

Instituto Federal de Educação, Ciência e  
Tecnologia do Rio de Janeiro

*Campus* Niterói

Técnico em Informática Integrado ao Ensino  
Médio

GABRIEL SALES RODRIGUES

MICHAEL MONTEIRO DE LIMA

MIRYAM ESTÁBILA NOGUEIRA DE SOUZA

**IF Resources:**  
**Controle e Monitoramento**  
**de Temperatura e Recursos**  
**Hídricos**

NITERÓI

2023.1

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

IF RESOURCES:  
CONTROLE E MONITORAMENTO DE TEMPERATURA E  
RECURSOS HÍDRICOS

Trabalho de Conclusão de Curso submetido  
ao corpo docente do Instituto Federal de Edu-  
cação, Ciência e Tecnologia do Rio de Janeiro  
como requisito parcial para a obtenção do  
grau de Técnico em Informática.

Orientador: Prof. DSc. Luiz Felipe da Silva  
Oliveira

Coorientador: Prof. DSc. Alexandre Domin-  
gues Gonçalves

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

**IF RESOURCES: CONTROLE E MONITORAMENTO DE  
TEMPERATURA E RECURSOS HÍDRICOS**

Trabalho de Conclusão de Curso submetido  
ao corpo docente do Instituto Federal de Edu-  
cação, Ciência e Tecnologia do Rio de Janeiro  
como requisito parcial para a obtenção do  
grau de Técnico em Informática.

**BANCA EXAMINADORA**

---

**Prof. DSc. Luiz Felipe da Silva Oliveira**  
Orientador/IFRJ

---

**Prof. DSc. Alexandre Domingues  
Gonçalves**  
Co-Orientador/IFRJ

Dedicamos este trabalho primeiramente a Deus, que nos concedeu a força e a sabedoria necessárias para vencer todos os obstáculos e chegar até aqui. Dedicamos também aos nossos amigos, pelos momentos de descontração que ajudaram a aliviar o estresse e a tensão durante essa caminhada acadêmica. Dedicamos esse trabalho também um ao outro, por não nos abandonarmos durante os 6 longos períodos letivos cursados.

# AGRADECIMENTOS

Ao Instituto Federal de Educação, Ciência e Tecnologia do Rio de Janeiro - Campus Niterói pela oportunidade de adquirir conhecimento por meio de atividades acadêmicas que proporcionam conhecimento prático. Aos docentes Alexandre Domingues Gonçalves, Luiz Felipe Silva Oliveira, Marcelo Japiassu Ramos, Samuel Max Gabbay e Vitor Gabriel de Souza Alencar por nos orientar não só no ponto de vista técnico do projeto, mas também no apoio e ensinamentos passados. Aos nossos amigos Jessyca Nalin Meireles, João Pedro Silverio Gama, Luiz Gabriel Pinheiro, Pedro Henrique Biega Lopes Bandeira, Roger Francin Queiroz do Nascimento, Samuel Asaph Lima Dias e Tainá Petrucio Fonseca por nos ajudarem, por nos apoiarem e por sempre estarem com a gente quando a gente mais precisou.

# RESUMO

O trabalho de conclusão de curso: “IF Resources: controle e monitoramento de temperatura e recursos hídricos”, busca tornar prática a fiscalização da temperatura das salas e laboratórios do campus, além dos níveis de água das caixas. O projeto visa o aprimoramento e inserção da eletrônica, tecnologias IoT e a Pesquisa e Extensão no dia a dia o IFRJ – Campus Niterói. O público-alvo do projeto são os alunos, professores e equipe de manutenção do campus, com o objetivo de promover o bem-estar de todos. Além disso, o projeto visa alertar os usuários sobre a temperatura antes mesmo de entrarem nas salas, evitando assim problemas de saúde causados por choques térmicos, por exemplo. Também visa alerta o nível d’água do campus, enviando assim desperdício ou escassez de água. A importância desse projeto reside na facilidade de controle da temperatura dos aparelhos de ar-condicionado, no monitoramento do nível d’água e na supervisão da temperatura e umidade das salas. A ideia é prevenir problemas de saúde decorrentes de dias quentes, uso inadequado dos aparelhos e escassez de água. Além disso, o projeto contribui para evitar o desperdício de energia no campus, o desgaste dos equipamentos e a falta de um recurso essencial como a água.

**Palavras-chave:** Eletrônica; IoT; Lora 32; Prototipação.

# ABSTRACT

This document presents the project: “IF Resources: control and monitoring of temperature and water resources”, seeks to make it practical to monitor the temperature of the rooms and laboratories on the campus, in addition to the water levels in the tanks. The project aims at the improvement and insertion of electronics, IoT technologies and Research and Extension in the daily life of the IFRJ - Campus Niterói. The project’s target audience is students, teachers and the campus maintenance team, with the aim of promoting the well-being of all. In addition, the project aims to alert users about the temperature even before entering the rooms, thus avoiding health problems caused by thermal shocks, for example. It also aims to alert the water level of the campus, thus sending waste or shortage of water. The importance of this project lies in the ease of controlling the temperature of the air conditioners, monitoring the water level and supervising the temperature and humidity in the rooms. The idea is to prevent health problems arising from hot days, inappropriate use of devices and water shortages. In addition, the project helps to avoid wasting energy on the campus, equipment wear and tear and the lack of an essential resource such as water.

**Keywords:** electronics; Lora 32; IoT and prototyping.

# LISTA DE FIGURAS

1	Interface do Wokwi . . . . .	16
2	Bibliotecas Wokwi . . . . .	17
3	Interface do Wokwi e do MQTT Dashboard . . . . .	18
4	Fotos do Primeiro Prototipo Físico . . . . .	19
5	Fotos do Segundo Prototipo Físico . . . . .	20
6	Fotos do Terceiro Prototipo Físico . . . . .	20
7	Fotos do Quarto Prototipo Físico . . . . .	21
8	Desenho do Circuito . . . . .	22
9	Diagrama do Circuito . . . . .	23
10	Esquemático do Circuito . . . . .	23
11	Imagem da primeira case modelada em 3D do projeto . . . . .	24
12	Imagem da case atual do projeto modelada em 3D . . . . .	24
13	Imagem da PCB que substituiria a Protoboard . . . . .	25
14	Imagem sobre o protocolo MQTT e seu padrão Publish-Subscribe . . .	26
15	Trecho de Código onde está definido parâmetros do protocolo MQTT .	27
16	Trecho de Código onde está definido objetos, strings e variáveis . . . .	27
17	Imagem onde mostra a biblioteca e o exemplo que foi retirado o funcio- namento do SendRaw . . . . .	28
18	Imagem da tela inicial do site . . . . .	29
19	Imagem da pagina do formulário de feedback . . . . .	30
20	Imagem da pagina que informa o nível da caixa d'água . . . . .	30
21	Imagem da pagina que informa a temperatura do ambiente . . . . .	31
22	Imagem da pagina que informa a umidade do ambiente . . . . .	31
23	Imagem da pagina de controle do ar-condicionado . . . . .	31
24	Imagem do painel da instância EC2 que está rodando na AWS . . . . .	32
25	Imagem do painel de configuração da imagem EMQX que está rodando na AWS . . . . .	32
26	Imagem do painel de configuração do site que está hospedado na hostinger	33
27	Imagem da parte interna do equipamento . . . . .	33
28	Imagem da parte superior do equipamento . . . . .	34
29	Imagem da conexão do sensor de tempereatura e umidade DHT22 no Lora 32 . . . . .	34
30	Imagem da conexão do transmissor infra-vermelho HX-53 no Lora 32 .	35
31	Representação do funcionamento do nível de água . . . . .	36
32	Imagem da conexão do sensor de nível de água Diy 4 no Lora 32 . . . .	36
33	Imagem do Equipamento Finalizado . . . . .	37



# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>10</b>
<b>1.1</b>	<b>Justificativa . . . . .</b>	<b>11</b>
<b>1.2</b>	<b>Objetivo. . . . .</b>	<b>11</b>
1.2.1	Objetivo Geral. . . . .	11
1.2.2	Objetivos Específicos . . . . .	11
<b>1.3</b>	<b>Metodologia . . . . .</b>	<b>12</b>
<b>2</b>	<b>DESENVOLVIMENTO DO PROJETO . . . . .</b>	<b>14</b>
<b>2.1</b>	<b>Descrição Geral da Proposta . . . . .</b>	<b>14</b>
<b>2.2</b>	<b>Lista de Requisitos . . . . .</b>	<b>15</b>
<b>2.3</b>	<b>Processo de Prototipagem . . . . .</b>	<b>16</b>
2.3.1	Primeiros Protótipos no Simulador . . . . .	16
2.3.2	Primeiros Protótipos Físicos . . . . .	19
2.3.3	Circuito Eletrônico . . . . .	22
2.3.4	Desenho 3D e Desenho da PCB . . . . .	24
<b>3</b>	<b>IMPLEMENTAÇÃO FINAL . . . . .</b>	<b>26</b>
<b>3.1</b>	<b>Passo a Passo . . . . .</b>	<b>26</b>
3.1.1	Passo 1 - Programação do hardware . . . . .	26
3.1.2	Passo 2 - Desenvolvimento do Site . . . . .	28
3.1.3	Passo 3 - Construção do hardware . . . . .	33
3.1.3.1	Sensor de temperatura e umidade DHT22 . . . . .	34
3.1.3.2	Transmissor Infra-vermelho HX-53 . . . . .	35
3.1.3.3	Sensor de Nível de Água Diy 4 (LC1BD04) . . . . .	35
<b>3.2</b>	<b>Produto Final . . . . .</b>	<b>37</b>
<b>4</b>	<b>CONSIDERAÇÕES FINAIS . . . . .</b>	<b>38</b>
<b>5</b>	<b>REFERÊNCIAS . . . . .</b>	<b>39</b>

# 1 INTRODUÇÃO

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. Além disso, também há um outro problema que afeta o campus que é a falta d'água. Vários alunos e funcionários do mesmo reclamaram sobre esse problema.

Visando solucionar esse problema, alguns alunos do semestre 2022.2 começaram a idealizar uma solução para tentar minimizar o problema reportado acima. Portanto, as seguintes propostas de projeto foram definidas: A construção de uma central de monitoramento e controle do ar-condicionado, e ao mesmo também um nível de água para uma caixa d'água ou reservatório.

Portanto, o equipamento desenvolvido é um termostato e um nível d'água, que tem como nome "IF Resources". Nessa central de monitoramento e controle é possível: monitorar o nível d'água, monitorar a temperatura ambiente de uma sala e o controlar o ar-condicionado. Isso é feito utilizando protocolo MQTT, já que permite envio e recebimento de informações.

Usaremos componentes como o Heltec Lora 32, um microcontrolador voltado para IoT, juntamente de um Sensor de Temperatura e Umidade e um transmissor infravermelho. Tudo isso conectado em uma PCB feita exclusivamente para o projeto e por fim, acoplado dentro de uma case impresso utilizando uma impressora 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 site 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, facilitar e proporcionar monitoramento de nível d'água e facilitar o monitoramento da temperatura e umidade dos ambientes de sala. 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 e/ou a falta d'água. Evitar o desperdício de energia do campus, desgastes do aparelho e falta de um recurso essencial como a água.

## 1.1 Justificativa

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. Também há a necessidade de monitorar o nível d'água do campus, já que frequentemente os alunos sofrem com a falta d'água. Então, um nível d'água via web traz uma praticidade no monitoramento que possam saber a hora certa de abastecer ou economizar a água do campus.

## 1.2 Objetivo

O objetivo do projeto é oferecer um equipamento que é um medidor de temperatura de ambiente, um controle de ar-condicionado e um nível d'água reunidos em sistema, ou seja, uma central de monitoramento e controle.

### 1.2.1 Objetivo Geral

O objetivo geral do projeto é suprir a necessidade de monitorar e controlar a temperatura das salas de aulas e também monitorar o nível d'água da instituição com o objetivo manter o bem-estar de todos, bem como prover meios de economizar água e energia da escola.

### 1.2.2 Objetivos Específicos

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

- Permitir a leitura da temperatura de um certo ambiente usando o Sensor de temperatura e umidade DHT22;
- 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;
- Permitir a leitura do nível d'água de uma caixa d'água ou reservatório de água usando o sensor eletrônico de nível d'água diy 4;
- Produzir um site que seja interface de controle e monitoramento do usuário;

- O Projeto seja fácil e prático de ser utilizado para o usuário e de simples manutenção, ou que seja simples de fazer melhorias, ou seja, upgrades no projeto;
- O Projeto seja Open Source, com todas as fontes e todo o projeto esteja postado no github dos criadores e que sirva de material para o aprendizado de outras pessoas;
- O Projeto sirva a instituição solucionando ou minimizando a sua dor, ou seja, atendendo as necessidades com a nossa solução proposta com a solução proposta.

## 1.3 Metodologia

A metodologia é um conjunto de passos e estratégias adotadas que facilitam o andamento e o desenvolvimento do projeto, tendo isso em mente, os pontos seguidos para a elaboração do projeto foram:

1. **Ideação:** A necessidade de criar um equipamento que monitora o nível de água e a temperatura de um ambiente; e que também controla um ar-condicionado via internet através de um site utilizando tecnologias IoT. Para isso utilizamos da uma técnica chamada brainstorming para organizarmos as ideias e selecionar as ideias a serem implementadas;
2. **Planejamento:** Para o acompanhamento do projeto, a definição das suas funcionalidades e suas tarefas foi feita na plataforma Trello (ATLASSIAN, 2017);
3. **Definição das funcionalidades:** As funcionalidades aplicadas são o monitoramento de água do campus, o monitoramento da temperatura e umidade do campus e o controle do ar-condicionado. No caso do controle, suas funções são: Ligar, Desligar, Aumentar a Temperatura, Diminuir a temperatura;
4. **Prototipação:** O protótipo trata-se de um conjunto dos componentes que são: Sensor de Temperatura e Umidade DHT22, Transmissor infravermelho HX-53, Sensor Eletrônico de Nível d'água Diy 4 Níveis e um microcontrolador Heltec Lora 32. A junção de todos esses componentes resulta em uma central de controle e monitoramento para o campus, a ideia é juntar todos esses componentes em uma case impressa em 3D;
5. **Configuração dos Ambientes de Desenvolvimento:** Os ambientes de trabalho do projeto são as tecnologias utilizadas para o desenvolvimento do projeto, e as que foram escolhidas pelos criadores são: Arduino IDE (ARDUINO, 2005), AWS (AMAZON WEB SERVICES INC, 2006), Docker (DOCKER INC, 2013), EasyEDA (EASYEDA INC, 2013), Figma (FIELD, 2016), Fritzing (FRITZING INC, 2014), Github (GITHUB INC, 2008), Hostinger (HOSTINGER INTERNATIONAL LTD

, 2004), TinkerCad (AUTODESK, 2011), UltiMaker Cura (ULTIMAKER, 2011), Visual Studio Code (MICROSOFT, 2015) e o Wokwi (CODEMAGIC LTD, 2019);

6. Implementação: Aplicar tudo que foi desenvolvido ao longo do projeto, aplicar todas as tecnologias utilizadas e transformar o protótipo em produto final, ou seja, o equipamento proposto pelo time de desenvolvedores do projeto em sua versão final para a defesa da ideia e do projeto. Ao final, aplicar e implementar o equipamento no campus para solucionar a dor;
7. Defesa: Para a defesa de TCC, esse projeto foi necessário a confecção do documento de TCC, a confecção de uma apresentação e a finalização do equipamento final para apresentação e demonstração de seu funcionamento e suas funcionalidades.

## 2 DESENVOLVIMENTO DO PROJETO

O desenvolvimento do projeto trata-se da aplicação de toda a metodologia dita capítulo anterior visando atender os objetivos do projeto desenvolvendo o equipamento proposto e justificando ele através da sua utilidade para solucionar a dor e as necessidades do campus na versão final do projeto. O desenvolvimento do projeto também busca mostrar todo o caminho, todos os passos e tudo que foi feito no projeto, desde sua descrição até os protótipos desenvolvidos ao longo tempo dentro do mesmo.

### 2.1 Descrição Geral da Proposta

A proposta do termostato é ser uma central de monitoramento e controle do ar-condicionado feita utilizando componentes como o Heltec Lora 32 que é uma placa de desenvolvimento que utiliza o chip Xtensa® Dual-Core 32-bit LX6, o mesmo do ESP32-S3FN8, porém com uma frequência maior de 240MHz. A placa também tem embutido o display OLED SSD1306 monocromático. Utilizamos também os componentes eletrônicos como: O Sensor de Temperatura DHT22, O Transmissor Infravermelho HX-53 e a placa de nível d'água Diy 4. Para o desenvolvimento do programa implementado no hardware foi utilizado a linguagem de programação C/C++ e o framework Arduino IDE, utilizando várias bibliotecas de suporte aos componentes, conexão à internet e protocolo MQTT.

Já o site foi desenvolvido usando tecnologias como HTML, CSS, JavaScript e Docker. No caso o HTML e CSS para a interface de usuário e o JavaScript e Docker para a comunicação com o Broker que se comunica com o hardware que está enviando informações dos sensores para o broker ou executa os comandos vindo do broker através de mensagens MQTT. Também usamos plataformas como a hostinger para a hospedagem do site e a aws para hospedar o container em uma instância criada no Amazon EC2.

O equipamento foi montado em uma protoboard de 400 pontos e acoplado dentro de uma case modelada em 3D no TinkerCad e impressa em 3D no campus com o material ABS e também utilizando o software UltiMaker Cura para efetuar o fatiamento 3D.

Os códigos do projeto são Open Source e estão disponíveis no github dos criadores do projeto para aqueles do campus que quiserem fazer manutenção ou quiserem melhorar o projeto fazendo upgrades.

## 2.2 Lista de Requisitos

- REQ1** - Deverá medir a temperatura e umidade do ambiente através de um Sensor de temperatura e umidade DHT22 conectado ao Lora 32 através de jumps em uma protoboard.
- REQ2** - Deverá controlar o ar-condicionado emulando um controle de ar-condicionado modelo Elgin Eco através de um transmissor infravermelho HX-53 conectado ao Lora 32 através de jumps e uma protoboard.
- REQ3** - Deverá monitorar o nível d'água de caixa d'água ou reservatório de água do campus através do sensor eletrônico de nível d'água diy 4.
- REQ4** - Deverá ter um site de controle e monitoramento online que tenha botões para acessar as páginas que informam o nível d'água, temperatura e umidade; e uma página com botões para cada controle do ar-condicionado: Ligar, desligar, aumentar temperatura e diminuir temperatura.
- REQ5** - Deverá usar o protocolo MQTT e o protocolo WebSocket para a comunicação entre o equipamento e o site.
- REQ6** - Deverá usar um container no Docker e uma instância EC2 no AWS para o suporte aos protocolos e ao broker utilizados no projeto.
- REQ7** - Deverá ter uma case impressa em 3D para comportar as peças e montar o equipamento proposto pelo projeto.

## 2.3 Processo de Prototipagem

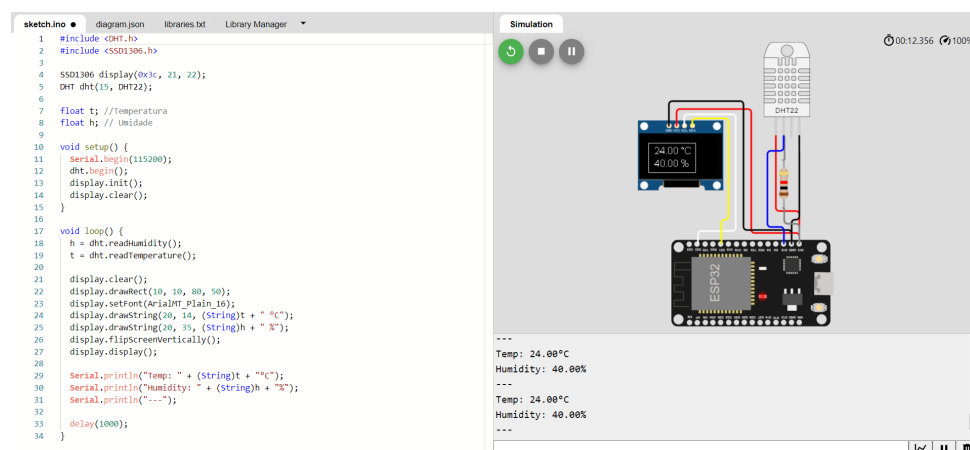
O processo de prototipagem do hardware está dividido em algumas partes como: Primeiros protótipos do projeto que foi feito no Wokwi um software de simulação de projetos eletrônicos online com suporte a ESP32, funções IoT do mesmo e alguns componentes necessários do projeto. Primeiros Protótipos Físicos, onde foi usado o software Arduino IDE para software programado em C/C++ e utilizando algumas bibliotecas que foi carregado no Heltec Lora 32, além das práticas de solda e montagem dos componentes físicos e também da criatividade dos desenvolvedores do projeto. No circuito eletrônico foi usado o software Fritzing para o desenho e montagem do circuito eletrônico do projeto. No desenho 3D foi utilizado o TinkerCad para a modelagem 3D da case do equipamento e o UltiMaker Cura para o fatiamento para impressão 3D da case. Por fim foi utilizado o EasyEDA para o desenho da placa de circuito impresso do projeto.

### 2.3.1 Primeiros Protótipos no Simulador

**Primeiro Protótipo:** 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 de temperatura e umidade DHT22 mostrando a temperatura coletada do “ambiente”. Porém isso o que é demonstrado abaixo foi um primeiro “esboço” do projeto.

- Interface do Wokwi com o protótipo:

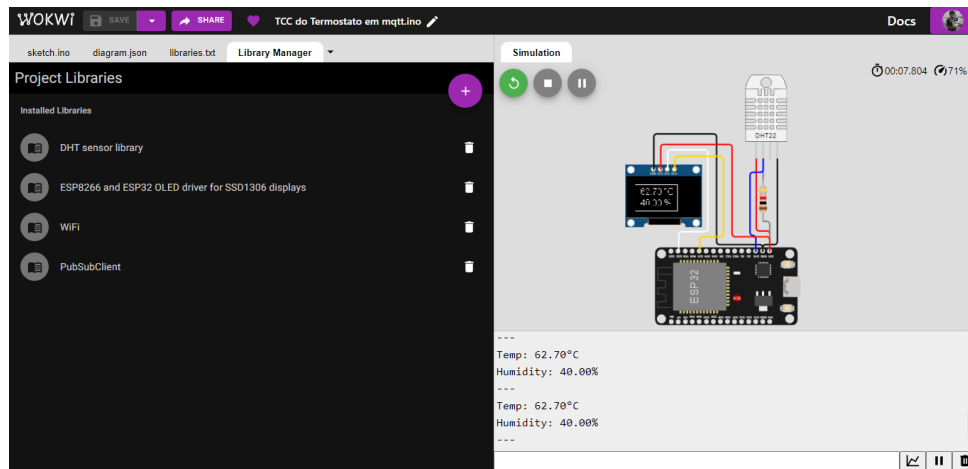
Figura 1 – Interface do Wokwi





- Bibliotecas utilizadas:

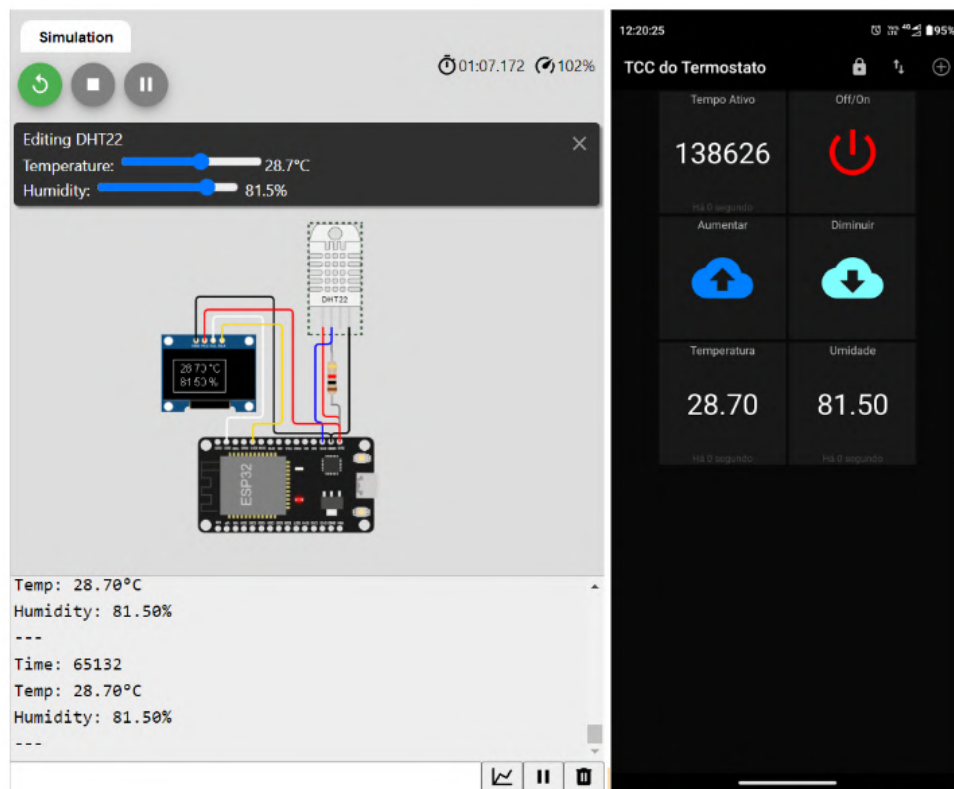
Figura 2 – Bibliotecas Wokwi



**Segundo Protótipo:** Nesse protótipo, usando como base o protótipo anterior, foi desenvolvido o código que se conecta a uma rede Wi-Fi com internet e que faz o envio das informações captadas pelo sensor de temperatura e umidade DHT22 para um broker MQTT. No código é mostrado que o envio das informações é feito através de um tópico MQTT e usando o aplicativo MQTT dashboard, criamos uma interface onde podemos colocar o tópico em uma caixa de exibição do aplicativo para justamente exibir as informações enviadas. Na prática o “hardware” não mudou nada, apenas o código que estará disponível dos criadores.

- Interface do Wokwi com o segundo protótipo e a interface de controle e monitoramento no MQTT dashboard:

Figura 3 – Interface do Wokwi e do MQTT Dashboard



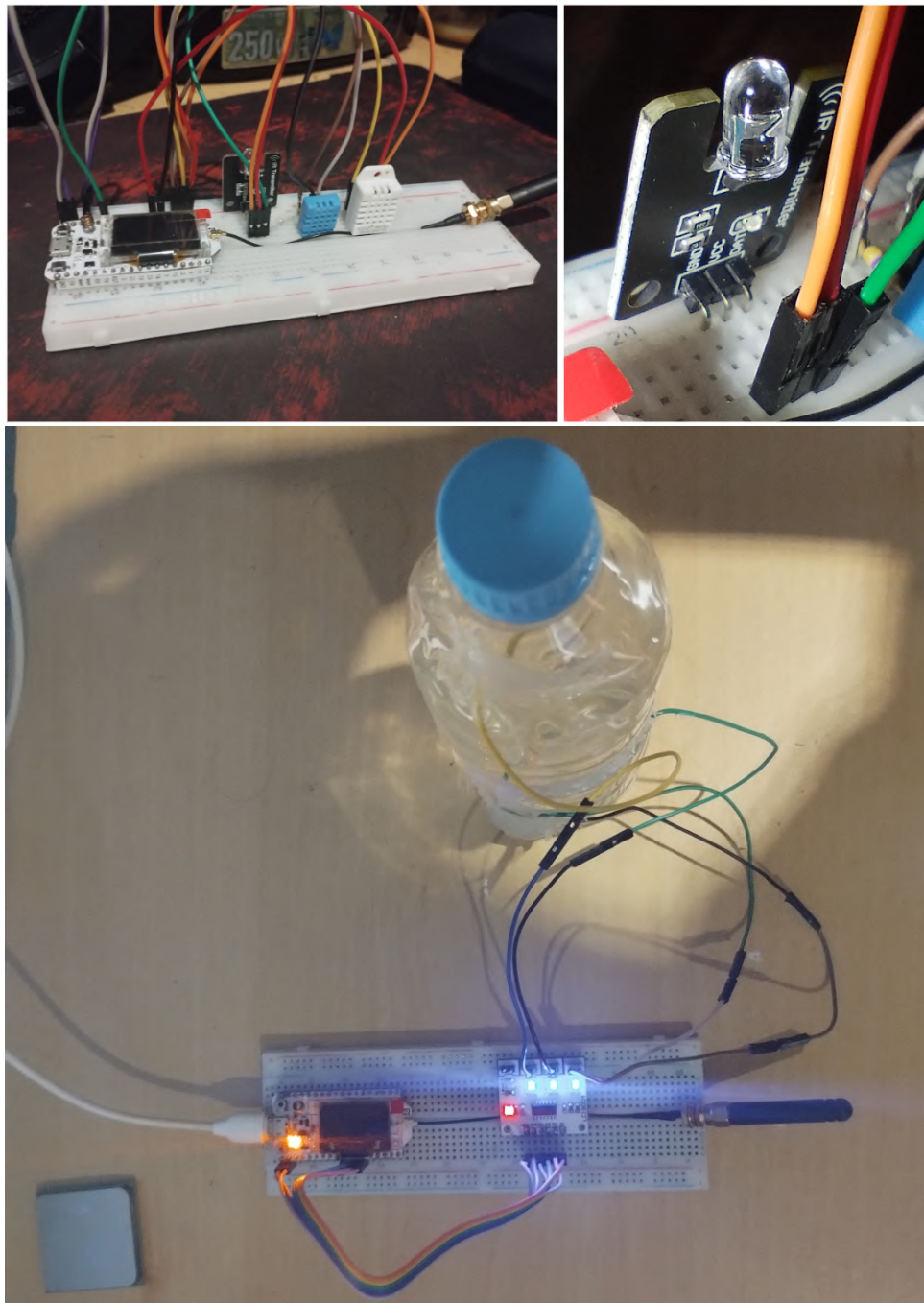
A configuração do MQTT Dashboard consiste em criar uma aba para onde ficará a sua interface e durante a criação dessa aba você deve colocar o endereço e a porta do broker MQTT que está sendo utilizado. No caso deste protótipo o endereço “broker.hivemq.com” com a porta “1883”. Depois disso deve-se criar 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. Esses botões de ligar e desligar, aumentar a temperatura e diminuir a temperatura são funcionais para o protótipo físico.

### 2.3.2 Primeiros Protótipos Físicos

**Primeiro Protótipo:** O primeiro protótipo físico foi montado inteiramente em uma protoboard de 830 pontos com jumps e não tinha uma case e tela ainda não exibia as informações de temperatura. Ao mesmo tempo, de forma separada tinha um protótipo específico para o nível d'água, ou seja, não era tudo junto no início do projeto e de novo, a tela ainda não exibia nada.

- Primeiro protótipo físico ainda usando apenas a protoboard

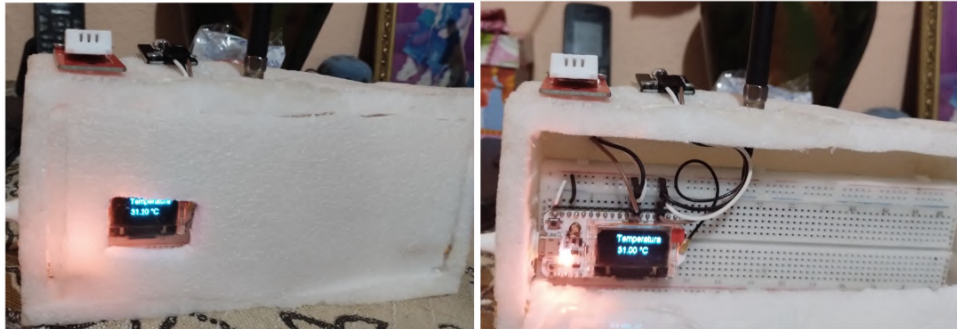
Figura 4 – Fotos do Primeiro Protótipo Físico



**Segundo Protótipo:** Para o segundo protótipo foi feito uma case feita de isopor com uma tampa que tem um buraco para a tela OLED com uma plástico transparente para proteger a tela, nesse protótipo ainda era usado a protoboard de 830 pontos. Como o nível d'água ainda estava separado, ainda não havia uma case para ele.

- Segundo protótipo físico com a case de isopor

Figura 5 – Fotos do Segundo Prototipo Físico



**Terceiro Protótipo:** Para o terceiro protótipo foram feitas as seguintes adaptações: primeiro que a case de isopor foi cortada ao meio já que houve a troca da protoboard de 830 pontos por uma protoboard de 400 pontos, assim diminuindo o tamanho do protótipo.

- Terceiro protótipo físico com a nova protoboard e a case de isopor reduzida

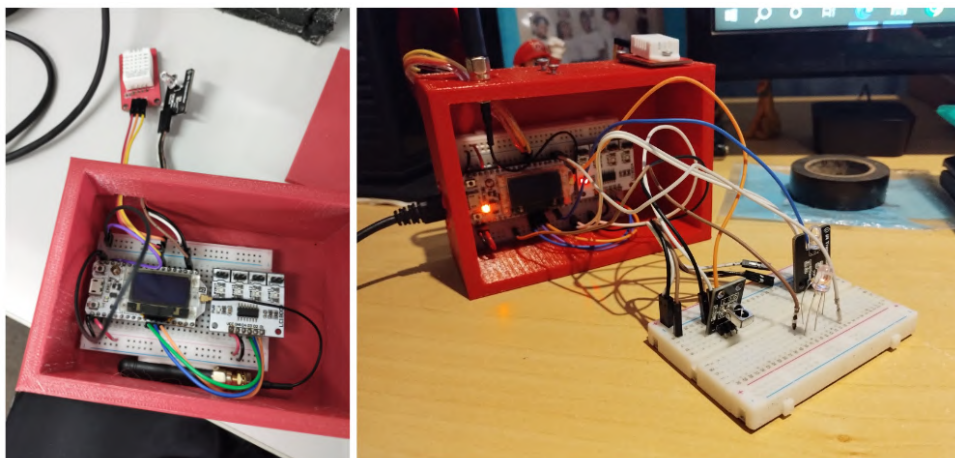
Figura 6 – Fotos do Terceiro Prototipo Físico



**Quarto Protótipo:** Para o quarto protótipo foi impresso em 3D uma case nova utilizando as medidas dos componentes e principalmente da protoboard como base para modelar a case. O equipamento foi impresso em material ABS, porém não ficou com todas as medidas corretas e também faltou a cavidade para a tela do Lora 32 na tampa e faltou a cavidade para a conexão USB do Lora 32. Basicamente era uma case fechada.

- Quarto protótipo físico com a nova case impressa em 3D

Figura 7 – Fotos do Quarto Prototipo Físico





### 2.3.3 Circuito Eletrônico

O circuito eletrônico foi desenhado no software Fritzing. Nele há o desenho do circuito, o diagrama e o esquemático. Explicando as conexões, temos um microcontrolador Lora 32 conectado à uma protoboard, através de jumps, o Sensor de Temperatura e Umidade DHT22 (Que está ligado na porta 25 do Lora 32), o Sensor Eletrônico de nível d'água diy 4 (Que está ligado nas portas 36, 37, 38 e 39 do Lora 32) e o Transmissor infravermelho HX-53 (Que está ligado na porta 13 do Lora 32) também estão conectados na protoboard nas respectivas trilhas que estão conectados os pinos que estão definidos na programação do Lora 32 para seu respectivo funcionamento. Infelizmente nesse circuito não terá a representação do Sensor Eletrônico de nível d'água diy 4 porque o mesmo não se encontra presente no software Fritzing.

- Imagens referentes ao circuito eletrônico

Figura 8 – Desenho do Circuito

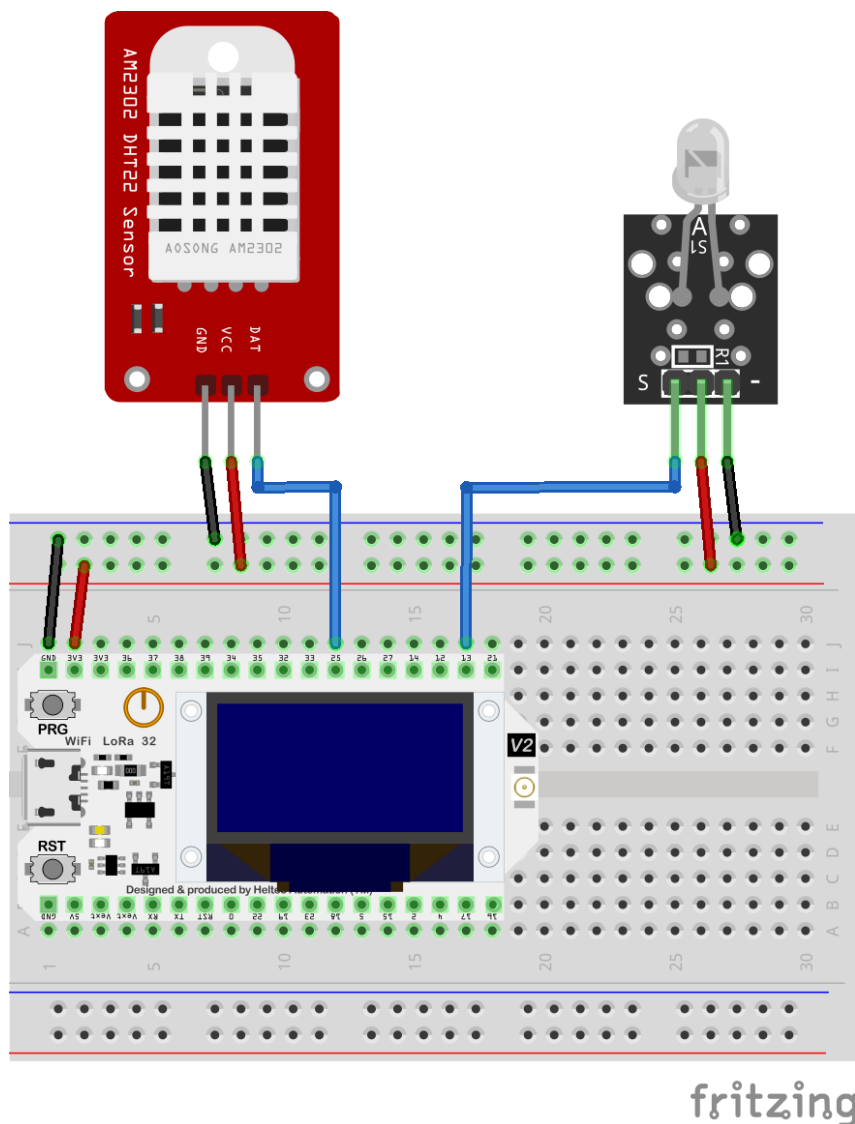


Figura 9 – Diagrama do Circuito

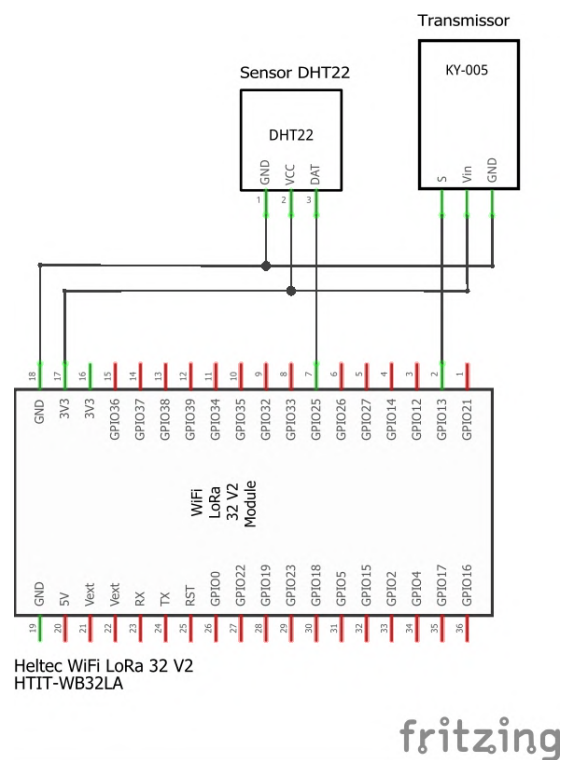
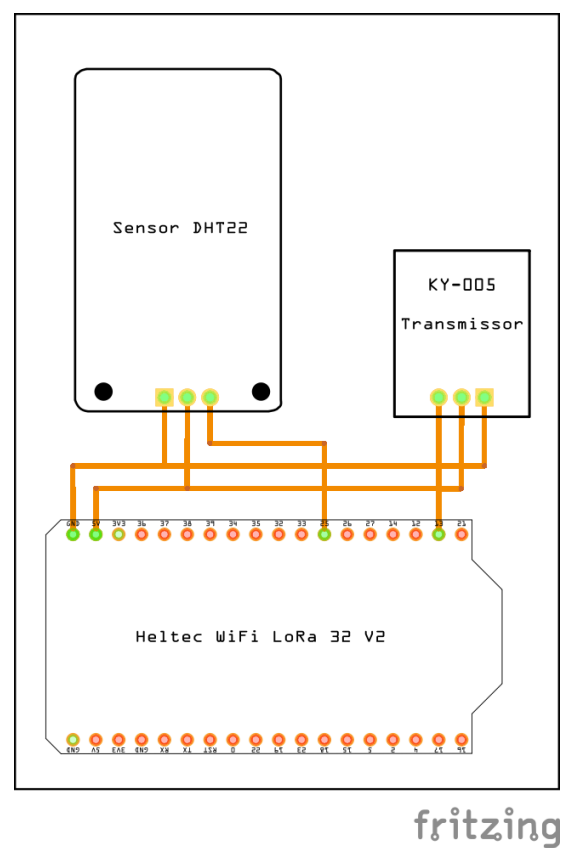


Figura 10 – Esquemático do Circuito

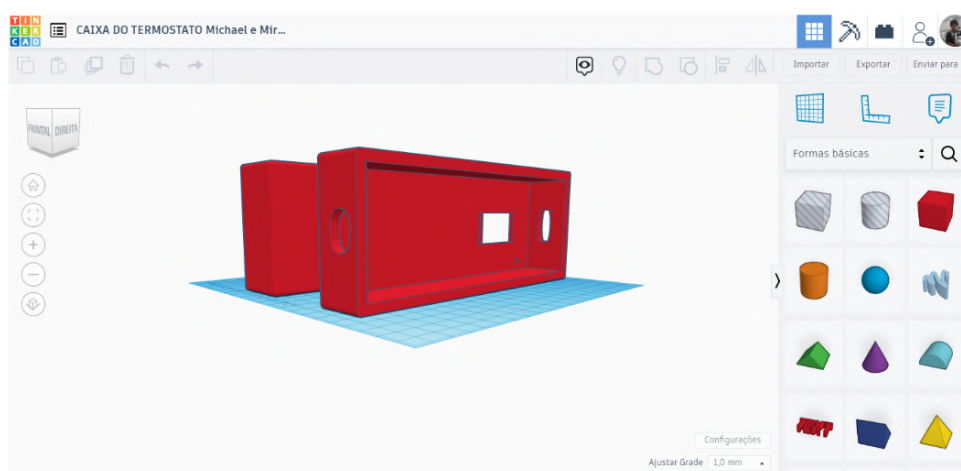


### 2.3.4 Desenho 3D e Desenho da PCB

A case foi desenvolvida por Gabriel Sales e Michael Lima para comportar todos os componentes necessários. Que são a protoboard de 400 pontos, o Lora 32, o Sensor de Temperatura e umidade DHT22, e o Transmissor Infravermelho HX-53. Ele foi desenvolvido em um software chamado Tinkercad da Autodesk em uma parte específica do software que é voltada para modelagem 3D.

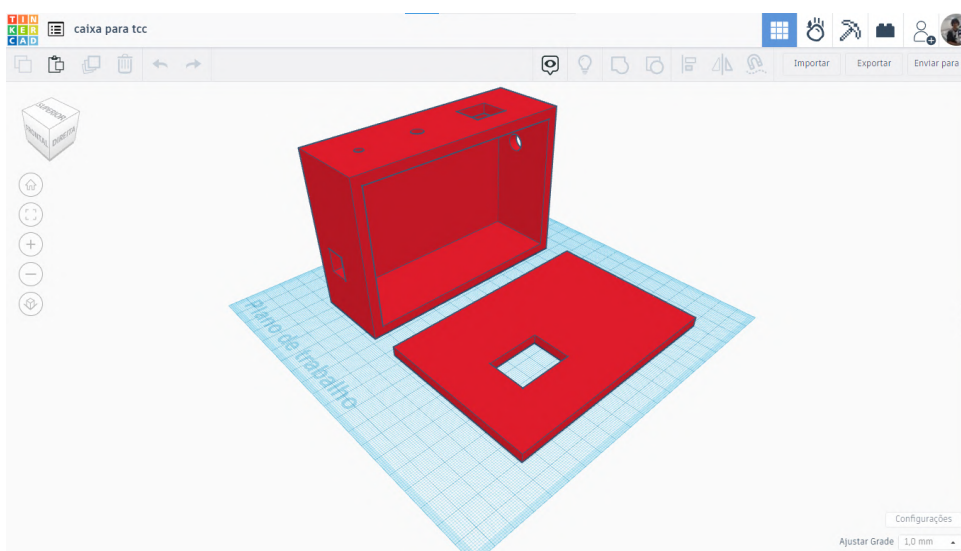
- Primeiro Modelo 3D da case feito anteriormente para comportar a protoboard de 830 pontos

Figura 11 – Imagem da primeira case modelada em 3D do projeto



- Último Modelo 3D feito para comportar a protoboard de 400 pontos com o seus componentes, assim sendo a versão final e definitiva da case para o equipamento

Figura 12 – Imagem da case atual do projeto modelada em 3D

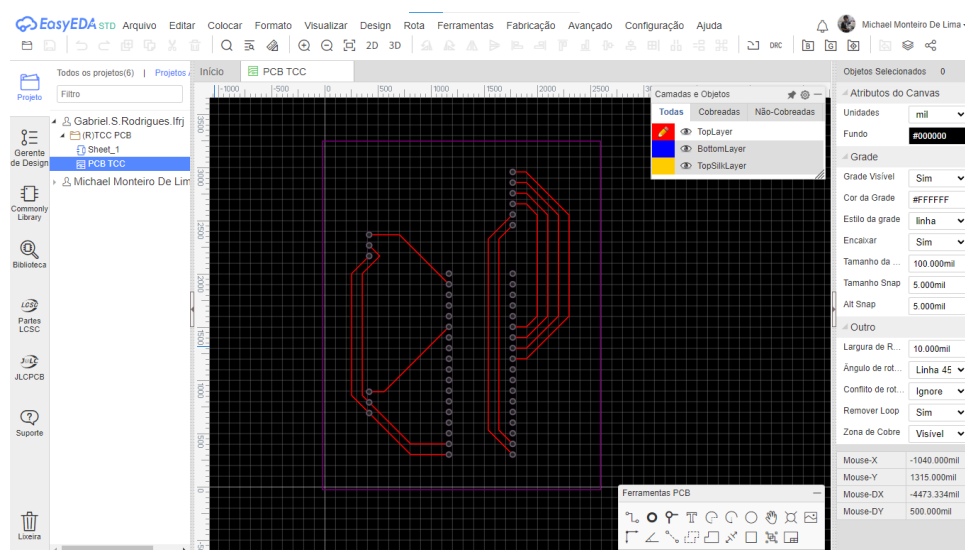




A PCB foi desenvolvida por Gabriel Sales e Michael Lima para comportar todos os componentes necessários em uma placa de circuito impresso sem a necessidade de usar protoboard, ou seja, substituindo a protoboard. A ideia é imprimir essa peça do projeto na prototipadora PCB Voltera V-One que está presente no campus, mas por motivos de tempo do co-orientador, a PCB não pode ser impressa a tempo da data de entrega do TCC e sua defesa. Porém o desenho do circuito impresso foi feito no software chamado EasyEDA.

- Nesse modelo temos o socket para o Lora 32 e as portas para os sensores: DHT22 e DIY 4; e para o transmissor infravermelho HX-53

Figura 13 – Imagem da PCB que substituiria a Protoboard



## 3 IMPLEMENTAÇÃO FINAL

A implementação final é no que diz respeito a parte final do projeto, a conclusão, o que foi feito e como os desenvolvedores do projeto efetuou na prática o projeto.

### 3.1 Passo a Passo

Nessa sessão mostraremos passo-a-passo como foi realizado o projeto.

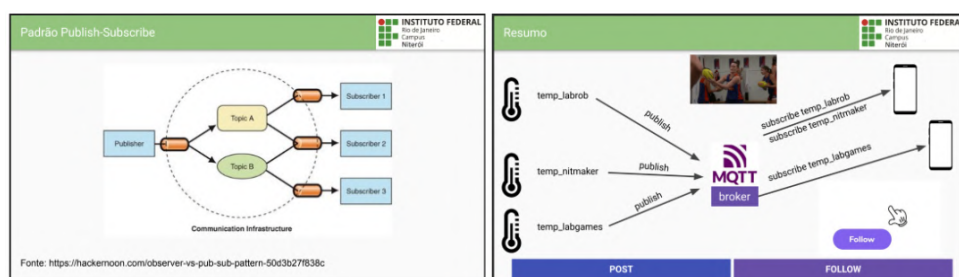
#### 3.1.1 Passo 1 - Programação do hardware

O Desenvolvimento do software integrado no hardware, que é o código em C/C++ desenvolvido que está carregado no Lora 32. A linguagem de programação a ser utilizada é o C/C++ usando uma IDE (Integrated Development Environment) chamada Arduino IDE como ambiente de desenvolvimento, que dá suporte além dessa linguagem e dá suporte a bibliotecas que estamos utilizando como DHT sensor library, Heltec ESP32 Dev-Boards, 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 utilizamos como o sensor de temperatura e umidade DHT22 e o transmissor infravermelho HX-53, que transmite os comandos do ar-condicionado.

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 emulando comandos de algum controle e atribuindo a botões no aplicativo.

- As imagens a seguir tirada de uma aula de hardwares programáveis II exemplifica isso:

Figura 14 – Imagem sobre o protocolo MQTT e seu padrão Publish-Subscribe



Com isso em mente, no código foi incluído um arquivo correspondente a configuração da porta do Transmissor Infravermelho e foi incluído as bibliotecas citadas no início do texto. E após isso foram definidos os parâmetros da rede e os parâmetros do protocolo MQTT como ID, Broker, Porta e os Tópicos:

Figura 15 – Trecho de Código onde está definido parâmetros do protocolo MQTT

```
16 //Definindo os parâmetros do protocolo MQTT como ID, Broker, Porta e os Tópicos
17 #define MQTT_ID "ifresources-tcc-lora32"
18 #define MQTT_BROKER "broker.emqx.io"
19 #define MQTT_PORT 1883
20 #define MQTT_MILLIS_TOPIC "tcc_ifresources_millis"
21 #define MQTT_SECONDS_TOPIC "tcc_ifresources_seconds"
22 #define MQTT_DHT_TEMP_TOPIC "tcc_ifresources_temp"
23 #define MQTT_DHT_HUMD_TOPIC "tcc_ifresources_humd"
24 #define MQTT_NIVEL_TOPIC "tcc_ifresources_nivel"
25 #define MQTT_SENDRW_TOPIC "tcc_ifresources_sendraw_manage"
```

Após isso foi definido o objeto para a conexão Wi-Fi e criação do cliente MQTT, definido as Strings usadas para o envio das informações, definido a banda de frequência de comunicação sem fio e definido o objeto para o DHT22 e definido variáveis e as funções.

Figura 16 – Trecho de Código onde está definido objetos, strings e variáveis

```
27 //Definindo objeto para a conexão Wi-Fi e criação do cliente MQTT
28 WiFiClient espClient;
29 PubSubClient MQTT(espClient);
30
31 //Definindo as Strings usadas para o envio das informações
32 char millis_str[10] = "";
33 char seconds_str[10] = "";
34 char temp_str[10] = "";
35 char humd_str[10] = "";
36 char nivel_str[10] = "";
37
38 //Definindo a banda de frequência de comunicação sem fio e definindo objeto para o DHT22
39 #define BAND 915E6
40 DHTesp dht;
41
42 //Variáveis do Nivel d'água
43 using namespace std;
44 bool isPressed = "";
45 int nivel = 0;
46 int buttonState = 0;
47
48 //Variáveis do Sensor DHT22
49 float currentTemp;
50 float currentHumidity;
51
52 // Variável global para armazenar o tempo inicial
53 unsigned long tempoInicial;
```

Depois disso, definidos 8 funções, sendo elas: “void setupWifi()”, “void setupMQTT”, “void getTemp()”, “void getHumidity()”, “void getNivel()”, “void setup(void)”, “void loop(void)” e “void mqtt\_ifresources\_callback(char\* topic, byte\* payload, unsigned int length)”.

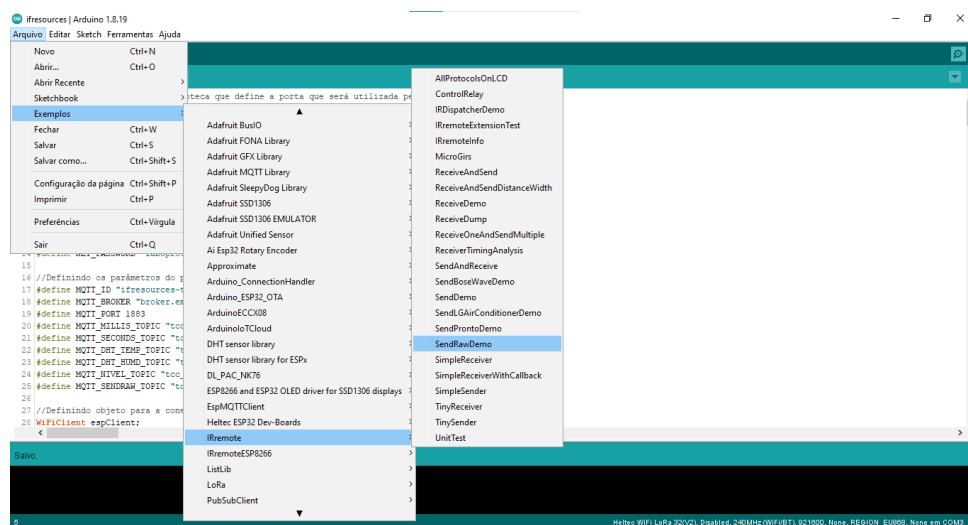
Explicando cada uma: a “void setupWifi()” é a função que conecta o Lora 32 em algum Wi-Fi, a “void setupMQTT” é a função que conecta o Lora 32 no Broker MQTT,

“void getTemp()” é função da Temperatura, “void getHumidity()” é função da Umidade, “void getNivel()” é função do Nível d’água, “void setup(void)” é a função onde é definido a configuração da primeira tela e a pinagem de alguns componentes, “void loop(void)” é a função onde é definido a configuração da segunda tela que mostra a temperatura e aonde é enviado as informações para o broker e “void mqtt\_ifresources\_callback(char\* topic, byte\* payload, unsigned int length)” é a função Callback onde é configurado os comandos do ar-condicionado de ligar, desligar, aumentar e diminuir a temperatura. Na função “void mqtt\_ifresources\_callback(char\* topic, byte\* payload, unsigned int length)” usamos como base dois códigos. Um código referência é o que se encontra no github do usuário “luizsoliveira” no repositório “arduinoExamples” lá tem um exemplo de código com comunicação bidirecional MQTT que tem uma função callback muito parecida com a utilizada neste código.

Dentro da função callback falada anteriormente, há uma sequência de condições “if e else if” dentro de cada uma delas, há um bloco condicional com um comando referentes ao sendRaw que é um array dos comandos infravermelhos captados pelos controles do ar-condicionado e isso foi retirado de um exemplo da biblioteca IRremote que o exemplo “SendRawDemo”, então por isso que há um arquivo específico chamado "PinDefinitionsAndMore.h" para definir o pino do transmissor infravermelho HX-53, que no caso é o pino 13, esse arquivo é derivado desse exemplo da biblioteca.

Para a visualização do código completo e compreensão melhor do mesmo, acesse o github dos desenvolvedores do projeto.

Figura 17 – Imagem onde mostra a biblioteca e o exemplo que foi retirado o funcionamento do SendRaw



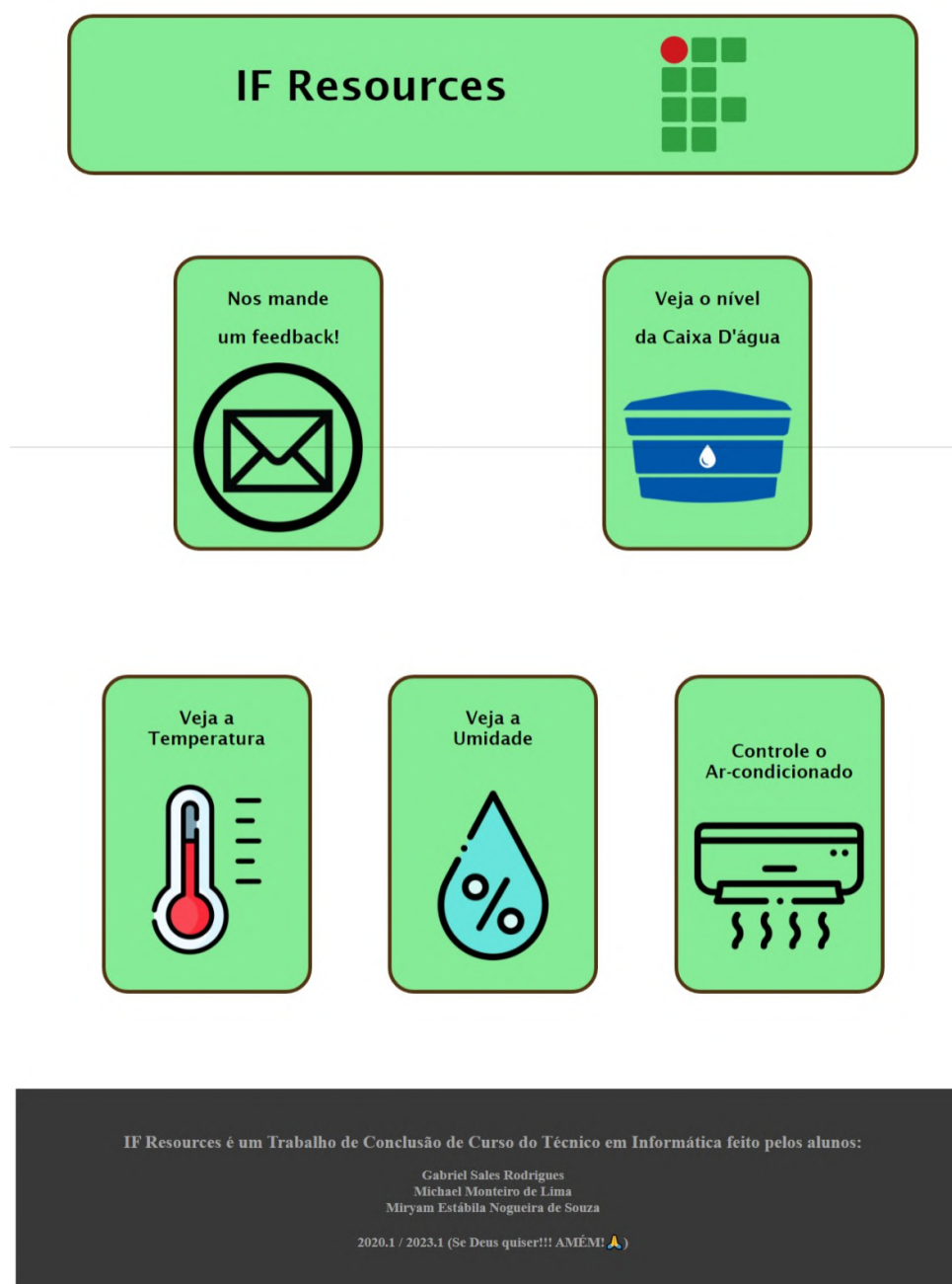
### 3.1.2 Passo 2 - Desenvolvimento do Site

O desenvolvimento do site, onde usamos linguagens de marcação e linguagens de programação como HTML, CSS e JavaScript, para isso usamos o Visual Studio Code como

ambiente de desenvolvimento.

Começando pelo Index.html, temos cinco botões levando para diferentes páginas, sendo elas:

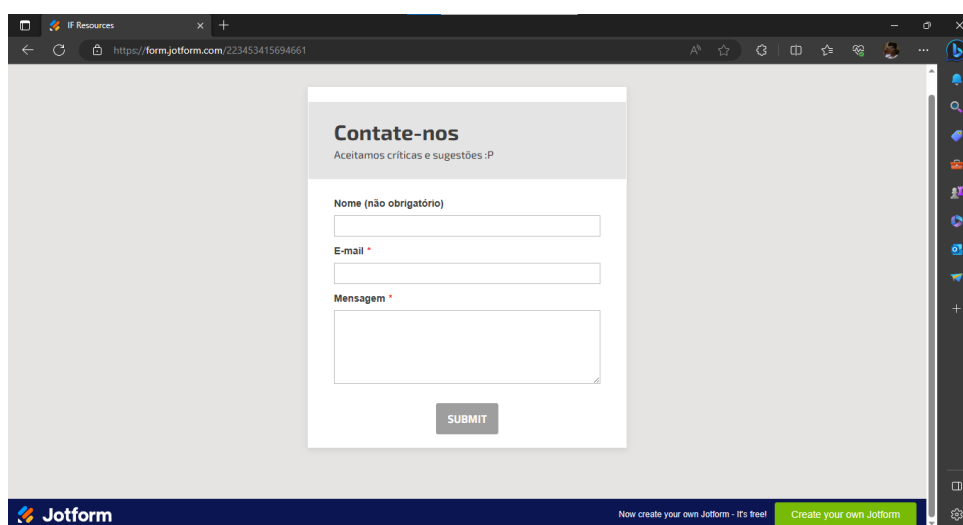
Figura 18 – Imagem da tela inicial do site



- **Nos mande um feedback!** - Envia o usuário para um site externo de formulários, que envia o feedback do mesmo para o e-mail dos desenvolvedores.
- **Veja o nível da Caixa D'água** - Envia o usuário para uma *web-page* que é *subscriber* do tópico "tcc\_ifresources\_nivel".

- **Veja a Temperatura** - Envia o usuário para uma *web-page* que é *subscriber* do tópico "tcc\_ifresources\_temp".
- **Veja a Umidade** - Envia o usuário para uma *web-page* que é *subscriber* do tópico "tcc\_ifresources\_humd".
- **Controle o Ar-condicionado** - Envia o usuário para uma *web-page* que é *publisher* do tópico "tcc\_ifresources\_sendraw\_manage".

Figura 19 – Imagem da pagina do formulário de feedback



The image shows a web browser window displaying a Jotform feedback form. The form is titled "Contate-nos" (Contact us) and includes the text "Aceitamos críticas e sugestões :P" (We accept criticism and suggestions :P). The form fields are: "Nome (não obrigatório)" (Name, not required), "E-mail" (with a red asterisk indicating it is required), and "Mensagem" (with a red asterisk indicating it is required). A "SUBMIT" button is located at the bottom of the form. The Jotform logo and a banner at the bottom of the form state "Now create your own Jotform - It's free! Create your own Jotform".

Figura 20 – Imagem da pagina que informa o nível da caixa d'água



Figura 21 – Imagem da pagina que informa a temperatura do ambiente



Figura 22 – Imagem da pagina que informa a umidade do ambiente



Figura 23 – Imagem da pagina de controle do ar-condicionado



Cada página faz seu respectivo papel por meio de um código em JavaScript que o conecta a *web-page* ao MQTT, atualizando assim a mensagem na tela (podendo ela ser de temperatura, umidade ou nível de água) ou enviando uma mensagem para o tópico. Essa mensagem enviada para o tópico `tcc_ifresources_senddraw_manage` foi configurada para enviar um comando por meio de infra-vermelho para o ar-condicionado fazer a ação desejada pelo usuário. Essa conexão entre a *web-page* e o MQTT é feito por meio de um protocolo *Web-socket*, protocolo esse que necessita de um container no *Docker* configurado com uma imagem EMQX que faz a comunicação do protocolo WebSocket e do protocolo MQTT. Esse container no *Docker* está instalado em uma instância EC2 que opera com o sistema operacional ubuntu na AWS. O site está hospedado na hostinger e tem o domínio

Figura 24 – Imagem do painel da instância EC2 que está rodando na AWS

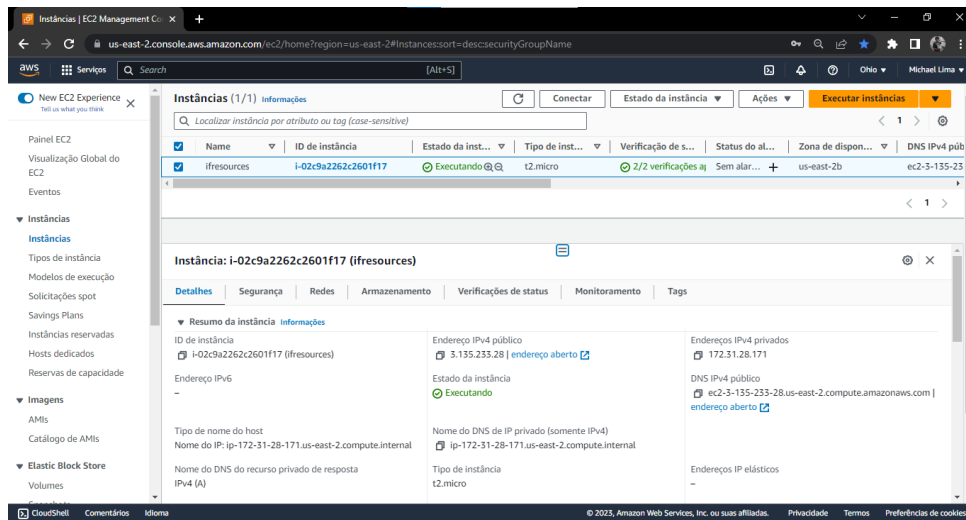


Figura 25 – Imagem do painel de configuração da imagem EMQX que está rodando na AWS

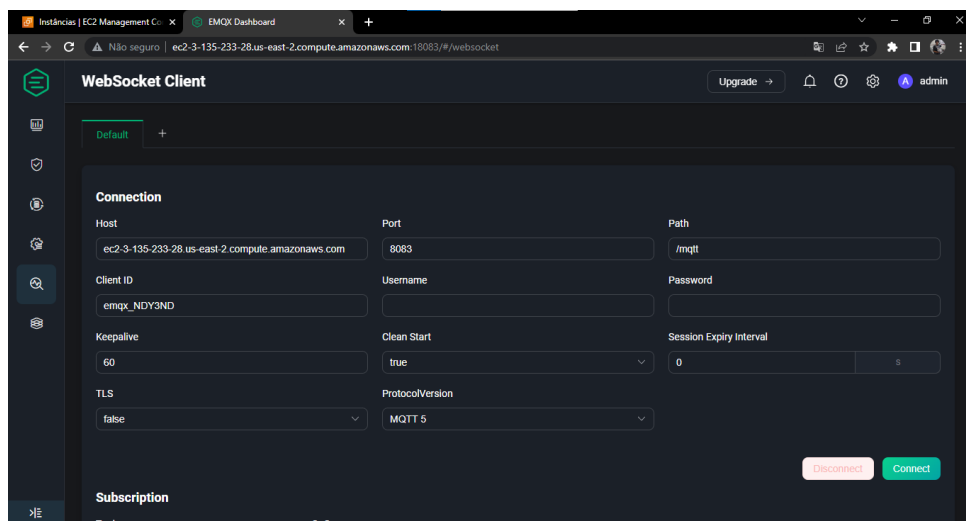
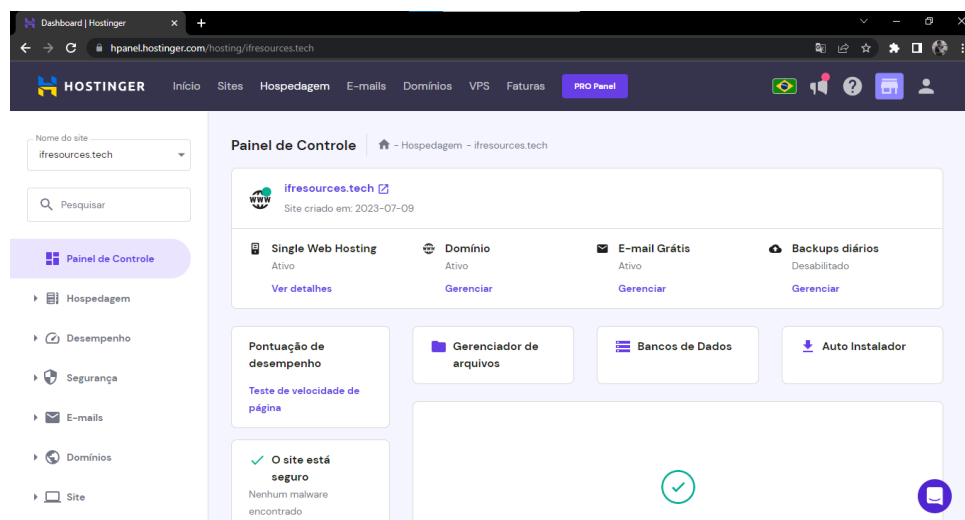




Figura 26 – Imagem do painel de configuração do site que está hospedado na hostinger

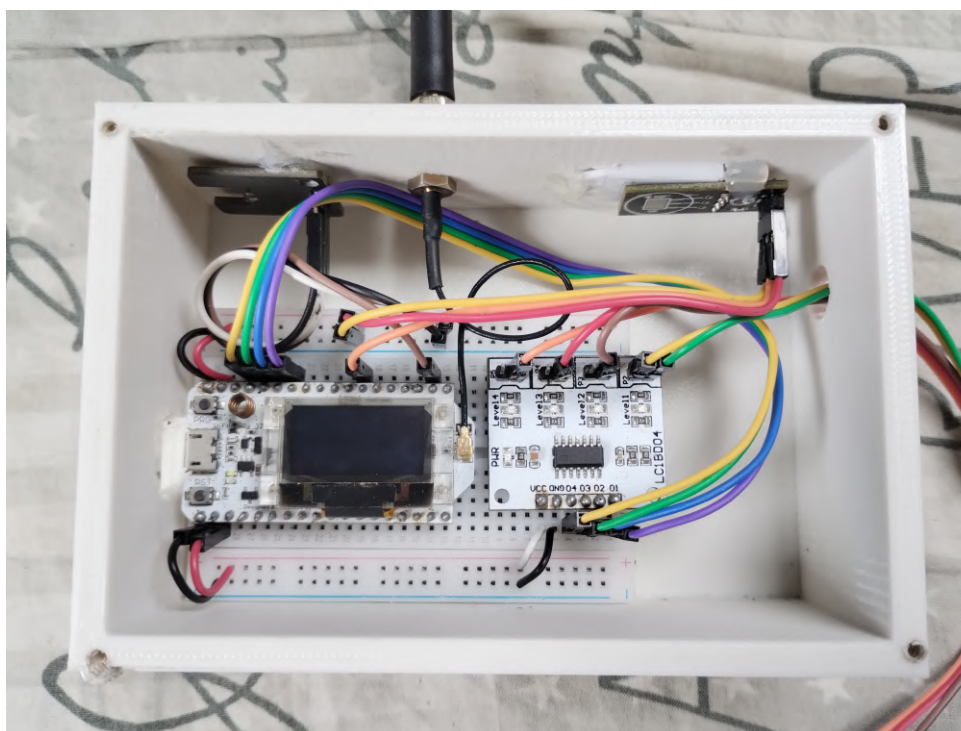


### 3.1.3 Passo 3 - Construção do hardware

Conectaremos ao Lora 32 os sensores de temperatura e umidade DHT22, sensor de nível de água Diy 4 e um transmissor infra-vermelho HX-53.

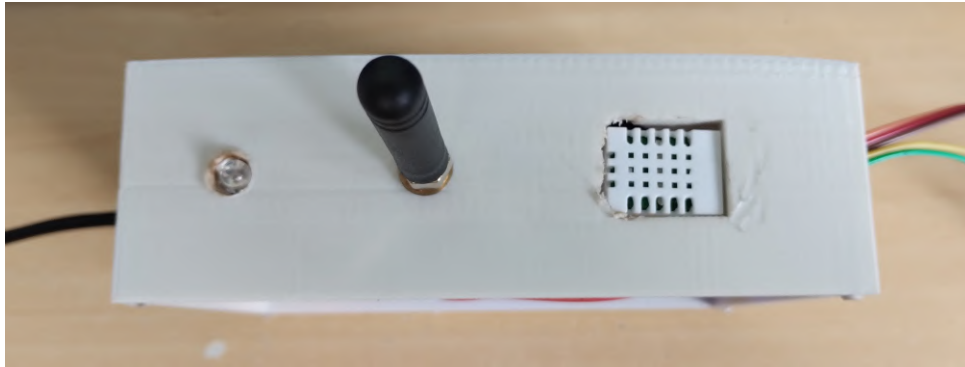
- Nessa imagem podemos ver a parte interna do equipamento que tem na parte superior: a antena do Lora 32, o sensor de temperatura e umidade DHT22 e o transmissor infravermelho HX-53. No meio temos conectado na protoboard de 400 pontos o Heltec Lora 32 e o sensor de nível de água Diy 4 e tudo conectado através de jumpers.

Figura 27 – Imagem da parte interna do equipamento



- Nessa imagem podemos ver na parte superior do equipamento a antena de comunicação do Lora 32 no centro dela, na direita o sensor de temperatura e umidade DHT22 e na esquerda o led do transmissor infravermelho HX-53.

Figura 28 – Imagem da parte superior do equipamento

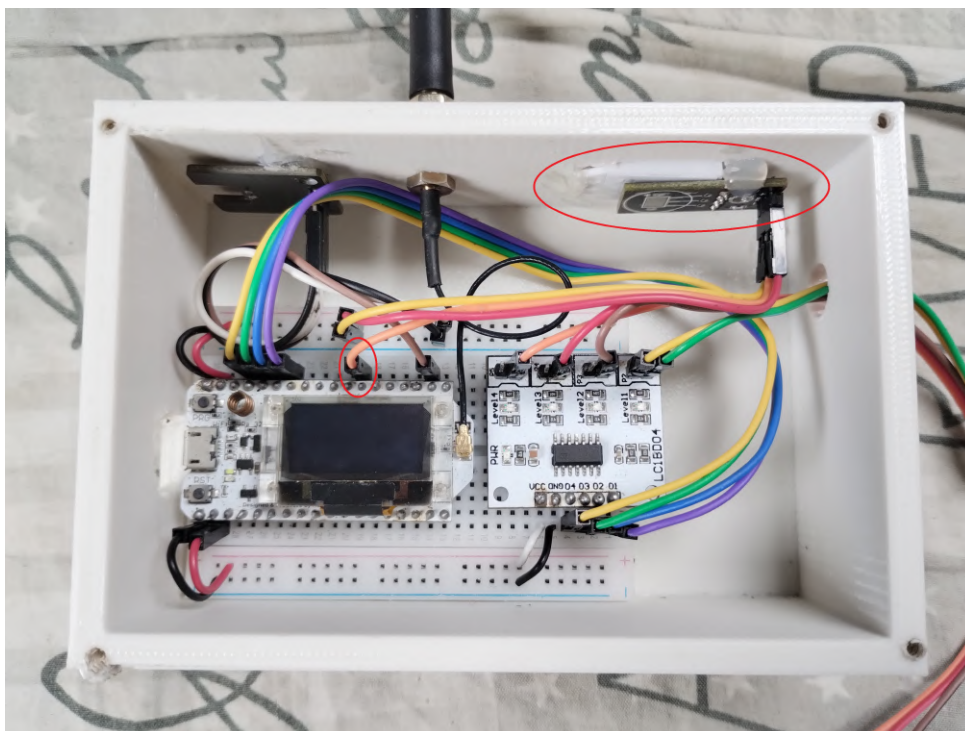


#### 3.1.3.1 Sensor de temperatura e umidade DHT22

Utilizaremos a porta 25 do LORA-32, definido na função "void setup(void)". O valor será lido como char (similar a string). As informações serão recebidas pelo micro-controlador e enviadas na tela e atribuídas no MQTT na função "void loop(void)".

- Nessa imagem podemos ver circulado em vermelho o sensor de temperatura e umidade DHT22 na parte superior direita da case e também circulado em vermelho um fio laranja conectado na porta 25 do Lora 32.

Figura 29 – Imagem da conexão do sensor de temperatura e umidade DHT22 no Lora 32

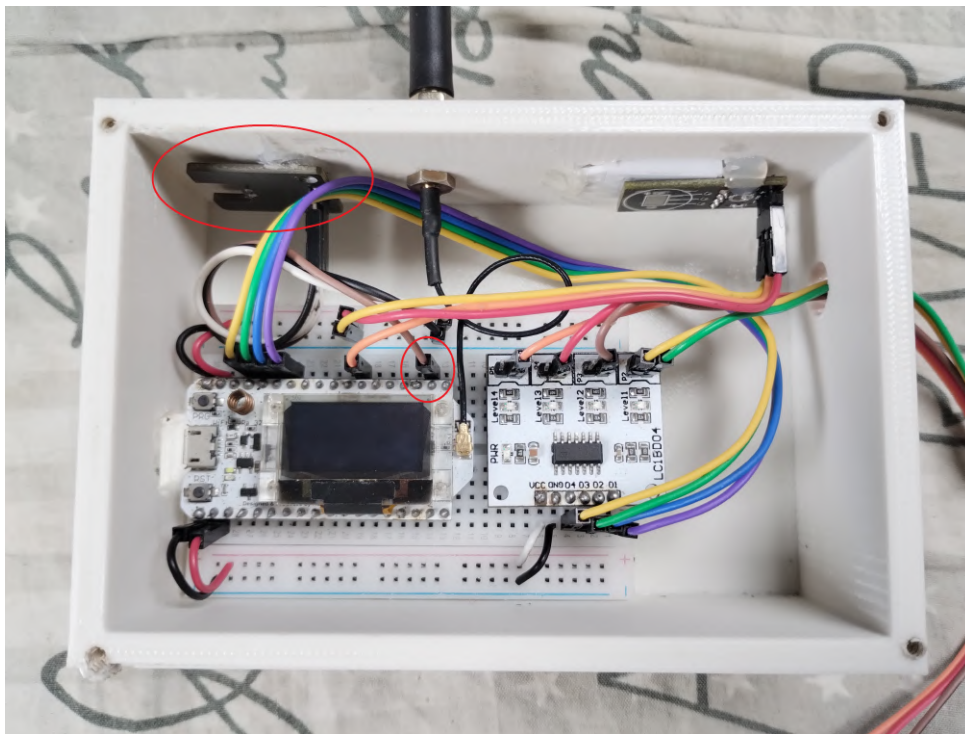


### 3.1.3.2 Transmissor Infra-vermelho HX-53

Utilizaremos a porta 13 do LORA-32, definido no arquivo "PinDefinitionsAndMore.h". definindo a função callback, há uma sequência de condições "if e else if" dentro de cada uma delas, há um bloco condicional com um comando referentes ao sendRaw que é um array dos comandos infravermelhos captados pelos controles do ar-condicionado, ou seja, cada array é um comando do controle do ar-condicionado.

- Nessa imagem podemos ver circulado em vermelho o transmissor infra-vermelho HX-53 na parte superior esquerda da case e também circulado em vermelho um fio marrom conectado na porta 13 do Lora 32.

Figura 30 – Imagem da conexão do transmissor infra-vermelho HX-53 no Lora 32



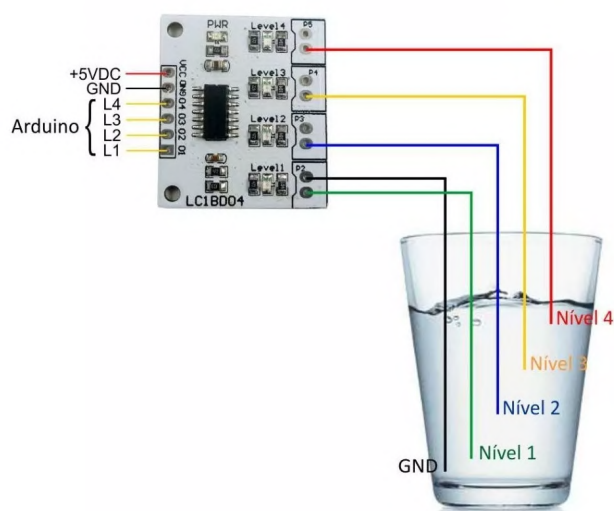
### 3.1.3.3 Sensor de Nível de Água Diy 4 (LC1BD04)

Utilizaremos as portas 36, 37, 38 e 39 do LORA-32. O sensor de nível de água Diy 4 funciona praticamente por meio de um curto-circuito entre os terminais dos níveis das placas, utilizando a condutibilidade elétrica da água. Os valores serão lidos pelo micro-controlador apenas como 1 e 0 e o papel de enviá-los para MQTT é atribuído na função "void loop(void)".



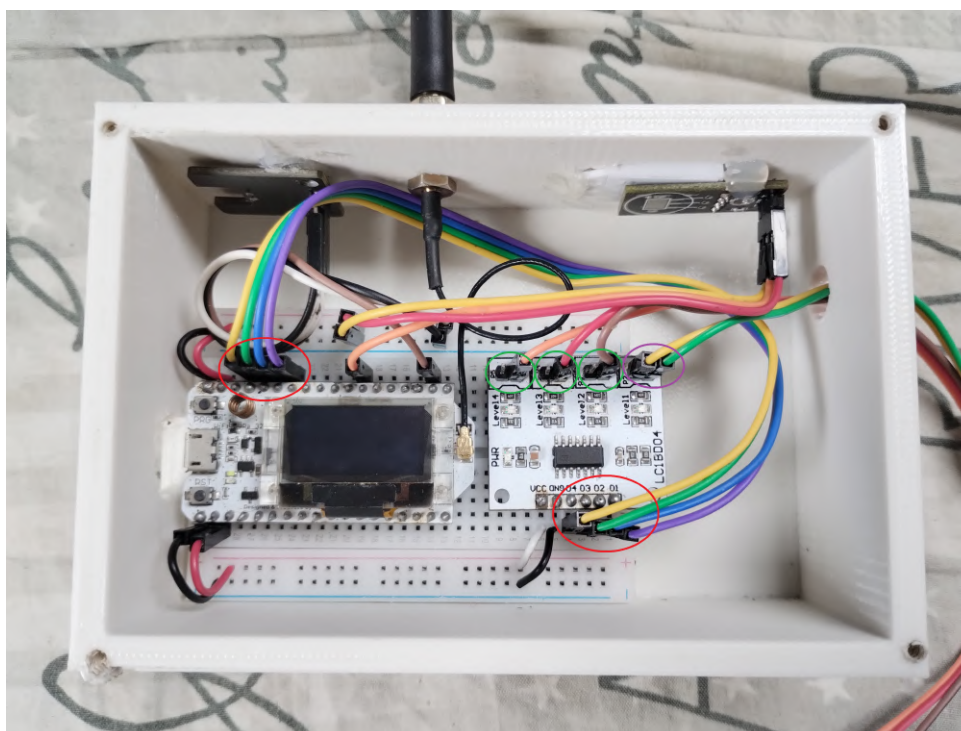
Figura 31 – Representação do funcionamento do nível de água

## Sensor de Nível de Água



- Nessa imagem podemos ver circulado em vermelho as portas D1, D2, D3 e D4 do sensor de nível de água Diy 4 e também circulado em vermelho os fios que estão conectados nas portas 36, 37, 38 e 39 do Lora 32. Circulado em verde estão as portas que quando estão em curto acendem um led e informam ao Lora 32 o respectivo nível d'água

Figura 32 – Imagem da conexão do sensor de nível de água Diy 4 no Lora 32



## 3.2 Produto Final

Aqui temos o produto final, o equipamento proposto pelo projeto em sua versão final, com a tampa parafusada com 4 parafusos de fenda e com uma abertura para a tela do Lora 32 que mostra a tela inicial quando é ligado e se conectado ao Wi-Fi e quando conectado, mostra a informação de temperatura na tela. No site mostra além da informação de temperatura, as informações de umidade e nível d'água. Com ele conectado a rede, o usuário pode controlar o ar-condicionado, deixando o equipamento apontado para o receptor infra-vermelho do aparelho e entrando no site na página de controle.

- Nessa imagem vemos o produto final, ou seja, o equipamento finalizado e pronto para uso.

Figura 33 – Imagem do Equipamento Finalizado



## 4 CONSIDERAÇÕES FINAIS

Esse projeto tinha como intuito facilitar a vida dos estudantes do Instituto Federal do Rio de Janeiro - Campus Niterói por meio de monitores de temperatura, umidade e nível de água e controles de ar-condicionado, a partir de tecnologias como C/C++, HTML, CSS, JavaScript e Docker.

O projeto também tem como objetivo incentivar os alunos e professores do IFRJ Campus Niterói o aprimoramento e inserção no mundo da eletrônica, das tecnologias IoT e a Pesquisa e Extensão no dia a dia dessa instituição de ensino. O projeto também buscar ajudar nos estudos de IoT, prototipação eletrônica, Impressão 3D e desenvolvimento de site ligados à equipamentos eletrônicos construídos pelas próprias explorando a cultura maker e filosofia do "Do It Yourself (DIY)".

Os códigos escritos e utilizados são de domínio público e estão todos disponíveis no GitHub no perfil dos criadores desse projeto. O github dos criadores e desenvolvedores do projeto. Gabriel Sales Rodrigues, Michael Monteiro de Lima e Miryam Estábila Nogueira de Souza se encontram no final das referências deste artigo.

## 5 REFERÊNCIAS

- ESP32 LORA WIFI SX1278. Gustavo Teixeira. Usinainfo.
- PROJETO ESP32 COM DISPLAY OLED E SENSOR AM2302 DHT22 MEDINDO TEMPERATURA E UMIDADE. Gustavo Teixeira. Usinainfo.
- CONHECENDO A PLACA WIFI LORA ESP32 (433MHZ / 868MHZ / 915MHZ). Euler Oliveira. Master Walker.
- ESP32 WITH DHT11/DHT22 TEMPERATURE AND HUMIDITY SENSOR USING ARDUINO IDE. Alan Miller. Random nerd tutorials.
- LUIZSTATION MQTT PUBLISHER. Luiz Felipe. GitHub.
- LUIZSTATION MQTT TEMPDHT11. Luiz Felipe. GitHub.
- AIR CONDITIONERS: RECORDING LONG INFRARED REMOTE CONTROL SIGNALS WITH ARDUINO. AnalysIR.
- CAPTURANDO E ENVIANDO CÓDIGOS INFRAVERMELHO DE AR-CONDICIONADO COM O ARDUINO - SENDRAW. Danilo Queiroz Barbosa.
- COMO FAZER CONTROLE REMOTO DE TV COM ARDUINO PRO MINI. Marlon Nardi.
- ESP32 LAB 01 - DISPLAY OLED, SENSOR DHT22 E HORA VIA NTP SERVICE. EasyTrom Labs.
- RECEPTOR INFRAVERMELHO (IR) - CURSO DE ARDUINO 41. Marlon Nardi.
- MQTT WEB DASHBOARD | MQTT WEB APP USING HTML , CSS AND JAVA SCRIPT. High Voltages.
- ACESSE DADOS VIA MQTT NO NAVEGADOR - IEC69. Internet e Coisas.
- COMO COLOCAR UM SITE NO AR (2023). Hostinger Brasil.
- HOW TO POINT DOMAIN NAME TO WEB HOSTING. Hostinger Academy.
- O QUE É DOMÍNIO E HOSPEDAGEM E COMO COLOCAR SEU SITE NO AR COM A HOSTINGER. Rafaella Ballerini.
- JAVASCRIPT MQTT CLIENT: A BEGINNER'S GUIDE TO MQTT.JS. Shifan Yu. EMQ Technologies Inc.

- 
- A QUICKSTART GUIDE TO USING MQTT OVER WEBSOCKET. Shifan Yu. EMQ Technologies Inc.
  - AWS ACADEMY CLOUD FOUNDATIONS. Amazon Web Services INC.
  - TCC. Gabriel Sales. Github.
  - IFResources. Michael Lima. Github.