

Manual de Instruções

THUNDERBOLTS
PROTOTIPAÇÃO IoT PARA LOCALIZAÇÃO

Controle do Documento

Histórico de revisões

Data	Autor	Versão	Resumo da atividade
17/11/2022	Mariana	1.0	Atualização da seção 1.1.
18/11/2022	Jean e Daniel	1.1	Atualização das seções 1.2, 2.1.
19/11/2022	Gustavo e Sofia	1.2	Atualização das seções 2.2, 2.3 e 3.
04/12/2022	Mariana, Jean, Sofia e Daniel	1.3	Atualização das seções 4 e 5.
13/12/2022	Jean, Sofia e Daniel	2	Seção 6
15/12/2022	Gustavo, Jean, Sofia, Daniel e Mariana	2.1	Revisão geral e correções

Índice

1. Introdução	4
1.1. Solução	4
1.2. Arquitetura da Solução	5
2. Componentes e Recursos	7
2.1. Componentes de hardware	7
2.2. Componentes externos	8
2.3. Requisitos de conectividade	8
3. Guia de Montagem	9
4. Guia de Instalação	11
5. Guia de Configuração	16
5.1. Configuração dos softwares	16
5.1.1. Arduino IDE	16
5.1.2. Arquivo de definição da API	25
5.1.3. Node.js	27
5.2. Configuração do código no Arduino	29

6. Guia de Operação	
6.1. Introdução à interface	31
6.2. Cadastro de novos ativos e de empréstimo	32
6.3. Relatórios	32
6.4. Equipamentos cadastrados	34
6.5. Equipamentos emprestados	35
6.6. Mapa	37
6.7. Lista	38
7. Troubleshooting	41
8. Créditos	43

1. Introdução

1.1. Solução

A escola bilíngue Beacon ofereceu uma problemática que vem sofrendo no dia a dia com relação ao empréstimo de aparelhos eletrônicos para os alunos, professores e colaboradores. Muitos deles, principalmente os alunos, recebem esses aparelhos e não lembram de devolver, o que acaba gerando muitos gastos para a escola, uma vez que os funcionários precisam procurar os aparelhos, sem saber onde eles estão.

Nesse contexto, a escola forneceu dados referentes à planta da escola e ao banco de dados com informações do inventário.

Pensando nisso, foi decidido que a melhor opção é criar um sistema híbrido (utilizando de ambos RFID e ESP32) que consiga localizar esses aparelhos, e mostrar isso em uma página web, para que os funcionários tenham facilidade de encontrá-los, e saber se os aparelhos saíram do perímetro da escola.

Para que isso seja possível, vai ser feita a prototipação de hardware com a programação de microcontroladores.

A solução deverá ser utilizada pelos funcionários que trabalham na escola, sendo que quando chegar o momento de fazer a verificação dos aparelhos eletrônicos, e dos ativos da escola, eles abram a aplicação web, que irá mostrá-los quais são os dispositivos e onde eles estão. Assim, facilitando o trabalho dos funcionários, que se necessário podem ir até o local, que o aparelho está localizado e encontrá-lo de forma mais simples e fácil.

Com a nossa solução, a Beacon deixará de ter gastos com novos aparelhos, além de poder redirecionar o tempo usado para procurar esses mesmos dispositivos.

1.2. Arquitetura da Solução

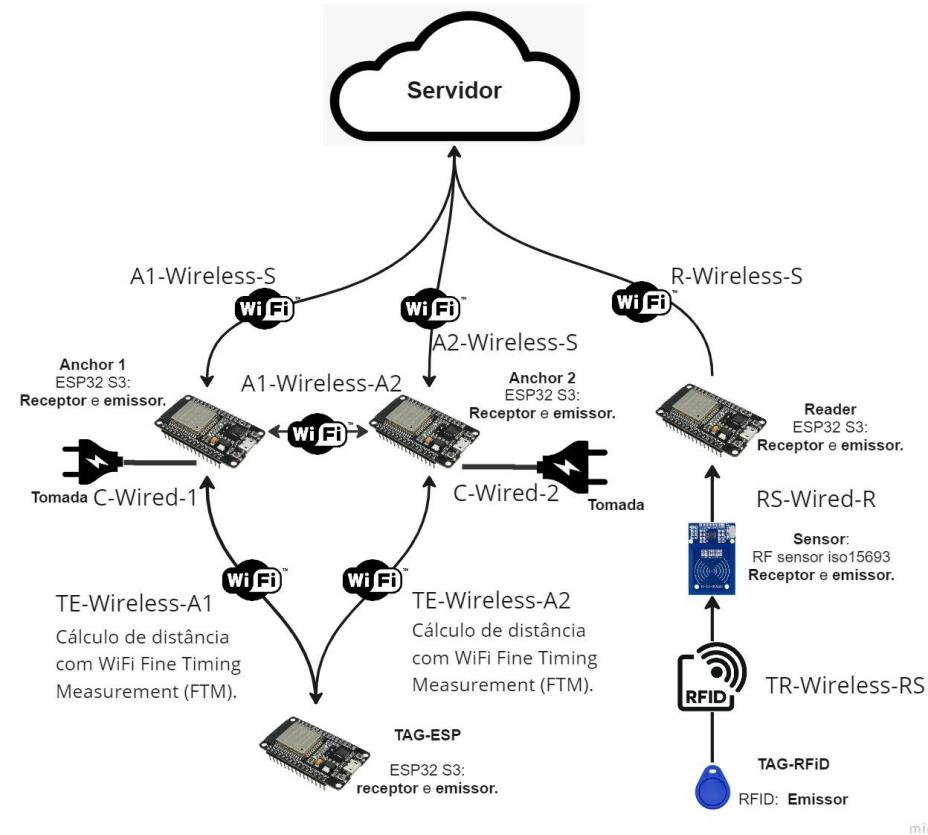


Figura 1: Diagrama representando a terceira versão da Arquitetura de Hardware do Sistema. A solução é híbrida, apresentando tanto a localização dos dispositivos tecnológicos através de WiFi, quanto a contabilização dos demais ativos da instituição através de RFID.

Fonte: Autoria Própria.

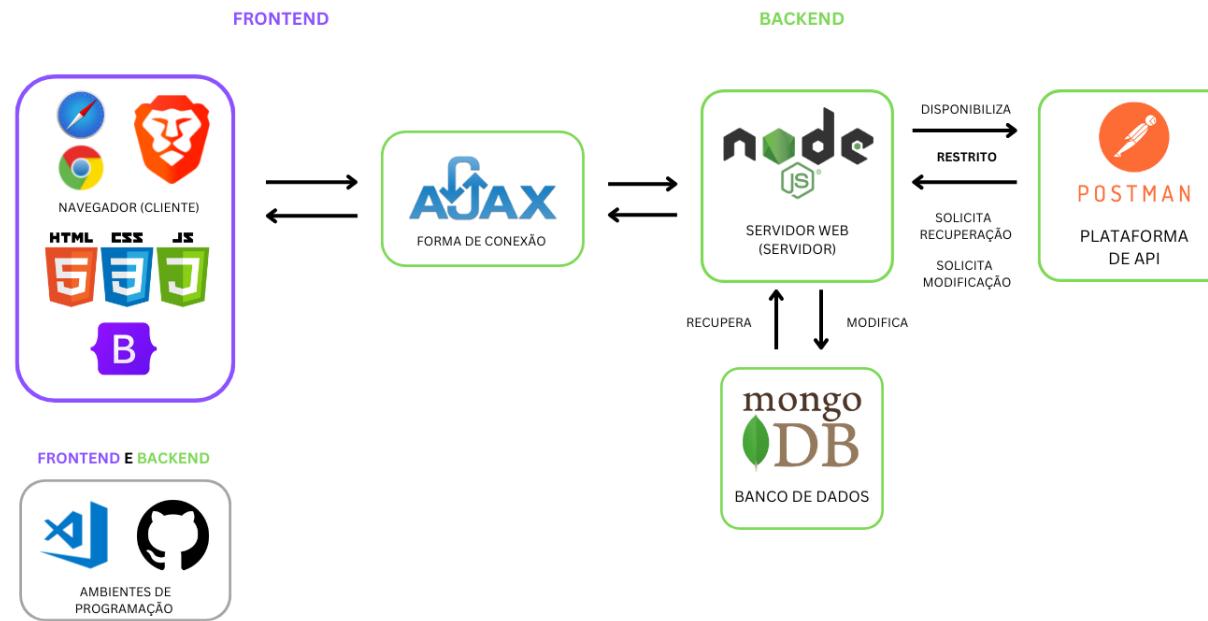


Figura 2: Diagrama de autoria própria, representando a terceira versão da Arquitetura de Software do Sistema. A solução é híbrida, apresentando tanto a localização dos dispositivos tecnológicos através de WiFi, quanto a contabilização dos demais ativos da instituição através de RFID.

Fonte: Autoria Própria.

2. Componentes e Recursos

2.1. Componentes de hardware

Para a montagem do dispositivo IoT, serão necessários os seguintes componentes:

- ESP32-S3
 - O ESP32-S3 é um XTensa LX7 MCU dual-core, capaz de rodar a 240 MHz. Além de 512 KB de SRAM interno, também possui a integração de 2.4 GHz , 802.11 b/g/n Wi-Fi e conectividade Bluetooth 5 (LE) que providencia suporte longa-distância. Tem 45 GPIOs programáveis e suporta vasta quantidade de periféricos. Comparado com o ESP32, ele suporta maiores SPI flash octais high-speed, e PSRAM com informação configurável e instruction cache.
- Light-emitting-diode (LED)
 - Um dispositivo semicondutor que emite luz quando a corrente flui por ele.
- Buzzer
 - Dispositivo de sinalização sonora via produção de zumbido através de meios mecânicos, eletromecânicos, ou piezoeletricos.
- TAG-RFID
 - Radio-frequency identification (RFID) usa campos eletromagnéticos para automaticamente identificar e rastrear etiquetas anexadas a objetos. Um sistema RFID consiste em um pequeno transponder de rádio, um receptor de rádio, e um transmissor. Quando acionado por um pulso de interrogação eletromagnético de um dispositivo leitor de RFID próximo, a etiqueta transmite dados digitais de volta ao leitor, geralmente um número de identificação. Esse número pode ser usado para rastrear estoque de mercadorias, por exemplo.
- RF SENSOR ISO15693
 - ISO/IEC 15693 é um regulation standard para tags de proximidade estabelecido para padronizar cartões que operam em 13,56 MHz. A frequência de 13,56 MHz é uma das frequências mundialmente utilizadas para categorias industriais, científicas, e médicas (ISM).
- Resistores
 - Um resistor é um componente elétrico passivo two-terminal que implementa resistência elétrica como um elemento de circuito. Nos circuitos eletrônicos, os resistores são usados para reduzir o fluxo de corrente, ajustar níveis de sinal, dividir tensões, polarizar elementos ativos, encerrar linhas de transmissão, *et cetera*.
- Jump wire

- Um jump wire é um ou mais fios elétricos em um cabo, com um conector ou pino em cada extremidade, que normalmente é utilizado para interligar os componentes de uma breadboard (ou outro protótipo/circuito de teste) internamente ou com outros equipamentos ou componentes de maneira que dispense soldagem. Jump wires são encaixados por meio da inserção de seus "end connectors" nos slots disponíveis em um breadboard.

2.2. Componentes externos

Para o projeto, foram utilizadas algumas tecnologias externas que ajudam na integração com o hardware, sendo elas:

- Computador:
 - Permite acesso à interface através do *FrontEnd*, permite verificações dos dados armazenados no *BackEnd*, e possibilita modificações nos dados armazenados nas *TAGs* utilizadas, bem como nos microcontroladores utilizados.
- Celular:
 - Possibilita o acesso à versão *mobile* da interface desenvolvida.

2.3. Requisitos de conectividade

O acesso da página web desenvolvida necessita de algumas etapas, que serão descritas a seguir:

- O ESP32-S3 deve estar conectado a uma fonte de energia, como um computador, para disponibilizar sua rede em WiFi;
- A partir disso, o computador que irá se conectar ao FrontEnd e BackEnd deste dispositivo deve acessar a rede WiFi do mesmo, permitindo assim a conexão e comunicação entre computador e dispositivos.

3. Guia de Montagem

O diagrama a seguir mostra as conexões entre os componentes descritos na seção 2.1, e suas respectivas portas, descritas na tabela 1.

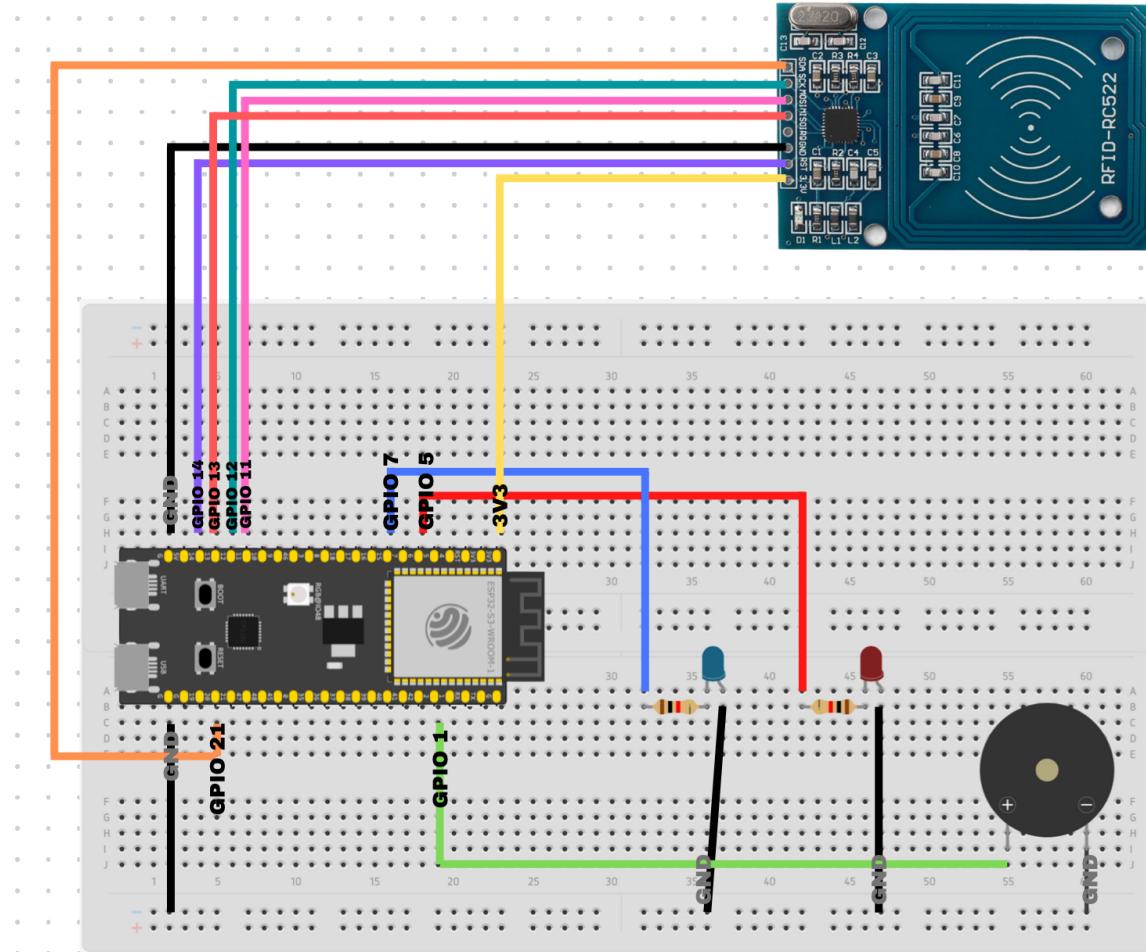


Figura 2: Exemplo de montagem do ESP32 S3 utilizando o RFID, com Leds que ajudam a entender o status da leitura. Fonte: Autoria Própria.

Tabela 1: Portas dos microcontroladores e sensores utilizados no protótipo do dispositivo.

Porta utilizada	Descrição da conexão
Porta GND, ESP32	Conecta na parte negativa da protoboard para energizar todos os fios conectados naquela sessão
Porta GPIO°21 ESP32	Conecta na porta SDA(responsável por receber e enviar dados para o RFID) do Leitor RFID
Porta GPIO°1 ESP32	Conecta na parte energizada da protoboard pelo GND
Porta GND, ESP32	Conecta na porta GND do Display LCD, para energizar o GND do Display
Porta GPIO°14 ESP32	Conecta na porta RST do Leitor RFID
Porta GPIO°13 ESP32	Conecta na porta SOZ do Leitor RFID
Porta GPIO°12 ESP32	Conecta na porta SCK do Leitor RFID
Porta GPIO°11 ESP32	Conecta na porta MOSI do Leitor RFID
Porta GPIO°7 ESP32	Conecta na protoboard com o resistor 10K que alimenta o positivo do LED, a outra perna do LED conecta na parte energizada do GND
Porta GPIO°5 ESP32	Conecta na protoboard com o resistor 10K que alimenta o positivo do LED, a outra perna do LED conecta na parte energizada do GND
Porta 3V3 ESP32	Conecta no 3V do leitor RFID, essas conexão serve para alimentar a energia do leitor RFID

Porta GND do leitor RFID	Conecta no GND da protoboard, para energizar a parte do GND do RFID
-----------------------------	--

4. Guia de Instalação

O passo-a-passo para instalar os dispositivos é:

1. Posicionar, acima de todas as portas das salas, o powerbank conectado a uma placa ESP-32 via cabo USB-C.

É importante ressaltar a necessidade de realizar tal posicionamento levando em consideração qualquer fator que possa interferir negativamente com o funcionamento a curto ou a longo prazo do equipamento. Exemplos incluem: umidade, chuva, temperaturas altas, acidentes durante limpeza da escola, e/ou vandalização.

Na imagem a seguir temos o exemplo de um local onde um ESP estará localizado, podendo ser acima das portas ou janelas:



Figuras 3 e 4: Apresentação da instalação dos microcontroladores no ambiente escolar. Fonte: Autoria Própria.

Outro ESP pode ser colocado próximo à porta do lado de fora da sala, como na imagem abaixo, o ESP32-S3 está próximo a porta, mas pode ser colocado acima da porta, na parte de madeira, de forma que dificulte o acesso dos alunos até ele.

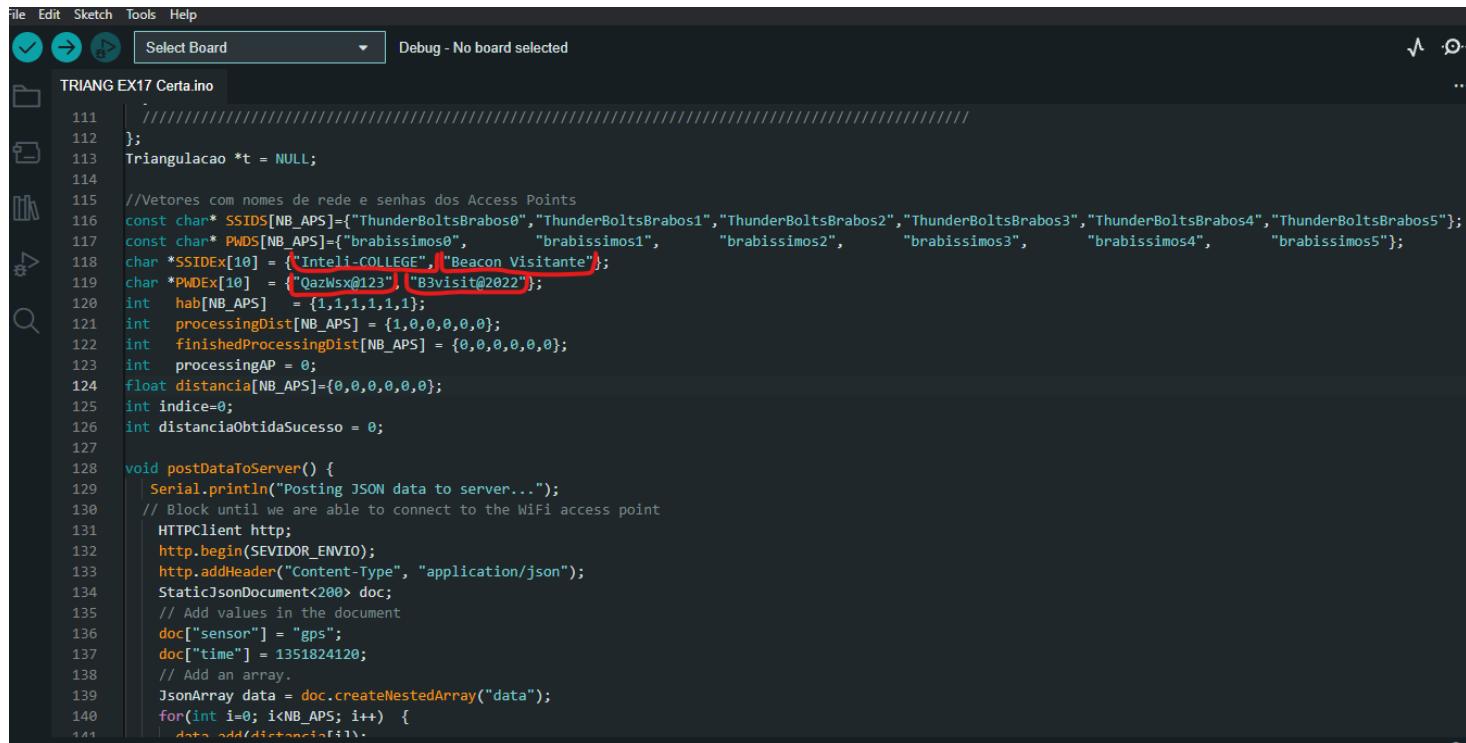


Figuras 5 e 6: Apresentação da instalação dos microcontroladores no ambiente escolar. Nessa imagem é possível observar uma possível instalação do ESP-32 ao lado de fora da porta da sala de aula que possui o ESP-32 apresentado nas figuras 3 e 4, e dessa forma, é possível haver um distanciamento permitindo a comunicação de conexão segura entre os ESP's. Fonte: Autoria Própria.

RECOMENDAÇÃO: recomendamos colocar uma identificação visual em cada ESP32 S3.

2. Para conectar à rede:

2.1. A conexão à rede é um procedimento integrado no código; o código deve ser executado, e, feito isso, os dispositivos estarão conectados à rede.



```

File Edit Sketch Tools Help
Select Board Debug - No board selected
TRIANG EX17 Certa.ino
111 //////////////////////////////////////////////////////////////////
112 };
113 Triangulacao *t = NULL;
114
115 //Vetores com nomes de rede e senhas dos Access Points
116 const char* SSIDS[NB_AP]={"ThunderBoltsBrabos0","ThunderBoltsBrabos1","ThunderBoltsBrabos2","ThunderBoltsBrabos3","ThunderBoltsBrabos4","ThunderBoltsBrabos5"};
117 const char* PWDS[NB_AP]={"brabissimos0", "brabissimos1", "brabissimos2", "brabissimos3", "brabissimos4", "brabissimos5"};
118 char *SSIDEx[10] = {"inteli_COLLEGE", "Beacon Visitante"};
119 char *PWDEx[10] = {"QazWsx@123", "B3visit@2022"};
120 int hab[NB_AP] = {1,1,1,1,1,1};
121 int processingDist[NB_AP] = {1,0,0,0,0,0};
122 int finishedProcessingDist[NB_AP] = {0,0,0,0,0,0};
123 int processingAP = 0;
124 float distancia[NB_AP]={0,0,0,0,0,0};
125 int indice=0;
126 int distanciaObtidaSucesso = 0;
127
128 void postDataToServer() {
129   Serial.println("Posting JSON data to server...");
130   // Block until we are able to connect to the WiFi access point
131   HTTPClient http;
132   http.begin(SEVIDOR_ENVIO);
133   http.addHeader("Content-Type", "application/json");
134   StaticJsonDocument<200> doc;
135   // Add values in the document
136   doc["sensor"] = "gps";
137   doc["time"] = 1351824120;
138   // Add an array.
139   JSONArray data = doc.createNestedArray("data");
140   for(int i=0; i<NB_AP; i++) {
141     data.add(distancia[i]);
142   }
143   http.print(doc);
144   if (http.begin() == HTTP_CODE_OK) {
145     http.end();
146   }
147 }
  
```

Figura 7: Na definição das variáveis SSIDEx e PWDEx, como apresentado na imagem acima, estão presentes o nome das redes e suas respectivas senhas as quais o ESP que estará presente nos computadores deverá se conectar para conectar-se também aos microcontroladores instalados no ambiente escolar. Fonte: Autoria Própria.

3. Para instalar software nos dispositivos:

O software desenvolvido pelo Thunderbolts pode ser instalado por meio do repositório oficial do GitHub (<https://github.com/2022M4T2-Inteli/Projeto2>) e editado por meio de uma IDE à sua escolha. (Recomenda-se o uso do Arduino IDE, pois foi a plataforma utilizada para o desenvolvimento deste projeto, e a qual será apresentada posteriormente neste documento.)

As propriedades, limites e alcances dos dispositivos em relação ao espaço destinado são:

- Aproximadamente 2m de margem de erro;
- Sugerido uma média de 1 dispositivo a cada 7m, para assegurar a conexão, a depender do tipo de espaço;
- Ou seja, uma média de 1,5 dispositivos por sala.

Portanto, para a instalação dos dispositivos na Beacon, será necessário que todos os dispositivos eletrônicos tenham um ESP32-S3. Terão aproximadamente 1,5 ESP's por sala, sendo 3 microcontroladores a cada 2 salas de aula, além dos ESP's que serão espalhados nos demais espaços da escola. Todos possuem cerca de 7 metros de distância entre eles, pois foi o distanciamento em que houve a melhor precisão de conexão dos microcontroladores e para possibilitar essas conexões os ESP's esses terão que estar conectados à rede wi-fi da escola para que, assim, possam comunicar-se entre si enviando as informações de estimativas de distância e posicionamento. Além disso, cada um deles precisará de uma bateria para alimentação de energia.

5. Guia de Configuração

Primeiramente, é necessário a instalação do ambiente de execução de servidores

5.1. Configuração dos softwares

Os softwares citados abaixo podem ser instalados pelo seu navegador.

5.1.1. Arduino IDE

Arduino IDE é um software utilizado para a construção do código em C++ presente no ESP32-S3. Para instalá-lo, siga o passo-a-passo abaixo.

1. Acesse o site <https://www.arduino.cc/en/software>

2. Dentro da caixa "DOWNLOAD OPTIONS", sinalizada na imagem abaixo pela seta, encontre o nome do seu sistema operacional e clique-o.

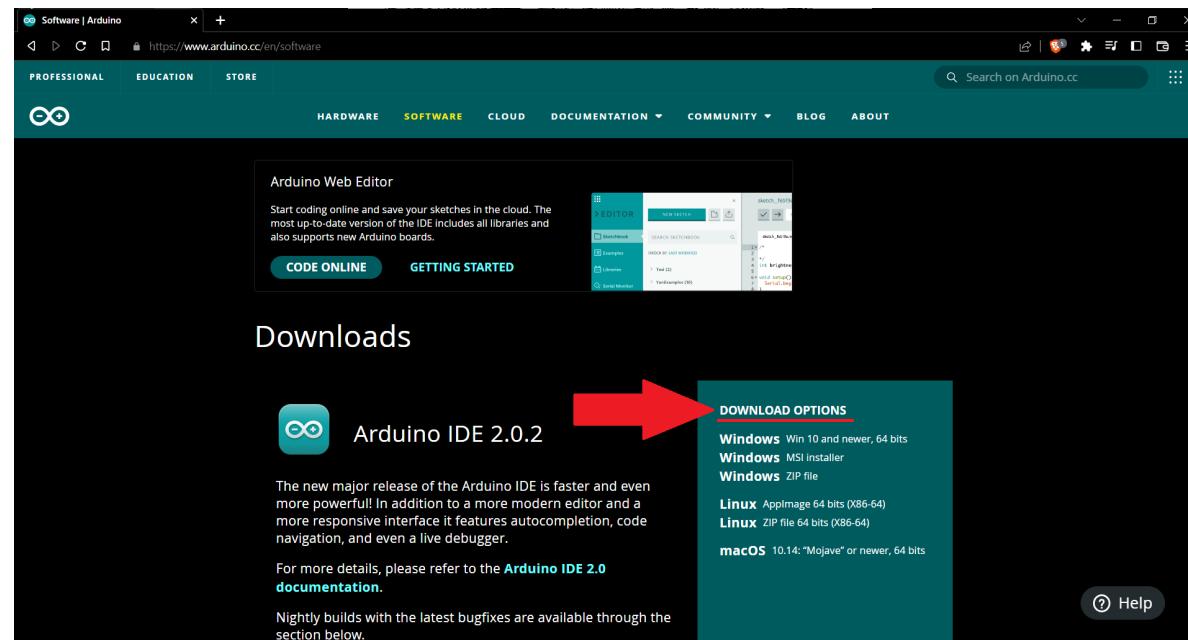


Figura 9: Site para instalação da IDE utilizada.

Fonte: <https://docs.arduino.cc/software/ide-v2/tutorials/getting-started/ide-v2-downloading-and-installing>

3. Clique "JUST DOWNLOAD" e execute o arquivo.

4. Após a instalação, será necessário adicionar a biblioteca do ESP32-S3. É recomendado a instalação de toda a biblioteca, posto que partes específicas dela, em alguns dispositivos, não funcionam como deveriam. Para isso, você deve seguir os seguintes passos:

Ao abrir o aplicativo, você encontrará esta tela:

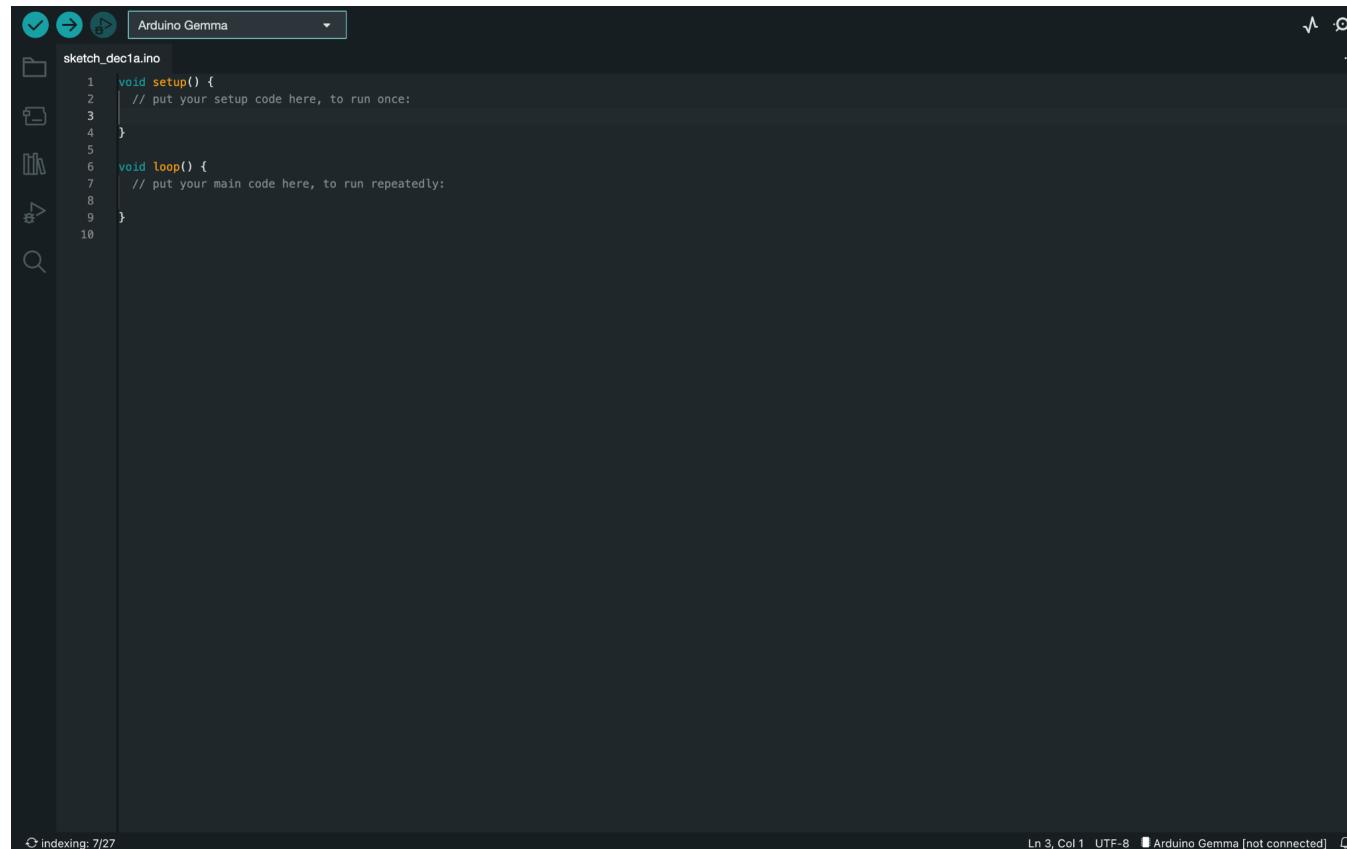


Figura 10: Imagem do Arduino IDE 2.0 com uma estrutura inicial de um código. Fonte: Autoria Própria.

Nas configurações do aplicativo, entre na seção de "Preferências":

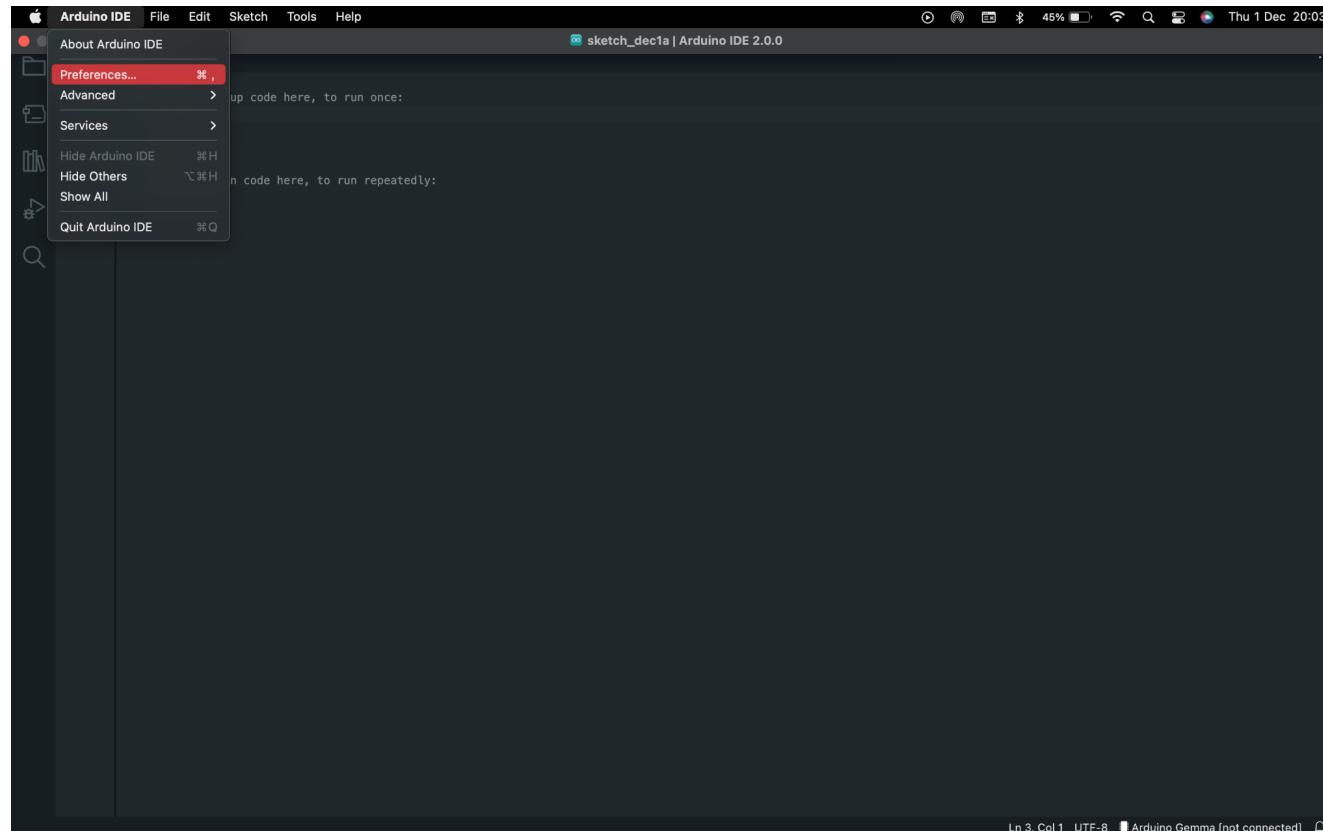


Figura 11: Imagem do Arduino IDE 2.0. Fonte: Autoria Própria.

Após entrar nas Preferências, você encontrará essa página

No espaço destacado abaixo, coloque esse link:

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

e, em seguida, aperte o botão "OK".

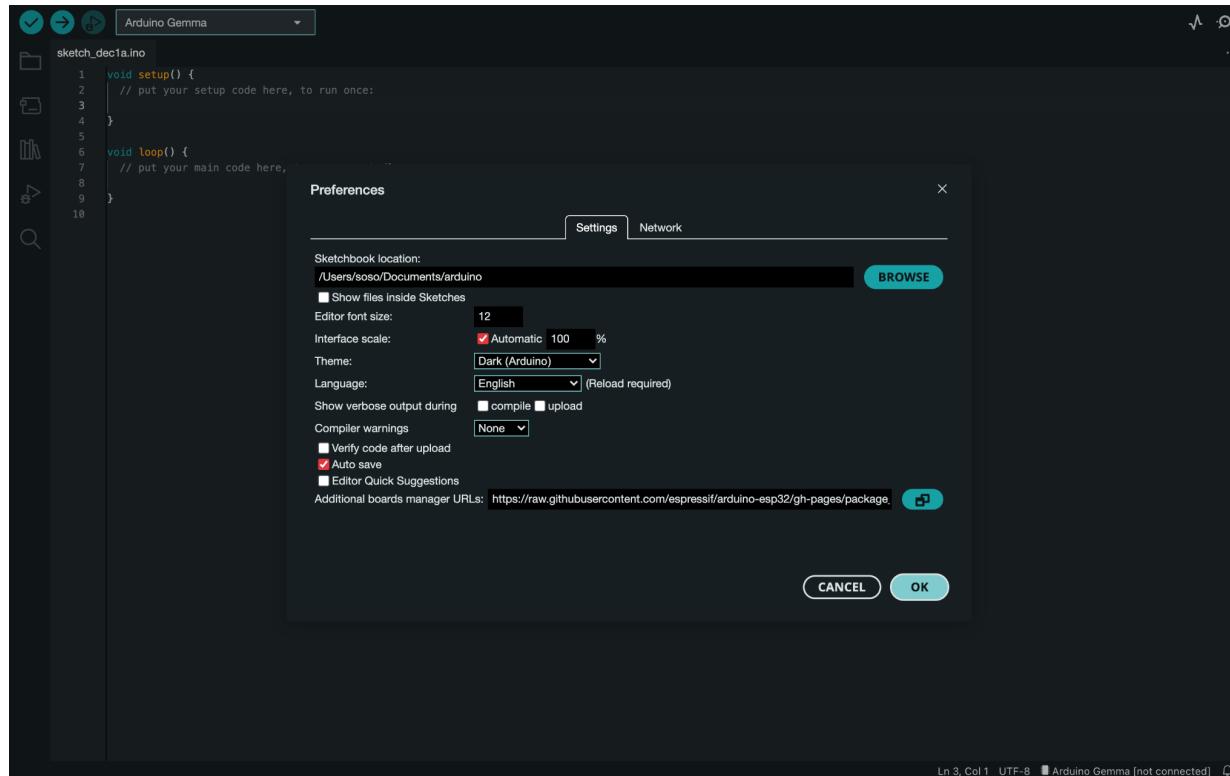


Figura 12: Imagem do Arduino IDE 2.0 com configurações. Fonte: Autoria Própria.

Feito isso, basta aguardar o aplicativo acabar a instalação. Ela pode levar de 20 a 40 minutos.

Com isso, você já possui o aplicativo para visualizar e modificar o código em C++, mas para isso, é necessário o código em si. Para baixá-lo, siga os passos abaixo:

Abra o repositório do nosso grupo no github por meio desse link: <https://github.com/2022M4T2-Inteli/Projeto2>

Clique no botão verde escrito "code".

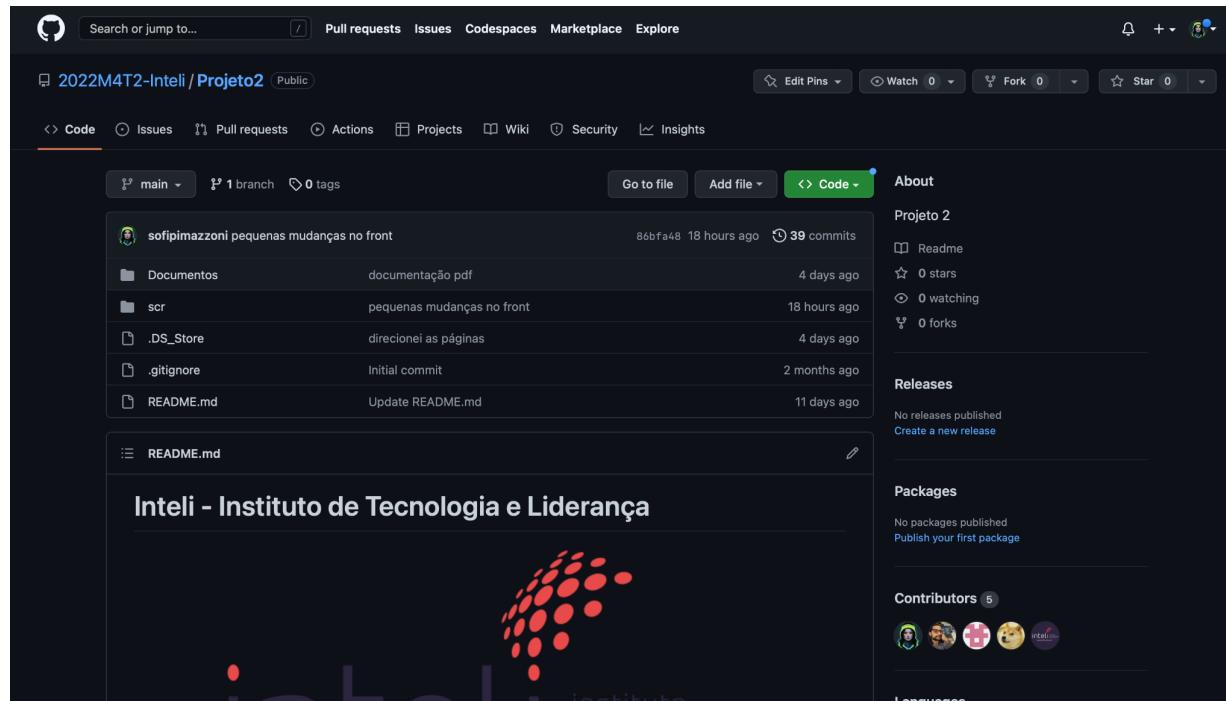


Figura 12: Imagem da página principal do Github do projeto. Fonte: Autoria Própria.

Após clicar, vão aparecer estas opções, e para adquirir os arquivos, clique em "Download ZIP".

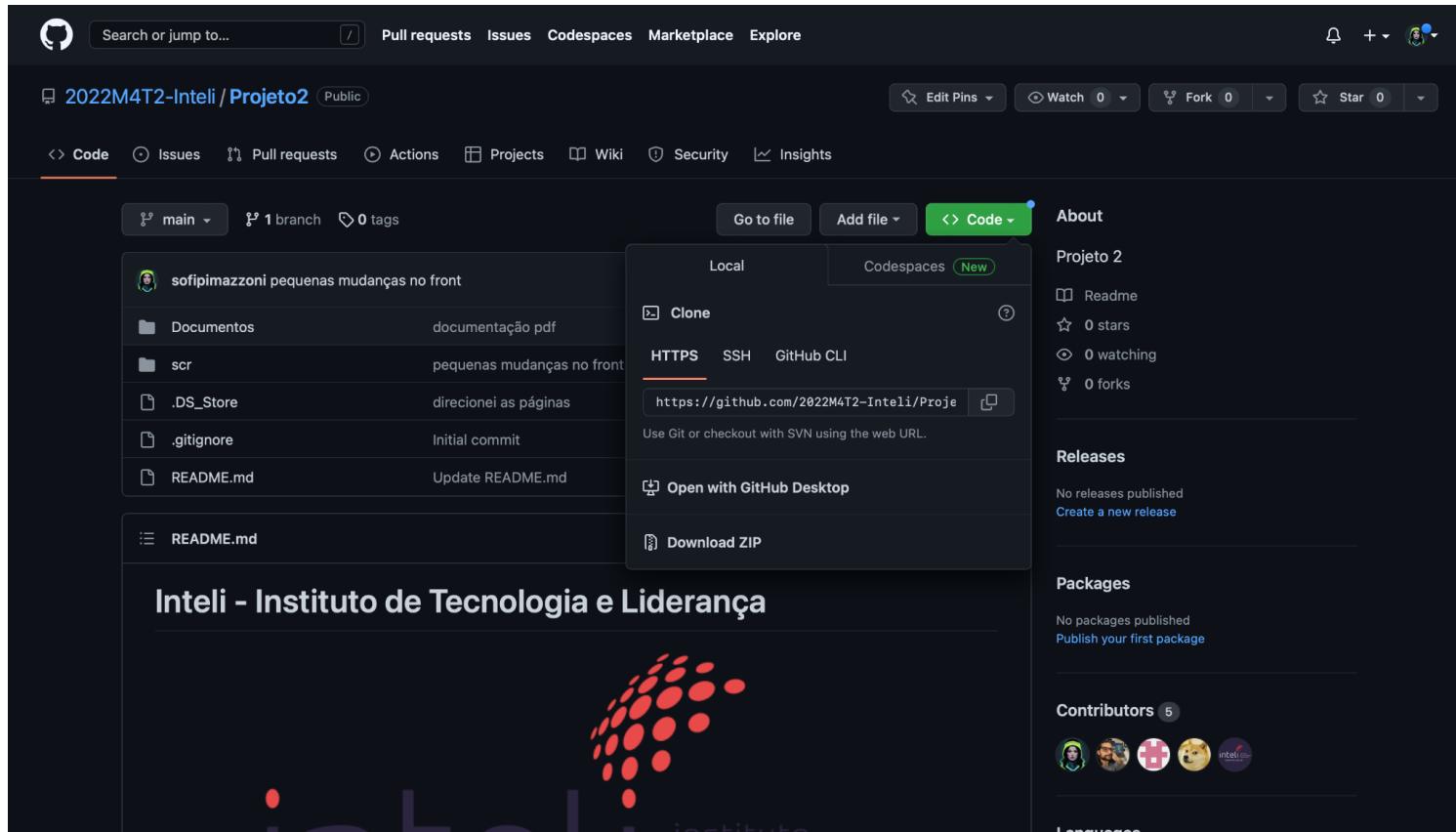


Figura 13: Imagem da página principal do Github do projeto. Fonte: Autoria Própria.

Com o ZIP instalado, basta descomprimir para que se torne uma pasta normal.

Dentro da pasta do projeto, você encontrará mais duas pastas: uma de documentos, e outra dos códigos. Acesse a pasta dos códigos ("scr").

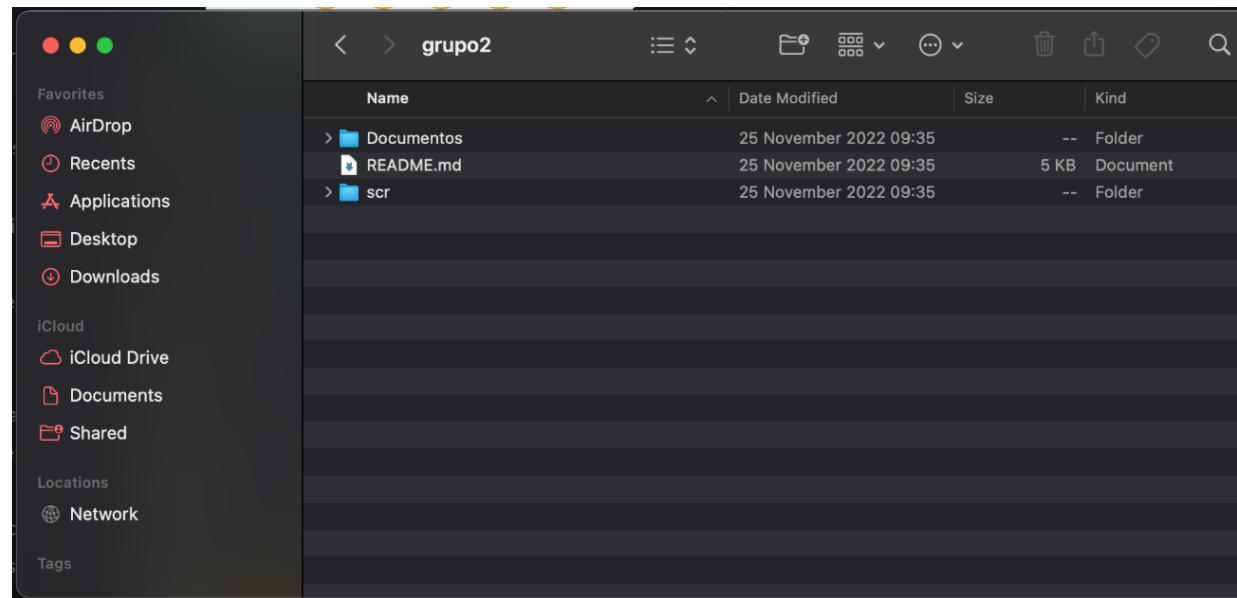


Figura 14: Imagem do Github do projeto pelo Github Desktop. Fonte: Autoria Própria.

Dentro da pasta "scr", existem mais três pastas, a que possui o arquivo que será usado no arduino IDE é a pasta "Circuito". Dentro dessa página você encontrará o arquivo do próprio Arduino IDE; basta abri-lo no aplicativo.

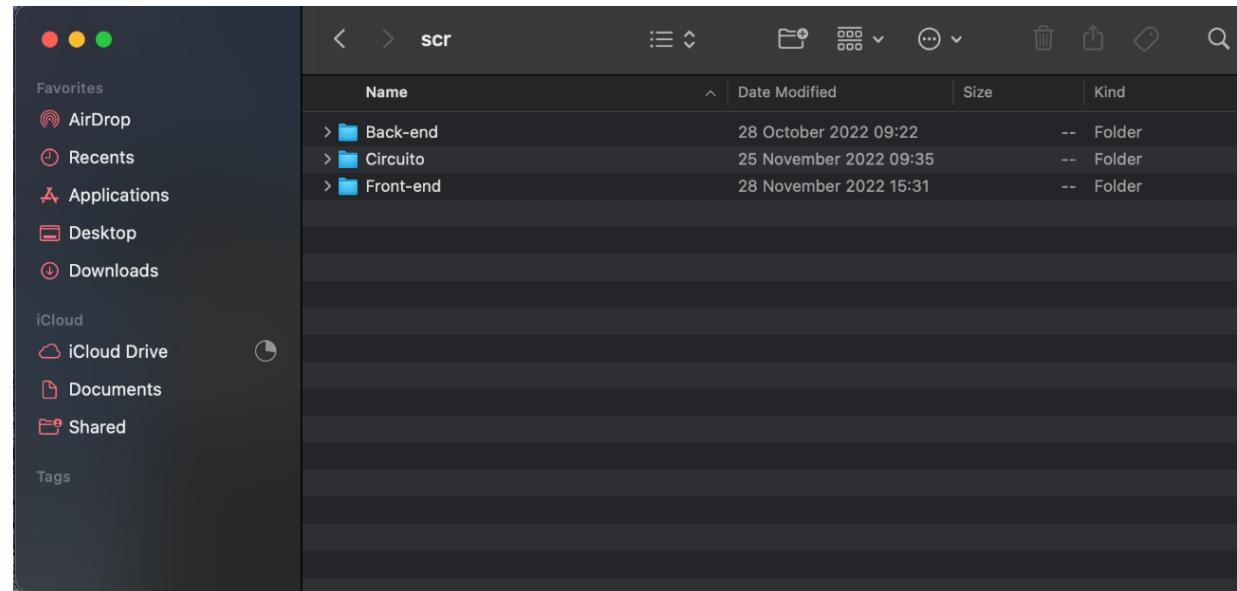


Figura 15: Imagem do Github do projeto pelo Github Desktop. Fonte: Autoria Própria.

OBSERVAÇÃO: Caso seja decidido por alterar ou atualizar o front-end, todos os seus arquivos estão organizados na pasta "Front-end". Recomendamos o editor de texto VS CODE para fazer tais alterações.

Além disso, qualquer alteração que seja necessária no back-end também pode ser feita pelo mesmo editor de texto, sendo que todos os arquivos estão localizados na pasta "Back-end".

5.1.2. Arquivo de definição da API

Nas configurações do seu computador abra a seção de Rede e Internet, vá para WiFi e abra as propriedades da rede em que o computador está conectado para adquirir o Endereço IPv4, como na imagem abaixo:

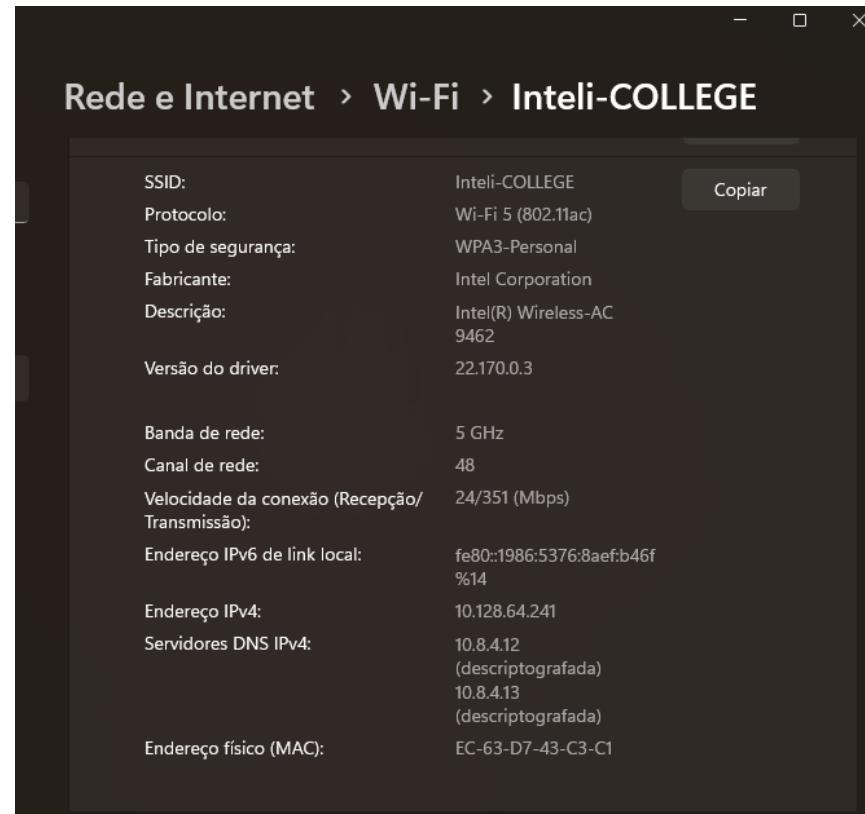
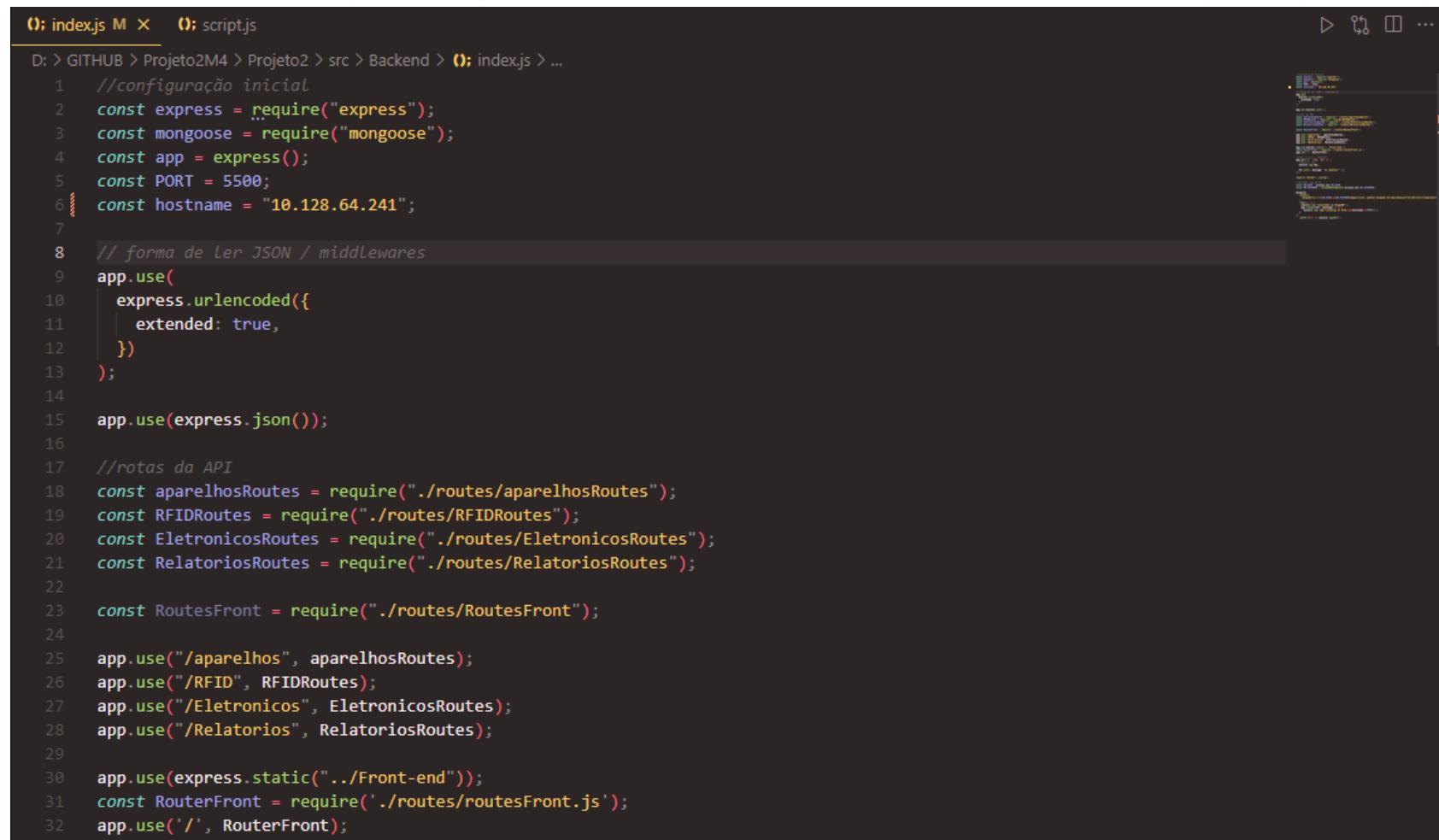


Figura 16: Imagem da configuração com informações da rede WiFi utilizada com informações necessárias para configurar o dispositivo. Fonte: Autoria Própria.

Após copiar o endereço como mencionado acima, vá até o arquivo “index.js” e coloque esse endereço na const “hostname = x”, como no exemplo abaixo, para definir o endereço de acesso e subir o servidor no seu computador pelo Node.js.



```

  0: index.js M X  0: script.js
D: > GITHUB > Projeto2M4 > Projeto2 > src > Backend > 0: index.js > ...
1  //configuração inicial
2  const express = require("express");
3  const mongoose = require("mongoose");
4  const app = express();
5  const PORT = 5500;
6  const hostname = "10.128.64.241";
7
8  // forma de ler JSON / middlewares
9  app.use(
10    express.urlencoded({
11      extended: true,
12    })
13  );
14
15 app.use(express.json());
16
17 //rotas da API
18 const aparelhosRoutes = require("./routes/aparelhosRoutes");
19 const RFIDRoutes = require("./routes/RFIDRoutes");
20 const EletronicosRoutes = require("./routes/EletronicosRoutes");
21 const RelatoriosRoutes = require("./routes/RelatoriosRoutes");
22
23 const RoutesFront = require("./routes/RoutesFront");
24
25 app.use("/aparelhos", aparelhosRoutes);
26 app.use("/RFID", RFIDRoutes);
27 app.use("/Eletronicos", EletronicosRoutes);
28 app.use("/Relatorios", RelatoriosRoutes);
29
30 app.use(express.static("../Front-end"));
31 const RouterFront = require('./routes/routesFront.js');
32 app.use('/', RouterFront);
  
```

Figura 17: Imagem do código com a informação obtida no passo anterior. Fonte: Autoria Própria.

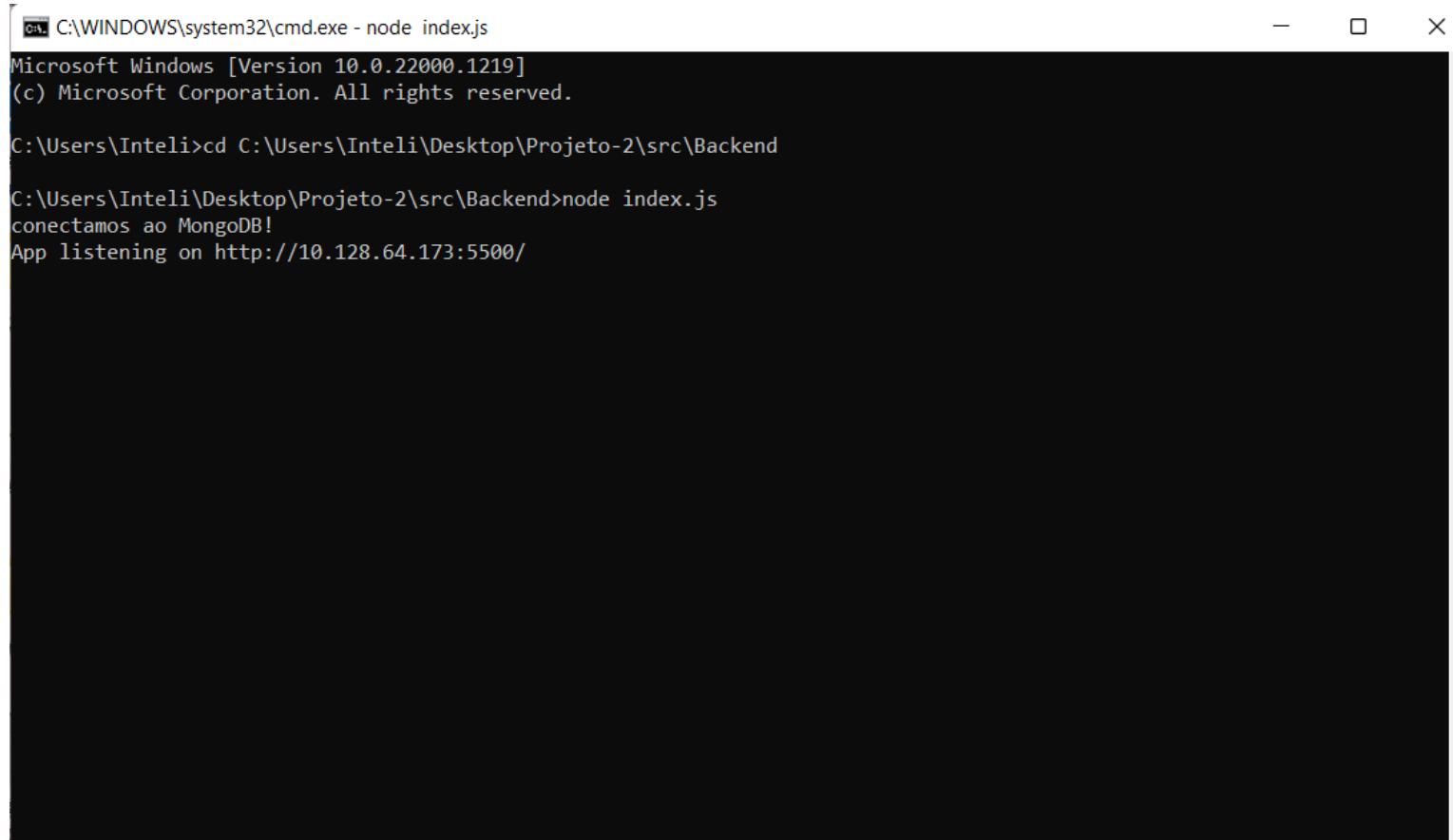
5.1.3. Node.js

Faça o download do ambiente de execução do Node.js para executar o arquivo JavaScript que contenha a inicialização do servidor local no computador e instale pelo prompt de comando a extensão “express”.



Figura 18: Imagem da página principal do node.js. Fonte: <https://nodejs.org/en/>.

Abra o prompt de comando do seu computador e copie o caminho do endereço onde está localizado o arquivo “index.js” e execute a linha de comando “node index.js”. Após a execução, o endereço de IP definido no arquivo JavaScript que estará retornando a página da interface ao subir o servidor aparecerá na tela com a requisição HTTP, endereço e porta de saída, como no exemplo abaixo:



```
C:\WINDOWS\system32\cmd.exe - node index.js
Microsoft Windows [Version 10.0.22000.1219]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Inteli>cd C:\Users\Inteli\Desktop\Projeto-2\src\Backend

C:\Users\Inteli\Desktop\Projeto-2\src\Backend>node index.js
conectamos ao MongoDB!
App listening on http://10.128.64.173:5500/
```

Figura 19: Imagem do prompt de comando com exemplo do caminho utilizado do arquivo. Fonte: Autoria Própria.

Copie e cole o endereço no seu navegador para acessar a página da interface criada.

5.2. Configuração do código no Arduino

Essa configuração é necessária para que seja possível rodar o código do Arduino no ambiente desejado, sendo necessário modificar apenas o nome do wi-fi, sua quantidade de acordo com os ESPs e a senha. Cada dispositivo ESP32 S3 precisa ter um wi-fi específico, mas todos podem compartilhar a mesma senha.

O wi-fi pode ser modificado na linha 7 (char* SSIDS[2] = {"wi-fi do primeiro ESP32 S3", "wi-fi do segundo ESP32 S3"}) e assim por diante.

- O número 2 entre colchetes representa a quantidade de ESP32 S3 conectados ao wi-fi. É necessário modificá-lo de acordo com a quantidade de ESPs cadastrados

A senha pode ser modificada na linha 8 (char* PWD = {"senha do primeiro ESP32 S3", "senha do segundo ESP32 S3"}) e assim por diante.

- Caso seja decidido usar a mesma senha para todos os ESPs, basta colocar da maneira como está na imagem abaixo.

A seguir, as imagens referentes às configurações de WiFi e senhas:

```
1 #include <WiFi.h>
2
3 //Mudar o número para determinado AccessPoint
4 #define APN 1
5
6 //Vetores com nomes de rede e senhas dos Access Points
7 char* SSIDS[5]={ "ThunderBoltsBrabos1", "ThunderBoltsBrabos2", "ThunderBoltsBrabos3", "ThunderBoltsBrabos4", "ThunderBoltsBrabos5"};
8 char* PWD="brabissimos";
9
10 char strIP[50] = "";
11 void setup() {
12     Serial.begin(115200);
13     //WiFi.mode(WIFI_STA);
14
15     WiFi.softAP(SSIDS[APN],PWD, 1, 0, 10, true);
16     IPAddress IP = WiFi.softAPIP();
17     strcpy(strIP,IP.toString().c_str());
18     DadosConexao();
19
20 }
21 void loop() {
```

Figura 20: Imagem com código utilizado nos ESP32 S3. Em vermelho, os campos de SSID, referente ao nome do WiFi de cada dispositivo, e PWD, referente a uma senha única para todos os dispositivos. Fonte: Autoria Própria.

6. Guia de Operação

6.1. Introdução à interface

Primeiramente, para que a interface tenha seu funcionamento tal qual foi projetada, é necessário que os microcontroladores embarcados estejam devidamente instalados e conectados à rede do ambiente escolar e à energia. Além disso, é necessário que as Tags RFID estejam todas devidamente implementadas nos ativos físicos da escola.

Após a finalização da ambientação física da escola e dispositivos, será possível entrar na plataforma web desenvolvida a qual trará inicialmente uma tela de login. Após essa tela, será possível visualizar o menu da interface. O menu é a página que contém todas as funcionalidades do sistema.



Figura 21: Imagem do Frontend do projeto. Fonte: Autoria Própria.

6.2. Cadastro de novos ativos e de empréstimo

Os dois primeiros tópicos não foram desenvolvidos. Eles estão como sugestão para o cliente implementar, como dito no final da seção 1.4.4 da documentação oficial (IoT Doc). Por conta disso, foi decidido deixar seus espaços na interface caso haja futuras implementações.

6.3. Relatórios

Ao entrar na terceira aba "relatórios", será possível visualizar algumas estatísticas de meses anteriores em relação aos ativos eletrônicos.

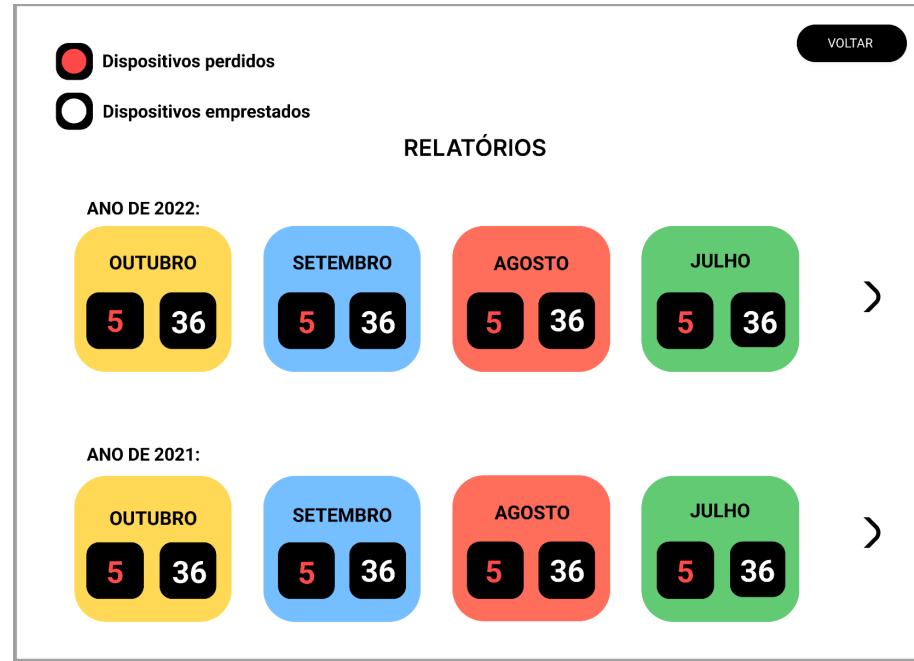


Figura 22: Imagem do Frontend de relatórios do projeto. Fonte: Autoria Própria.

No canto superior esquerdo, a legenda mostra o que os números dentro dos quadrados significam.

- Em vermelho, o número de equipamentos perdidos por mês
- Em branco, o número de equipamentos emprestados por mês

A interface foi pensada para que o usuário tivesse estatísticas do ano atual e do ano anterior, para que pudesse fazer análises e comparações. Além disso, ao clicar nas setas no lado direito, o usuário tem acesso aos relatórios dos outros meses.

As informações de equipamentos perdidos e emprestados por mês serão coletadas do banco de dados usado pelo cliente. Caso seja optado por não aplicar a nossa sugestão (seção 6.2), o cliente terá que integrar a nossa interface com o seu sistema.

6.4. Equipamentos cadastrados

Ao entrar na quarta aba "equipamentos cadastrados", será possível visualizar uma tabela com os equipamentos cadastrados pelo sistema.

Atualmente a tabela possui apenas 6 colunas, sendo elas: "ID eletrônico", "Número patrimônio", "Número de série", "Modelo", "Cor" e "Localização".

Nessa página, é possível recuperar os dados do inventário do próprio sistema da escola e preencher a tabela com esses por meio da inserção e alteração das informações no banco de dados. Assim haverá a possibilidade de ter um controle maior dos ativos por meio da utilização da interface juntamente com o inventário da empresa.

Na coluna "Localização" haverá um auto incremento das distâncias captadas e, assim, aproximá-las às áreas das salas em relação à posição definida no mapa como central. Dessa forma, a visualização da localização terá uma determinada precisão em relação ao posicionamento dos dispositivos dentro do ambiente escolar, e apesar desta precisão, haverá também uma pequena margem de erro relacionada à distância armazenada nessas colunas, e dessa forma, recomenda-se que haja uma mobilização para o local indicado pela tabela.

Abaixo, é possível visualizar essa tabela na interface:

[VOLTAR](#)

EQUIPAMENTOS CADASTRADOS

ID eletrônico	Número patrimônio	Número de série	Modelo	Cor	Localização
085767	000000	000000	ChromeBook	Cinza	Sala 1
085767	000000	000000	ChromeBook	Cinza	Sala 1
085767	000000	000000	ChromeBook	Cinza	Sala 1
085767	000000	000000	ChromeBook	Cinza	Sala 1

[GERAR EXCEL](#)

Figura 23: Imagem do Frontend de equipamentos cadastrados do projeto. Fonte: Autoria Própria.

6.5. Equipamentos emprestados

Ao clicar no botão de “equipamentos emprestados” da página inicial será possível visualizar uma tabela com as colunas “ID”, “Nome”, “Data retirada” e “Data devolução”, na interface, que apresentam respectivamente o ID do equipamento emprestado, o nome da pessoa que pegou o dispositivo emprestado, a data de retirada do dispositivo e a sua respectiva data de devolução. É importante destacar que o funcionamento dessa tabela é concomitante ao funcionamento da tabela na interface apresentada na seção 6.4 do Manual, ou seja, ela poderá estar integrada com o sistema da escola e as informações serão passadas para o banco de dados para que seja integrada à interface. Abaixo, é apresentada a tabela dos equipamentos emprestados:

[VOLTAR](#)

EQUIPAMENTOS EMPRESTADOS

ID	Nome	Data retirada	Data devolução
085767	Jean Oliveira	20/10/2022	20/10/2022
094733	Sofia Pimazzoni	20/10/2022	20/10/2022
076325	Mariana Silva	20/10/2022	20/10/2022
035279	Gustavo Monteiro	20/10/2022	20/10/2022

Figura 24: Imagem do Frontend de equipamentos emprestados do projeto. Fonte: Autoria Própria.

6.6. Mapa

Ao clicar no botão “Mapa” do menu, será possível visualizar o mapa do ambiente escolar, sendo possível escolher qual ativo você quer visualizar (aparelhos eletrônicos, cadeiras ou mesas). Além disso, é possível pesquisar por um ativo específico, que ao ser identificado no sistema, irá aparecer logo abaixo da barra de pesquisa.



Figura 25: Imagem do Frontend da visualização por mapas do projeto. Fonte: Autoria Própria.

A parte "Alertas", no retângulo escuro, irá aparecer apenas se for identificado que um ativo saiu do perímetro escolar, seja ele eletrônico ou não. Essa informação será lida por um sensor que ficará na porta da escola, e será enviada para o banco de dados. Mais uma vez, o cliente poderá decidir em aplicar as nossas sugestões de cadastro (seção 6.2) ou usar o seu sistema próprio. O sistema, independente de qual for, tem a função de enviar essa informação para a nossa interface, alertando o usuário de que um ativo saiu da escola.

Ainda no retângulo escuro, a parte "Ativos Perdidos" irá aparecer somente se o sistema reconhecer que um ativo eletrônico não foi devolvido na data prevista. Para fazer essa identificação, ele terá que chegar no banco de dados escolhido pelo cliente na data que estava prevista para a devolução do ativo eletrônico e, se ela passar, o sistema irá reconhecer e mostrar para o usuário.

No mapa em si, o usuário possui 3 opções: visualizar a localização dos aparelhos eletrônicos, das cadeiras ou das mesas. A localização dos aparelhos eletrônicos será identificada pelos ESP32 S3, que irão enviar os dados coletados para um Back-end. Esse sistema se comunicará como Front-end, mostrando a localização dos aparelhos por meio de bolinhas no mapa. Já as cadeiras e as mesas, serão identificadas por meio do RFID. Cada ativo terá uma etiqueta que, quando lida, irá passar essa informação para o Front-end, da mesma forma que com os ativos eletrônicos e também será representada por bolinhas.

O usuário tem a opção de escolher qual tipo de ativo ele quer visualizar a localização.

6.7. Lista

Ao clicar no botão "Lista" no menu, será possível visualizar os ambientes escolares em lista. Essa é uma forma alternativa de representação do mapa, portanto, possui a maioria de suas funcionalidades, sendo elas todas presentes no retângulo na esquerda.



PESQUEZE POR UM ATIVO ESPECÍFICO

Search:

VOLTAR

Lugar	Dispositivos
Sala 1	4 >
Sala 2	3 >
Sala 3	4 >

GERAR EXCEL

ALERTAS

ID: 90324854
MacBook
Sofia Pimazzoni - (00/00/0000 - 00/00/0000)

ATIVOS PERDIDOS

ID: 90324854
MacBook
Sofia Pimazzoni - (00/00/0000 - 00/00/0000)

ID: 90324854
MacBook
Sofia Pimazzoni - (00/00/0000 - 00/00/0000)

Figura 26: Imagem do Frontend de visualização por lista do projeto. Fonte: Autoria Própria.

A tabela possui 2 colunas: "Lugar" e "Aparelhos". A primeira identifica os ambientes da escola, e a segunda, quantos ativos possuem nela.

Para obter essas informações, é pré-definido ao sistema o perímetro dos ambientes, dessa forma, ele consegue saber quantos ativos estão presentes em cada lugar e se comunicar com o Front-end.

Do lado direito da tabela, existem algumas setas. Essas são clicáveis e levam o usuário para uma outra página, que é possível visualizar abaixo:

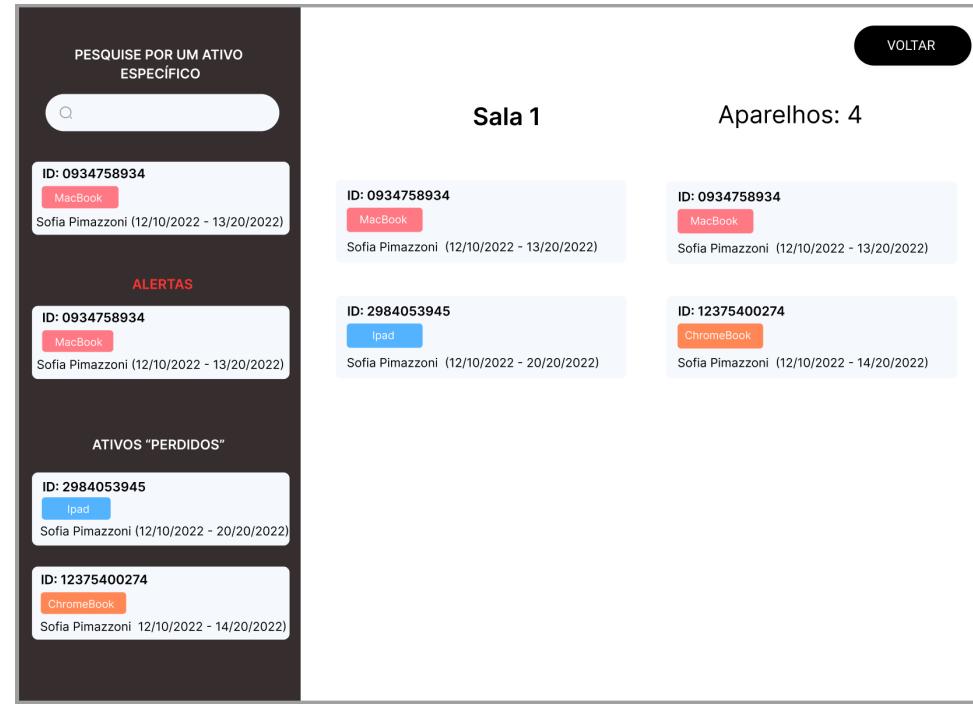


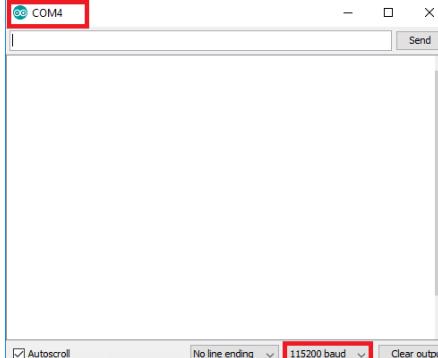
Figura 27: Imagem do Frontend com ambientes e equipamentos por sala do projeto. Fonte: Autoria Própria.

Nessa nova página, é possível visualizar especificamente, cada ativo que está presente naquele ambiente.

7. Troubleshooting

Tabela 2: Problemas possíveis do projeto e suas soluções.

#	Problema	Possível solução
1	Falta de bateria.	Trocar ou recarregar bateria.
2	ESP32-S3 (Anchor) ou (TAG) inoperante mesmo com bateria.	Trocar dispositivo ESP32-S3, e adicionar seu código respectivo.
3	ESP32-S3 (FTM) falha ao se comunicar com ESP32-S3 (Anchor).	Trocar dispositivo ESP32-S3, e adicionar seu código respectivo.
4	Exposição de componente(s) a água ou a umidade.	O componente exposto deve ser desconectado e secado o quanto antes. Caso seriamente danificado, ele deve ser substituído para prevenir curtos.
5	Arduino IDE não está funcionando com o ESP32-S3.	Utilizar PlatformIO IDE.

	Estado do equipamento adulterado por transeuntes ou por limpeza incorreta.	Reversão do equipamento ao estado prévio à adulteração, o que inclui, caso cabível, substituição de peça(s) danificada(s). Se possível, solicitar aos adulteradores futura não-interferência com o equipamento.
6		
7	Queda do equipamento.	O equipamento deve ser reposicionado, e o motivo da queda descoberto e prevenido. Caso algum componente quebre com a queda, ele deve ser reposto.
8	Sistema inoperacional após queda de energia / de WiFi.	O sistema deve ser reiniciado.
9	Cabo com interior exposto mas ainda funcional.	Cobrir parte exposta com fita isolante.
10	"Serial Monitor" do Arduino IDE não funciona	<p>Certificar que o port e o baud rate selecionados são os corretos.</p>  <p><i>imagem: randomnerdtutorials</i></p>

8. Créditos

Alan Rozensztajn Schipper - Arquitetura de solução e Guia de configuração

Daniel Quintão Dávila - Componentes e Recursos e Guia de configuração

Gustavo Monteiro - Introdução, Componentes e Recursos, Arquitetura de software e Troubleshooting

Igor Scapin Francisco Garcia - Introdução e Guia de montagem

Jean Lucas Rothstein Machado - Componentes e Recursos, Guia de configuração, Guia de instalação e Guia de montagem

Mariana Silva de Paula - Introdução, Componentes e Recursos e Guia de instalação

Sofia Moreiras Pimazzoni - Guia de montagem, Guia de configuração e Créditos