



Manual de Instruções

**THUNDERBOLTS
BEACON SCHOOL**

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.

Índice

1. Introdução	3
1.1. Solução	3
1.2. Arquitetura da Solução	3
2. Componentes e Recursos	4
2.1. Componentes de hardware	4
2.2. Componentes externos	4
2.3. Requisitos de conectividade	4
3. Guia de Montagem	5
4. Guia de Instalação	6
5. Guia de Configuração	7
6. Guia de Operação	8
7. Troubleshooting	9
8. Créditos	10

1. Introdução

1.1. Solução (sprint 3)

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 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 (sprint 3)

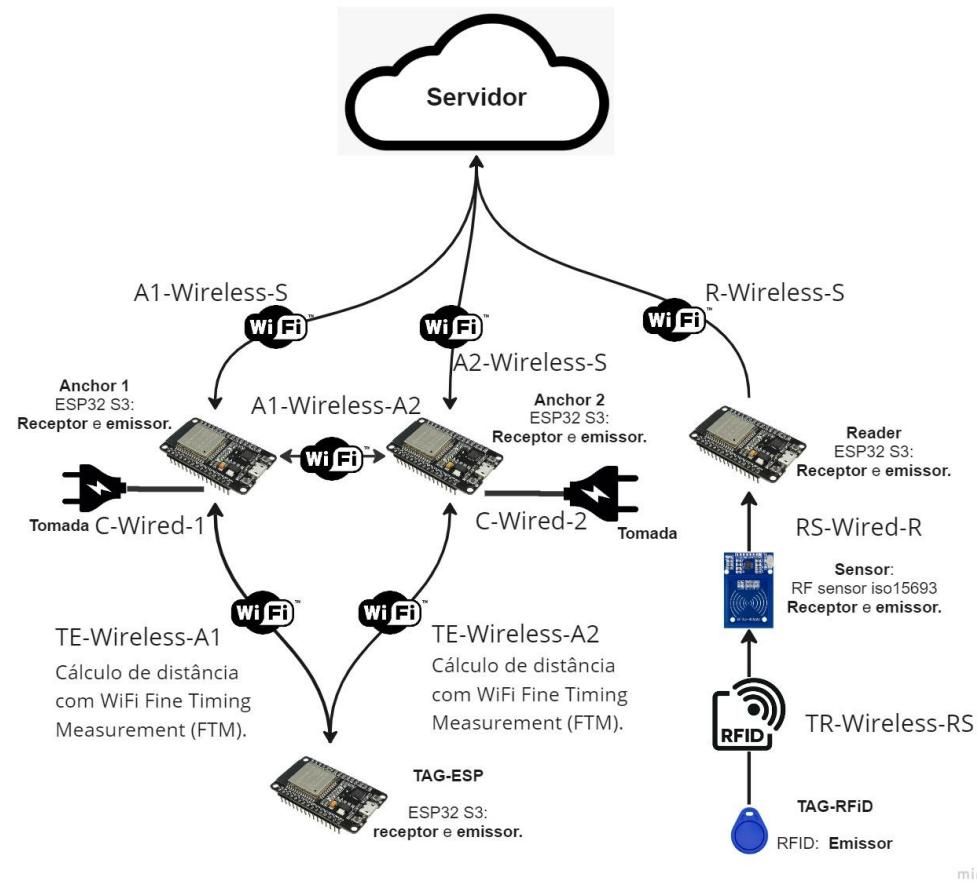


Figura 1: Diagrama de autoria própria, representando a terceira versão da Arquitetura de Hardware do Sistema. A solução apresenta a localização dos dispositivos tecnológicos através de WiFi, e a solução para contabilizar os demais ativos da instituição através de RFID.

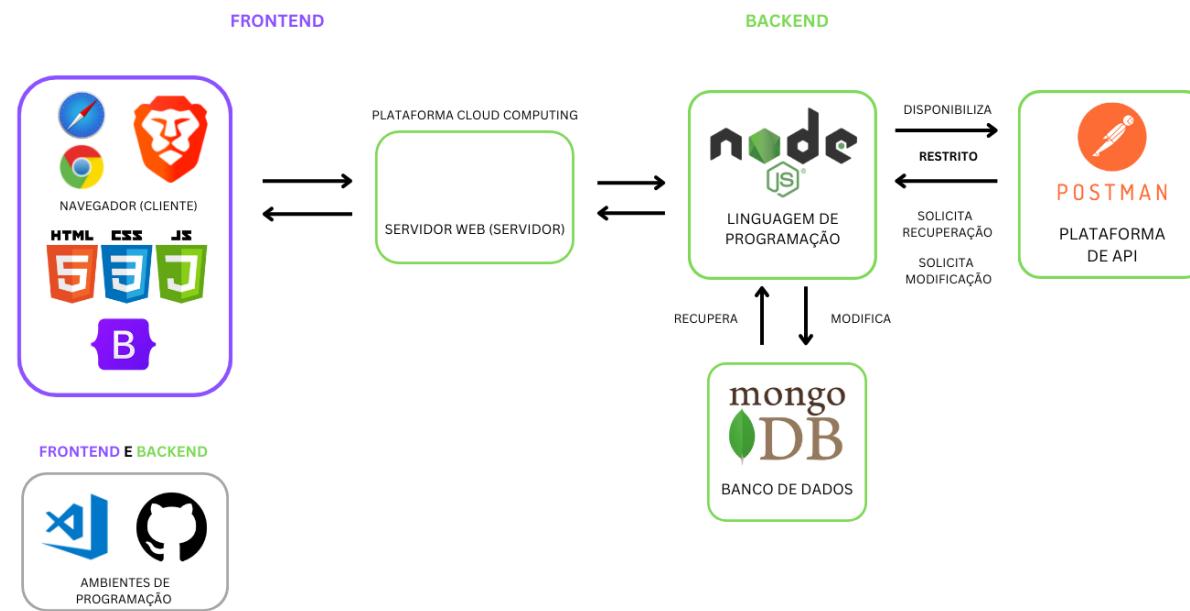


Figura 2: Diagrama de autoria própria, representando a terceira versão da Arquitetura de Software do Sistema. A solução apresenta a localização dos dispositivos tecnológicos através de WiFi, e a solução para contabilizar os demais ativos da instituição através de RFID.

2. Componentes e Recursos

(sprint 3)

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 piezoelétricos.
- 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.

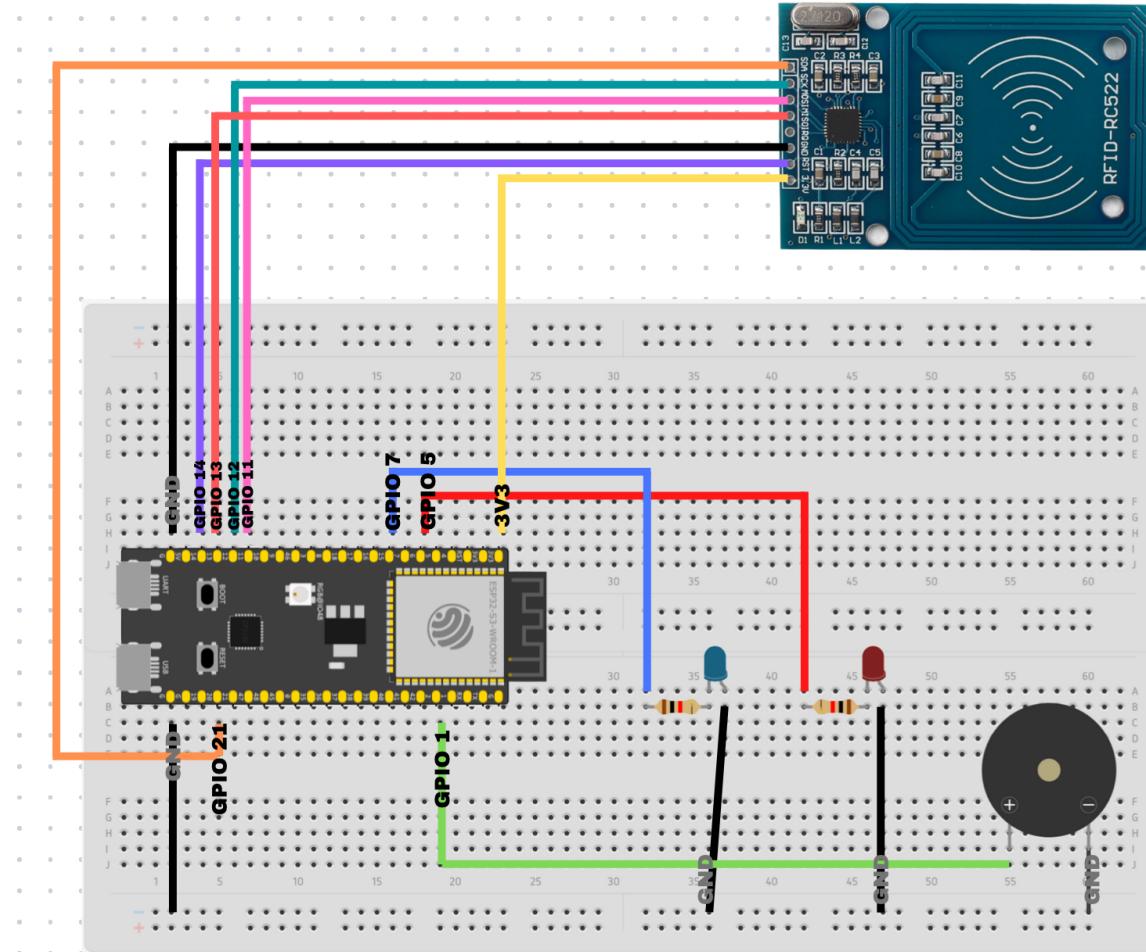


Figura 2: Exemplo de montagem do ESP32 S3 utilizando o RFID, com Leds que ajudam a entender o status da leitura.

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, essa 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

(sprint 4)

Descreva passo-a-passo como instalar os dispositivos IoT no espaço físico adequado, conectando-os à rede, de acordo com o que foi levantado com seu parceiro de negócios.

Não deixe de especificar propriedades, limites e alcances dos dispositivos em relação ao espaço destinado.

Especifique também como instalar softwares nos dispositivos.

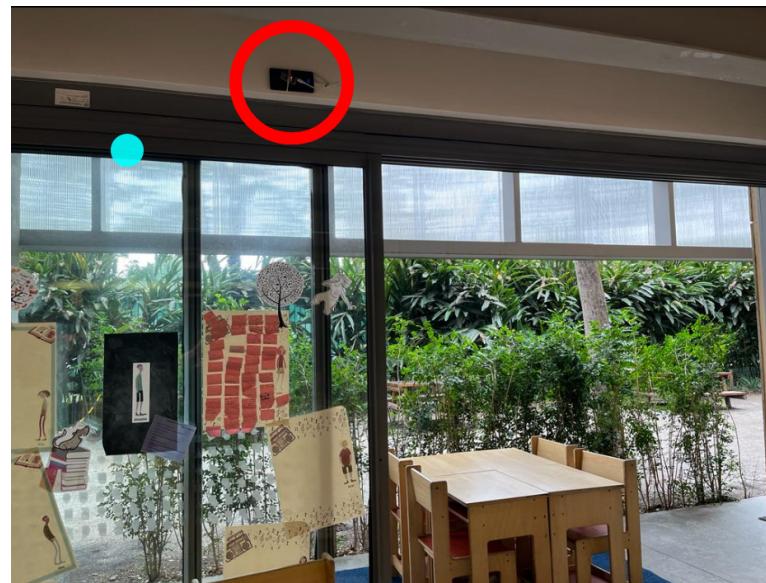
Utilize fotografias, prints de tela e/ou desenhos técnicos para ilustrar o processo de instalação.

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.

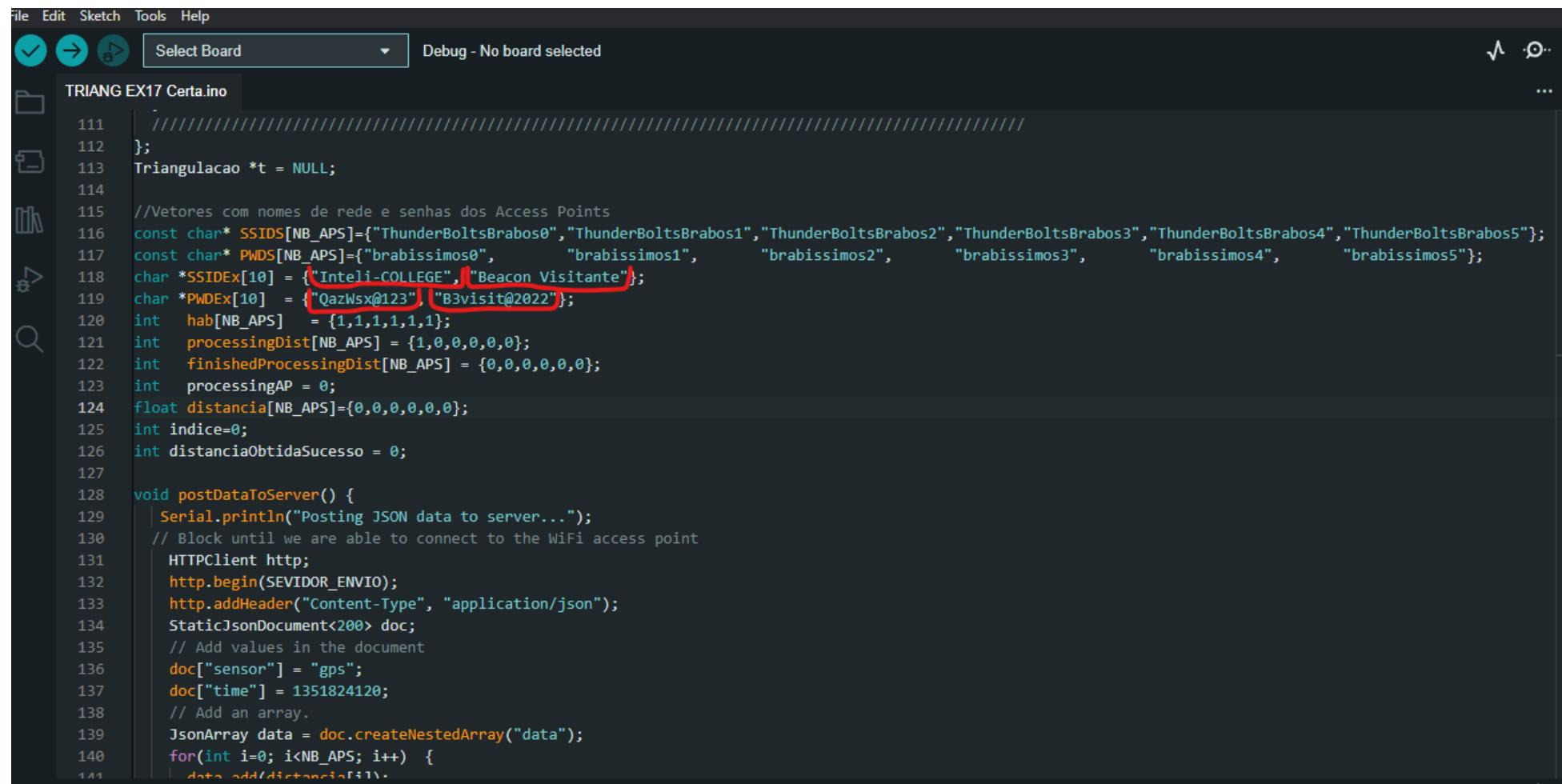
Outro ESP pode ser colocado próximo à porta do lado de fora da sala, como na imagem abaixo, o ESP32-S3 está próximo a porta, mas pode ser colocado acima da porta, na parte de madeira, de forma que dificulte o acesso dos alunos até ele.



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.

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] = {"0azWsx@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 }
```

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.

```

5         WiFi.removeEvent(ARDUINO_EVENT_WIFI_FTM_REPORT);
6         WiFi.disconnect();
7         esp_wifi_ftm_end_session();
8     }
9     if(processingAP == NB_AP){
10        processingAP = 0;
11        Serial.println("inserindo as distancias 1");
12        t->putDistancia(0,distancia[0]);
13        t->putDistancia(1,distancia[1]);
14        t->putDistancia(2,distancia[2]);
15        Serial.println("inserindo as distancias 2");
16        Serial.println(t->pontoXMedio(),t->pontoYMedio());
17        Serial.println("Mostrou coordenadas ");
18        Serial.println("Conectar na internet e enviar dados para o servidor!");
19        ConectarEnviarRede(0);
20    }
21    processingDist[processingAP] = 1;
22 }
23 }
24 }
25 }
```

Figura 8: Na função “ConectarEnviarRede()”, apresentada acima, é necessário definir a posição da rede, na lista apresentada na figura 7, a ser configurada pelo ESP presente no computador para conectar-se à rede do ambiente escolar e aos demais microcontroladores presentes nas salas de aula.

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 desses 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

(sprint 4)

Descreva passo-a-passo como configurar os dispositivos IoT utilizando os equipamentos devidos (ex. smartphone/computador acessando o servidor embarcado ou a página na nuvem).

Utilize fotografias, prints de tela e/ou desenhos técnicos para ilustrar o processo de configuração.

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

5.1. Configuração dos softwares

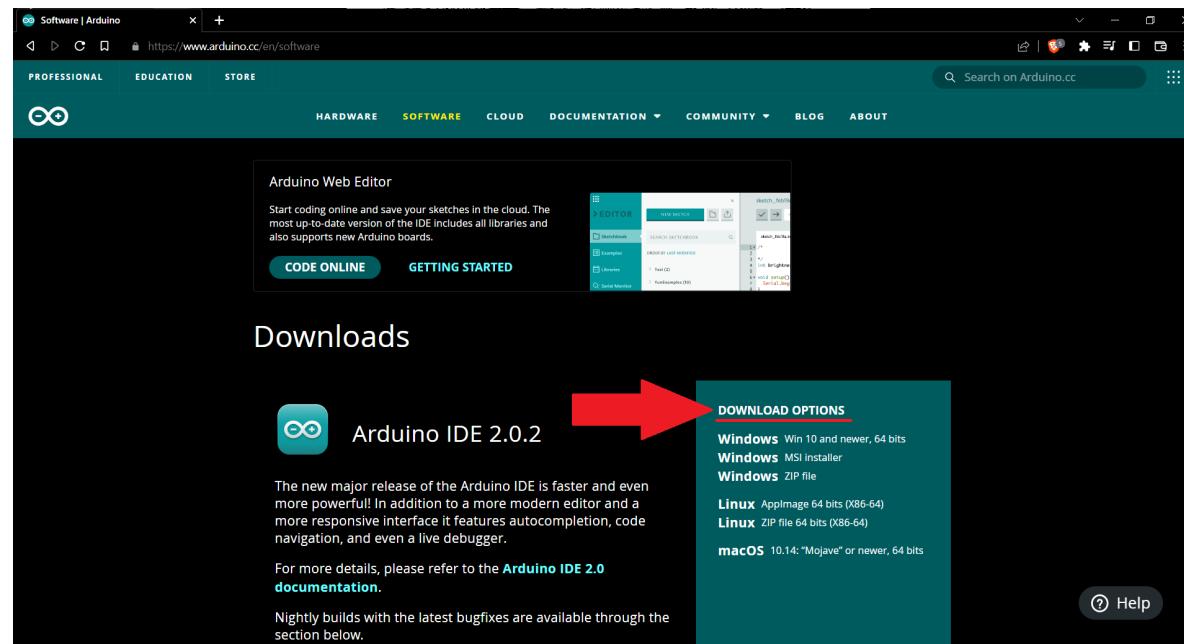
Os softwares citados abaixo podem ser instalados pelo seu navegador.

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

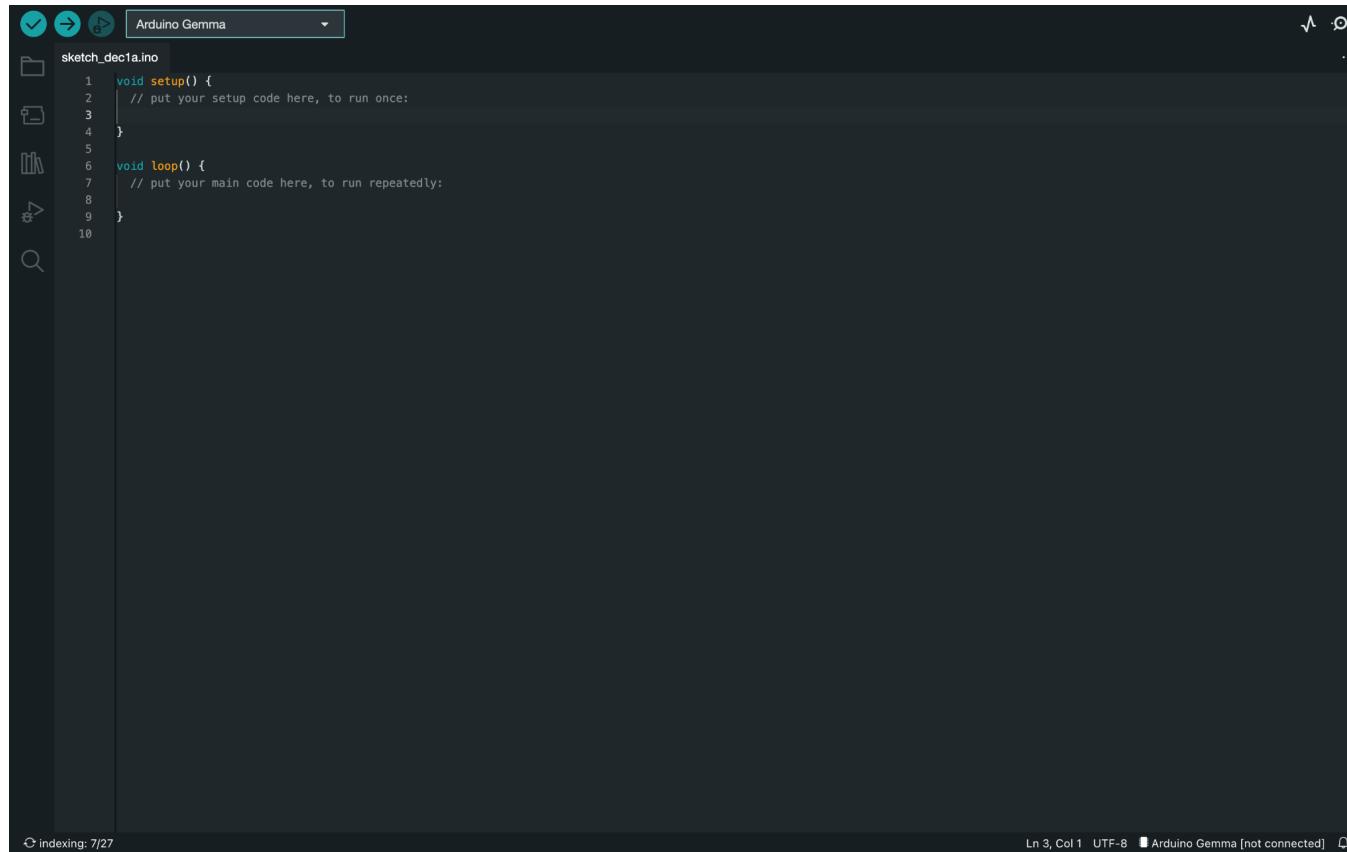


clique-o.

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:

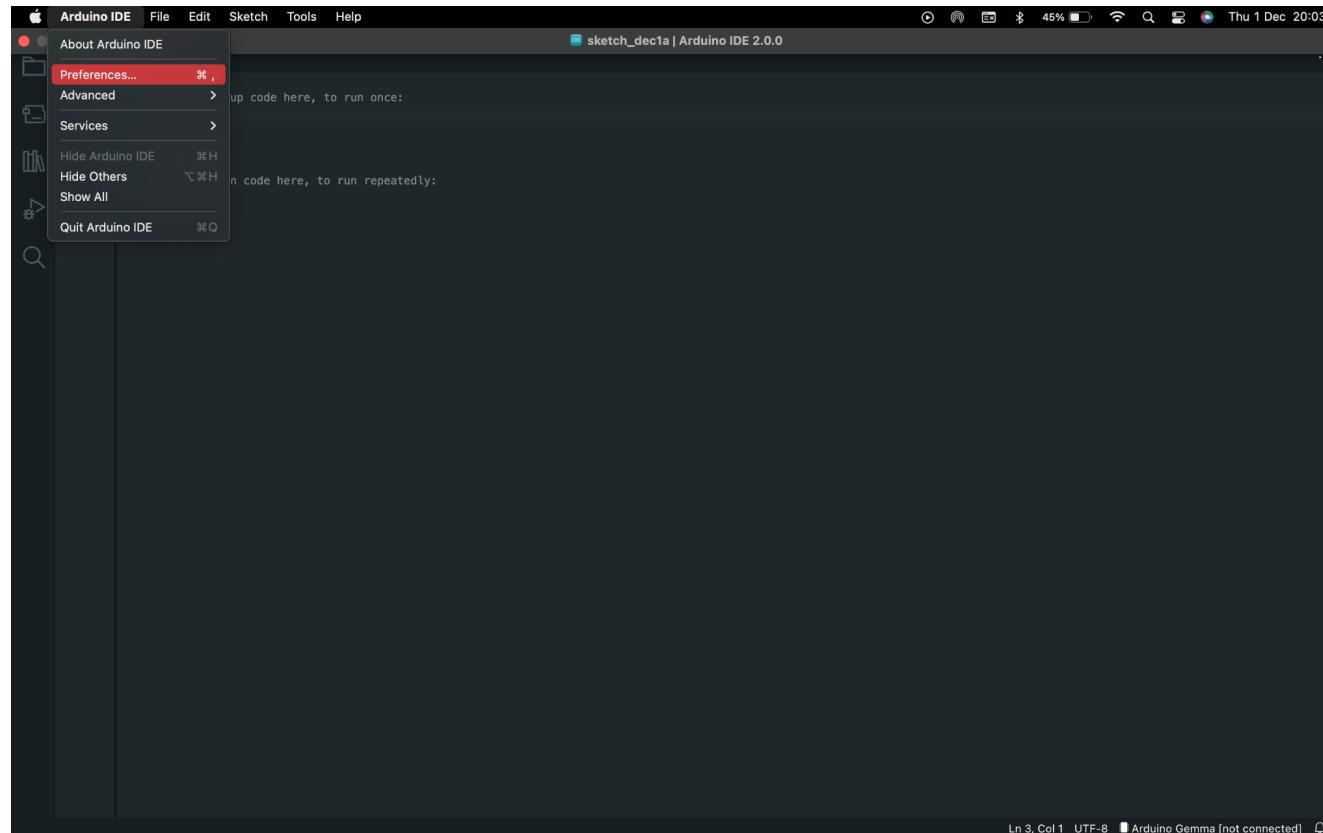


The screenshot shows the Arduino IDE interface. The title bar reads "Arduino Gemma". The central area displays a blank sketch named "sketch_dec1a.ino" with the following code:

```
1 void setup() {  
2     // put your setup code here, to run once:  
3  
4 }  
  
5 void loop() {  
6     // put your main code here, to run repeatedly:  
7  
8 }  
9  
10
```

The left sidebar contains icons for file operations (New, Open, Save, etc.) and a search function. At the bottom of the interface, there is status information: "indexing: 7/27", "Ln 3, Col 1 UTF-8", "Arduino Gemma [not connected]", and a refresh icon.

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

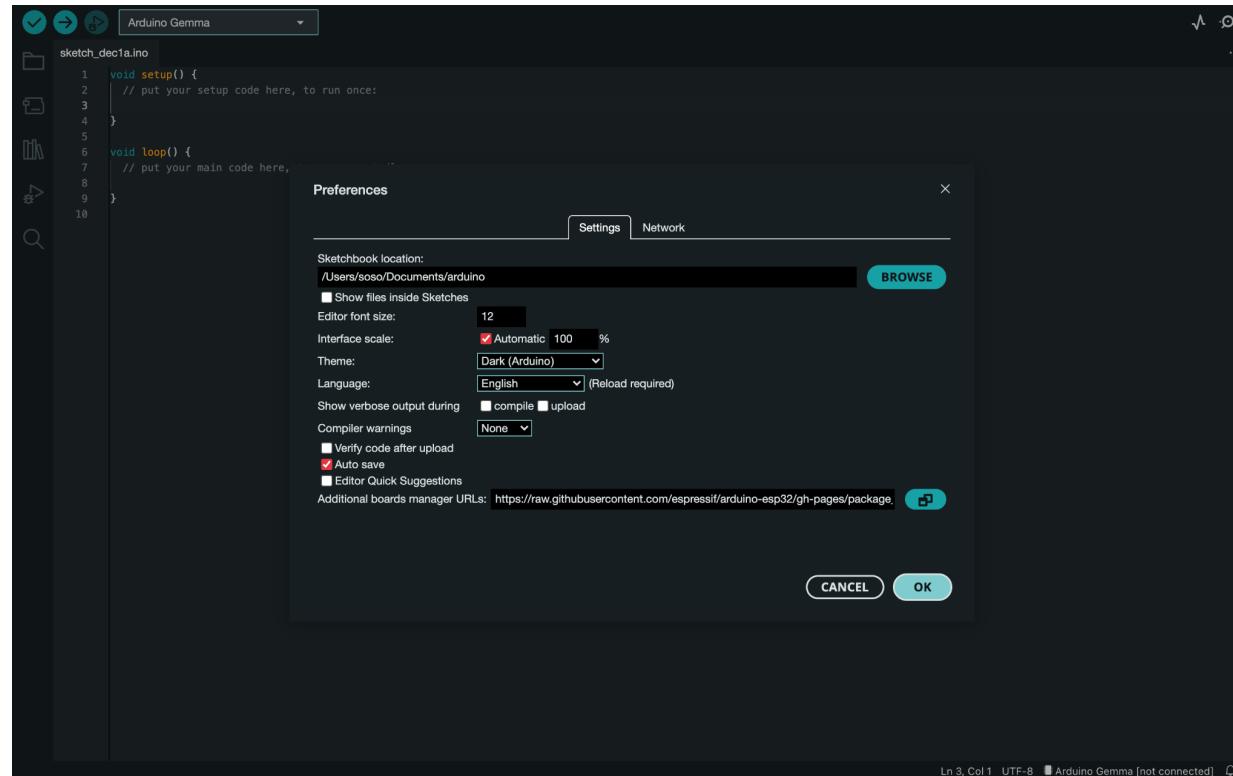


Após entrar nas Preferências, você encontrará essa página

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".

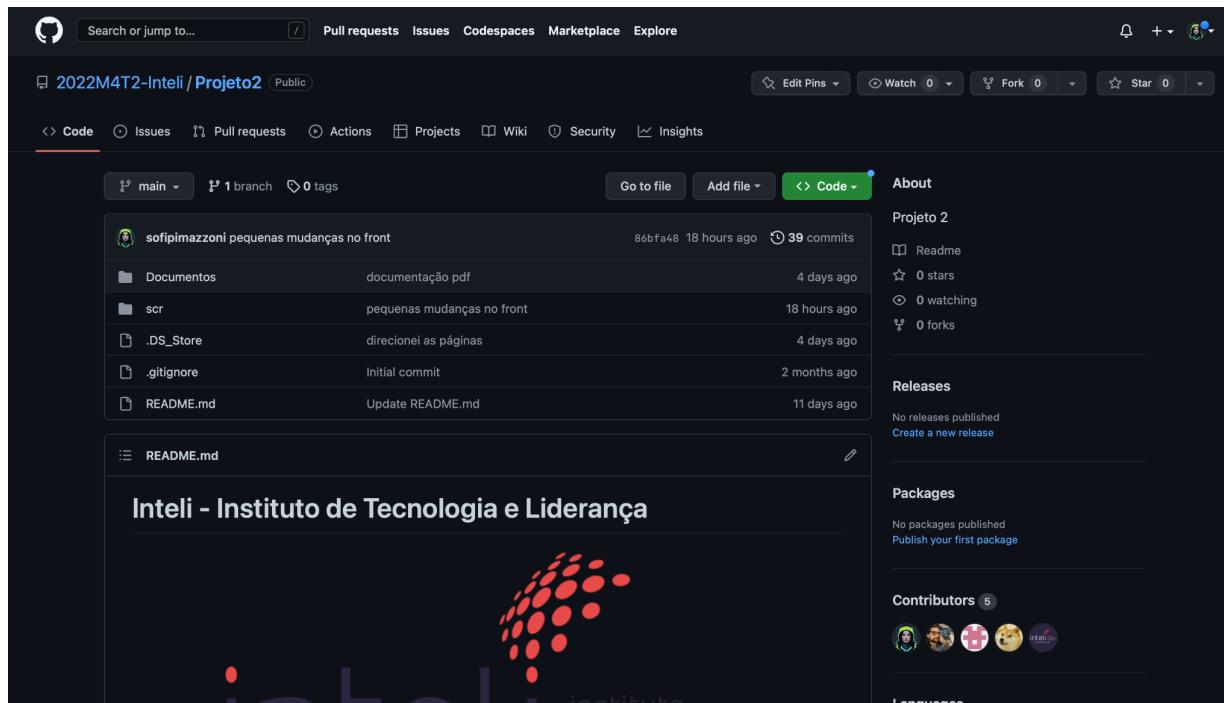


Feito isso, basta aguardar o aplicativo acabar a instalação. Ela pode levar de 20 a 40 minutos.

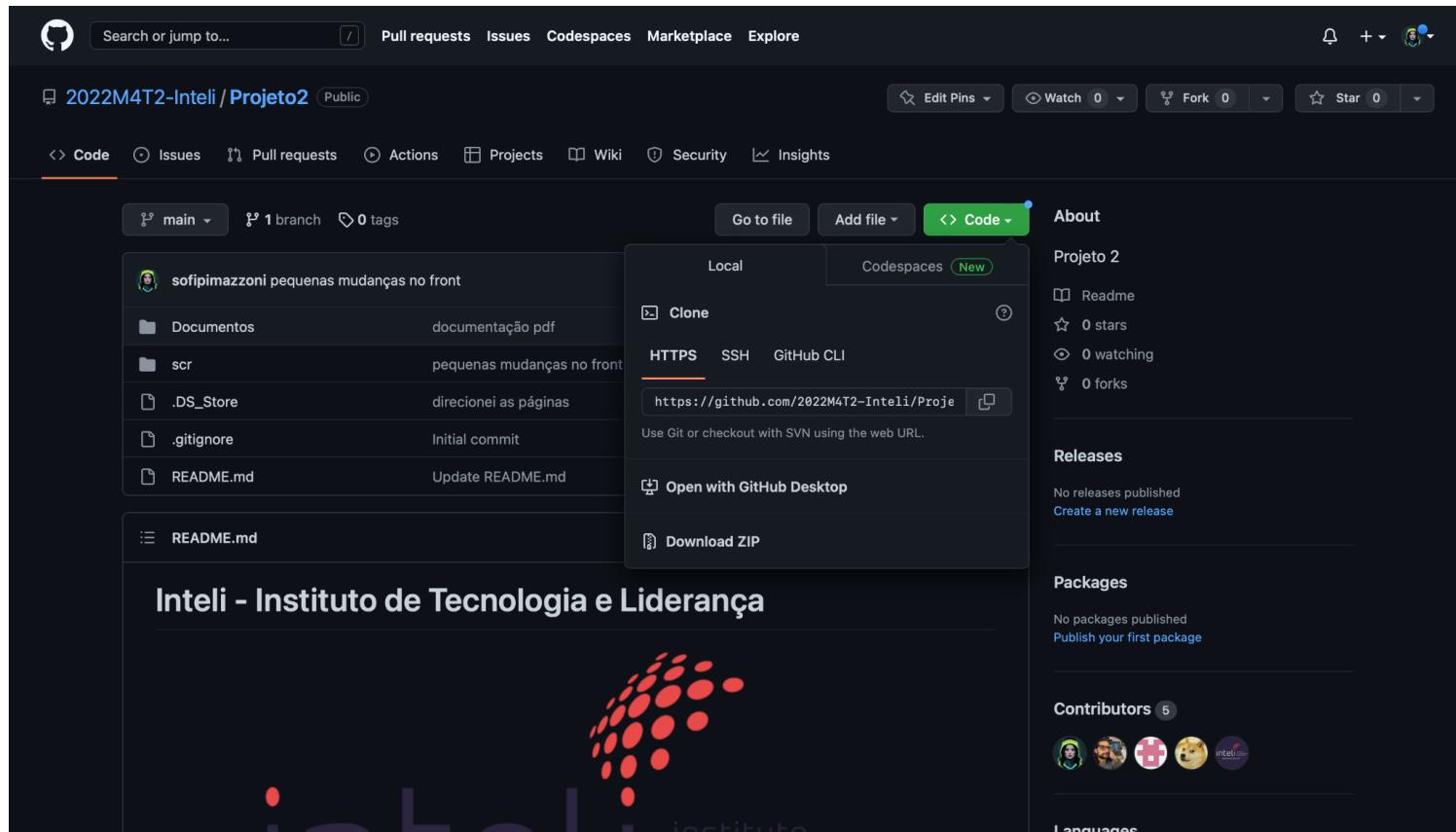
Com isso, você já possui o aplicativo para visualizar e modificar o código em C++, mas para isso, é necessário o código em si. Para baixá-lo, siga os passos abaixo:

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

Clique no botão verde escrito "code".

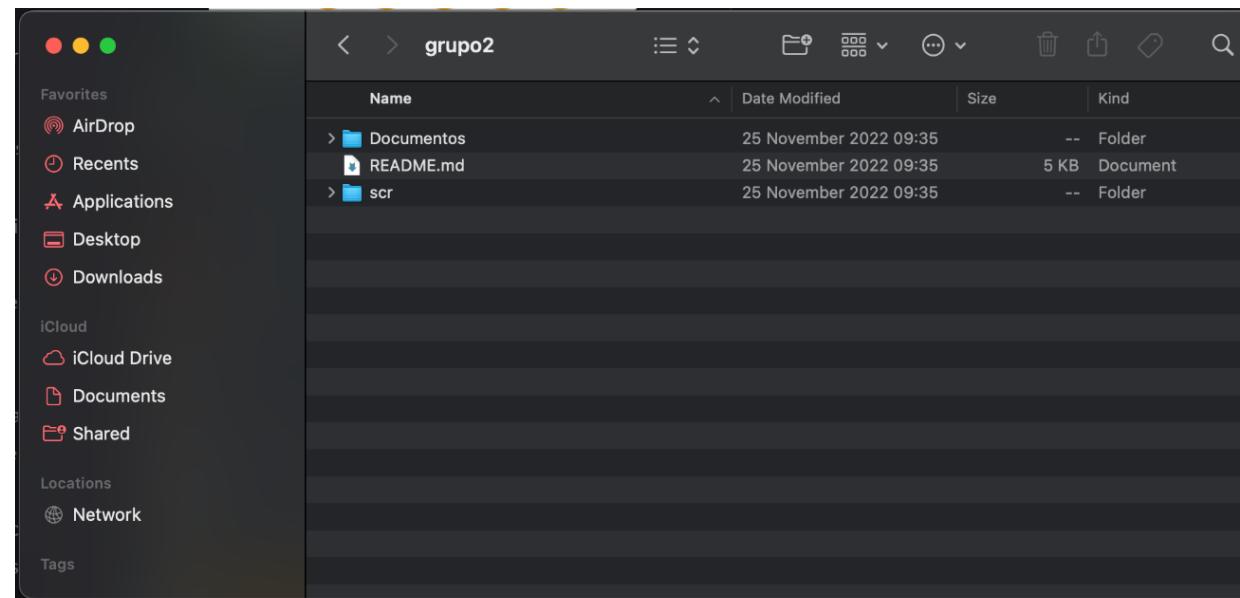


Após clicar, vão aparecer estas opções, e para adquirir os arquivos, clique em "Download ZIP".

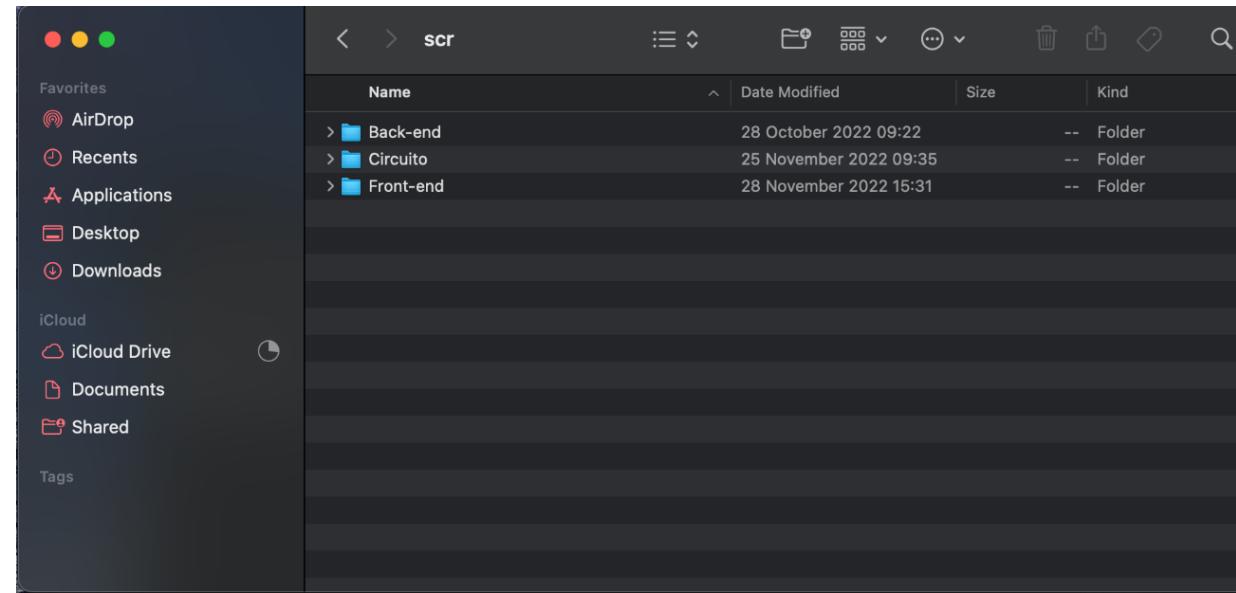


Com o ZIP instalado, basta descomprimir para que se torne uma pasta normal.

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



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.



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".

6. Guia de Operação

(sprint 5)

Descreva os fluxos de operação entre interface e dispositivos IoT. Indique o funcionamento das telas, como fazer leituras dos dados dos sensores, como disparar ações através dos atuadores, como reconhecer estados do sistema.

Indique também informações relacionadas à imprecisão das eventuais localizações, e como o usuário deve contornar tais situações.

Utilize fotografias, prints de tela e/ou desenhos técnicos para ilustrar os processos de operação.

7. Troubleshooting

(sprint 5)

Liste as situações de falha mais comuns da sua solução (tais como falta de conectividade, falta de bateria, componente inoperante etc.) e indique ações para solução desses problemas.

#	Problema	Possível solução
1		
2		
3		
4		
5		

8. Créditos

(sprint 5)

Seção livre para você atribuir créditos à sua equipe e respectivas responsabilidades