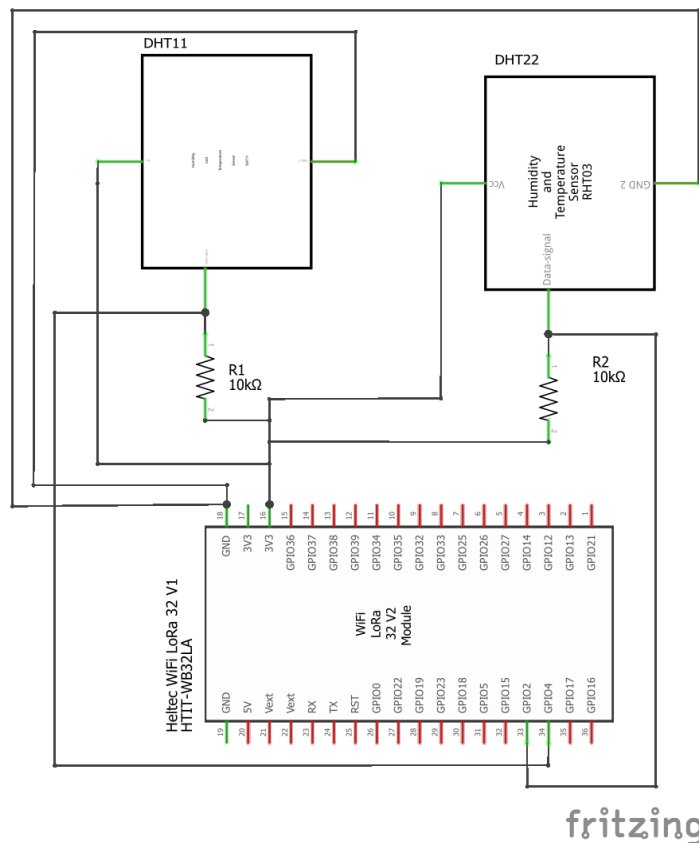


2.3.5 Protótipo Físico:

Abaixo está uma parte do circuito eletrônico que haverá nesse termostato, as imagens abaixo são: O circuito eletrônico no simulador Fritzing, O Esquema eletrônico e o PCB do circuito. Por enquanto, nesse protótipo abaixo só tem dois sensores de monitoramento que são o DHT11 e DHT22 com 2 resistores de 10k Ω , um para cada sensor. Isso conectado à uma protoboard que está conectada no Lora 32. Logo abaixo há um protótipo que tem os componentes descritos anteriormente com a adição de transmissor infravermelho HX-53 que ainda não está sendo utilizado. Assim como no Wokwi, esses esquemas abaixo e as fotos do protótipo abaixo são só um “esboço”, tendo muita coisa que será mudada e melhorada, mas mantendo a ideia central e os componentes.

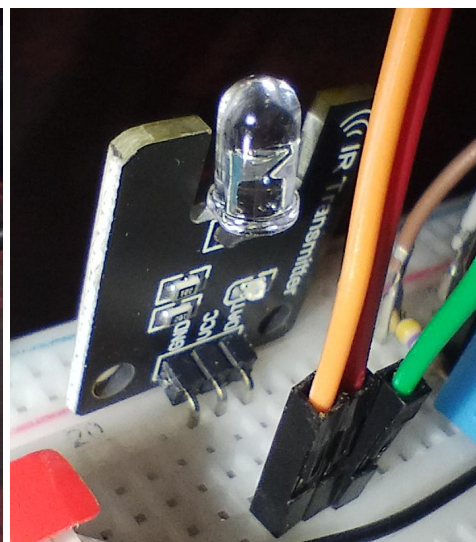
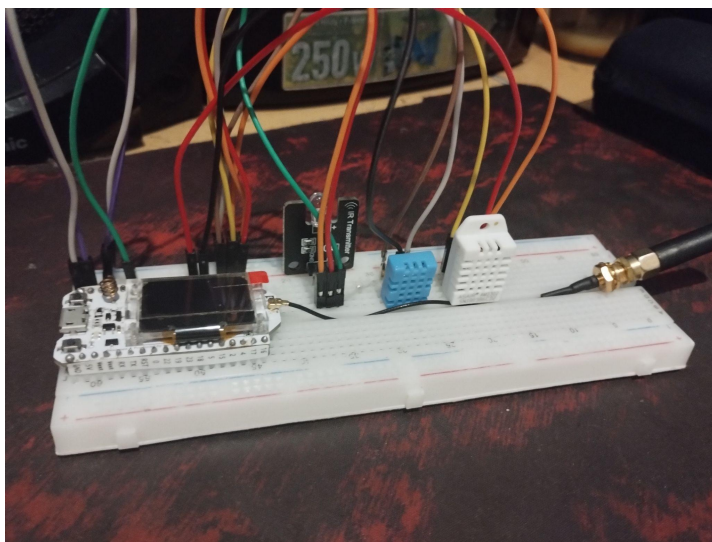
- Circuitos Eletrônicos:





fritzing

- Primeiro protótipo físico do projeto com um Lora 32 e com o transmissor infravermelho HX-53:



Referências aos protótipos: TCC DO TERMOSTATO. Michael Lima. GitHub. Disponível em: <<<https://github.com/MichaelLimaDeveloper/TCC-do-Termostato>>>, acesso feito em 18/12/2022.

2.4. Orçamento do Projeto

Basicamente abaixo é uma tabela com todos os gastos com o projeto até agora. O valor de cada produto e o subtotal de todos. Todos os links são de onde realmente foram comprados a maioria dos itens, não incluindo o frete caso o produto tenha sido comprado pela internet. Aqui foi contando o valor de cada componente e não o valor bruto com a soma de frete dos itens ou reposição de componentes.

Descrição (c/link)	Qtd	Valor unit.	Subtotal	Situação
Heltec Lora 32	1	R\$ 92,21	R\$ 92,21	Comprado
Cabo USB	1	R\$ 6,66	R\$ 98,87	Comprado
Protoboard	1	R\$ 17,39	R\$ 116,26	Comprado
Resistor 10K (conjunto com 10)	1	R\$ 1,09	R\$ 117,35	Comprado
Sensor De Temperatura E Umidade Dht11 Arduino	1	R\$ 8,98	R\$ 126,33	Comprado
Sensor De Temperatura E Umidade Dht22 Arduino	1	R\$ 29,98	R\$ 156,31	Comprado
Sensor De Corrente Dc Não Invasivo Sct-013	1	R\$ 35,00	R\$ 191,31	Comprado
Kit transmissor e receptor infravermelho HX-53 e Ky-022	1	R\$ 7,25	R\$ 198,56	A Caminho

3. REFERÊNCIAS BIBLIOGRÁFICAS

3.1. Fontes das referências em sites, fóruns e artigos na internet

CONHECENDO A PLACA WIFI LORA ESP32 (433MHZ / 868MHZ / 915MHZ). Euler Oliveira. Master Walker. Disponível em:

<<<https://blogmasterwalkershop.com.br/embarcados/esp32/conhecendo-a-placa-wifi-lora-esp32-433mhz-868mhz-915mhz>>>, acesso feito em 18/11/2022.

PROJETO ESP32 COM DISPLAY OLED E SENSOR AM2302 DHT22 MEDINDO TEMPERATURA E UMIDADE. Usinainfo. Disponível em

<<<https://www.usinainfo.com.br/blog/projeto-esp32-com-display-oled-e-sensor-am2302-dht22-medindo-temperatura-e-umidade/>>>, acesso feito em 18/11/2022.

ESP32 LAB 01 – DISPLAY OLED, SENSOR DHT22 E HORA VIA NTP SERVICE.

Easytromlabs. Disponível em:

<<[ESP32 WITH DHT11/DHT22 TEMPERATURE AND HUMIDITY SENSOR USING ARDUINO IDE. Alan Miller. Random nerd tutorials. Disponível em:](https://easytromlabs.com/esp32/esp32-lab-01-display-oled-sensor-dht22-e-hora-via-ntp-service/?print=print#:~:text=O%20sensor%20de%20umidade%20e,5%20%25%20de%20acertos%20na%20leitura.>>, acesso feito em 18/11/2022.</p></div><div data-bbox=)

<<<https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-sensor-arduino-ide/>>>, acesso feito em 18/11/2022.

LUIZSTATION MQTT PUBLISHER. Luiz Felipe. GitHub. Disponível em:

<<<https://github.com/luizoliveira/arduinoExamples/blob/main/LuizStationMQTTPublisher/LuizStationMQTTPublisher.ino>>>, acesso feito em 28/11/2022.

LUIZSTATION MQTT TEMPDHT11. Luiz Felipe. GitHub. Disponível em:

<<<https://github.com/luizoliveira/arduinoExamples/blob/main/LuizStationMQTTTempDHT11/LuizStationMQTTTempDHT11.ino#L5>>>, acesso feito em 07/12/2022.

3.2. Fontes das referências em vídeos na internet

COMO FAZER CONTROLE REMOTO DE TV COM ARDUINO PRO MINI. Marlon Nardi. Disponível em:

<<https://www.youtube.com/watch?v=-1SKYm74tjY&ab_channel=MarlonNardi>>, acesso feito em 28/11/2022.

ESP32 LAB 01 – DISPLAY OLED, SENSOR DHT22 E HORA VIA NTP SERVICE. EasyTrom Labs. Disponível em:

<<https://www.youtube.com/watch?v=843zY-stQaw&ab_channel=EasyTromLabs>>, acesso feito em 07/12/2022.

RECEPTOR INFRAVERMELHO (IR) - CURSO DE ARDUINO #41.

Marlon Nardi. Disponível em:

<<<https://www.youtube.com/watch?v=sax7Mcfbj7I&t=781s>>>, acesso feito em 20/12/2022.