



# Manual de Instruções

BeMore  
IPT

## Controle do Documento

### Histórico de revisões

Data	Autor	Versão	Resumo da atividade
08/12/2023	Grupo 4 - BeMore  Última atualização : Thiago Volcati	Versão 1 - Sprint 4	Criação do Documento Inicial
17/12/2024	Grupo 4 - BeMore  Última atualização: Luiza Faria Petenazzi	Versão 2 - Sprint 5	Revisão do documento

# Índice

1. Componentes e Recursos	3
1.1. Componentes externos	3
1.2. Requisitos de conectividade	5
2. Guia de Montagem	7
3. Guia de Instalação	15
4. Guia de Configuração	21
5. Guia de Operação	25
6. Como descartar este dispositivo	29
7. Troubleshooting	32

# 1. Componentes e Recursos

Nesta seção, iremos listar os componentes, tanto de hardware quanto de software, que fazem parte da solução proposta. Esses componentes são essenciais para garantir o funcionamento da solução. A seguir, detalharemos todos os itens que integram a solução, desde os dispositivos físicos até as tecnologias utilizadas para desenvolvimento e comunicação.

## 1.1. Componentes externos

### Hardware

1. **ESP32**: O microcontrolador de 32 bits com conectividade Wi-Fi e Bluetooth é o componente central do sistema. Ele coleta dados dos sensores, processa as informações e as transmite via MQTT para o broker HiveMQ. Esse dispositivo também se conecta à rede Wi-Fi, possibilitando a comunicação entre os sensores e a aplicação backend.
2. **BME280**: Este sensor mede pressão, temperatura e umidade, e é usado para monitorar as condições ambientais ao redor dos equipamentos industriais. Ele fornece dados sobre a temperatura e pressão nos locais de operação dos equipamentos.
3. **MMA845X**: Acelerômetro de 3 eixos que mede aceleração e vibração, detectando padrões anormais que possam indicar falhas ou desgaste dos equipamentos. Sua sensibilidade ajuda na detecção precoce de falhas mecânicas.
4. **HC-SR04**: Sensor ultrassônico utilizado para medir distâncias com alta precisão. Ele monitora o deslocamento e tempo de atividade da prensa, ajudando a verificar seu tempo de funcionamento e movimentos durante a operação.
5. **Display LCD 16x2**: Tela de exibição que apresenta dados em tempo real, como temperatura e vibração, diretamente no dispositivo. Este display oferece um feedback visual imediato sobre o status do sistema.
6. **LED RGB**: O LED de múltiplas cores é usado para indicar o estado do sistema, como alertas, falhas ou operação normal, oferecendo feedback visual para os usuários.
7. **Botão**: Um botão físico que permite a interação manual com o sistema, podendo ativar ou desativar os sensores conforme necessário. Esse componente oferece controle direto no dispositivo.
8. **Jumpers (fios)**: Fios de conexão usados para interligar todos os componentes eletrônicos, como o ESP32, sensores, display e atuadores. Eles asseguram a comunicação e o fluxo de dados entre os elementos do sistema.

9. **Antena Wi-Fi:** Essencial para garantir uma conexão Wi-Fi estável e de longo alcance, permitindo que o ESP32 se conecte de forma confiável à rede, facilitando a comunicação com o broker MQTT.

10. **Resistores:** Componente eletrônico utilizado para controlar a corrente elétrica nos circuitos, protegendo os sensores e o microcontrolador, evitando sobrecargas.

## Software

1. **Arduino IDE:** Utilizado para programar o microcontrolador ESP32. Através da IDE, é possível implementar a lógica de funcionamento dos sensores, o envio dos dados via MQTT e outras funcionalidades do sistema.
2. **VS Code:** Editor de código utilizado para o desenvolvimento da aplicação backend e frontend. Oferece suporte para linguagens como C# e JavaScript, além de facilitar a integração com outras ferramentas de desenvolvimento.
3. **C/C++:** Linguagens de programação utilizadas para o desenvolvimento do firmware do ESP32, permitindo a implementação das funções de leitura de sensores, comunicação via MQTT e controle dos atuadores.
4. **C#:** Linguagem de programação usada no desenvolvimento do backend. É responsável pelo processamento dos dados

recebidos do broker MQTT e pela comunicação com o banco de dados.

5. **.NET:** Framework utilizado para o desenvolvimento do backend, que consome os dados via MQTT, processa as informações e interage com o banco de dados PostgreSQL.
6. **HiveMQ:** Broker MQTT utilizado para gerenciar a troca de mensagens entre os dispositivos IoT e o backend. Ele permite a comunicação em tempo real, garantindo a entrega confiável dos dados.
7. **Next.js:** Framework de desenvolvimento para a construção do frontend da aplicação web, que exibe os dados em tempo real nos dashboards interativos, proporcionando uma interface de usuário responsiva e intuitiva.
8. **Recharts:** Biblioteca gráfica utilizada para gerar gráficos interativos no frontend, permitindo a visualização clara dos dados dos sensores e facilitando a análise de performance dos equipamentos.
9. **TailwindCSS:** Framework CSS utilizado para a estilização do frontend. Com suas classes utilitárias, ele permite a criação de interfaces consistentes e adaptáveis, agilizando o desenvolvimento da aplicação.
10. **Prisma ORM:** Ferramenta de mapeamento objeto-relacional utilizada para facilitar a interação entre o backend .NET e o banco de dados PostgreSQL. O Prisma ORM simplifica consultas e manipulação dos dados armazenados.

## 1.2. Requisitos de conectividade

### Redes e Protocolos de Rede

1. **Wi-Fi:** A comunicação entre os dispositivos IoT e a plataforma de backend será realizada através de uma rede Wi-Fi dedicada, garantindo uma conexão estável e rápida para o envio contínuo de dados dos sensores. O ESP32 será o dispositivo responsável por se conectar a essa rede e enviar dados para o broker MQTT.
2. **MQTT (Message Queuing Telemetry Transport):** O protocolo de comunicação MQTT será utilizado para a troca de mensagens entre os dispositivos IoT e o backend. O MQTT é um protocolo leve e eficiente, ideal para dispositivos IoT, permitindo a troca de dados em tempo real com baixa latência. O broker MQTT (HiveMQ) gerenciará a comunicação, garantindo a entrega das mensagens e a confiabilidade dos dados.
3. **HTTP/HTTPS:** A comunicação entre o frontend da aplicação web (desenvolvido com Next.js) e o backend será feita por meio de requisições HTTP ou HTTPS. O uso de HTTPS assegura a segurança da troca de dados, criptografando as informações trocadas entre o servidor e o cliente.

### Especificações de Back-End

1. **Broker MQTT - HiveMQ:** O HiveMQ atuará como o broker central para a troca de mensagens. Ele será responsável por gerenciar a publicação e a assinatura dos tópicos MQTT, permitindo que os dispositivos IoT enviem dados para o backend de forma eficiente. A arquitetura baseada em MQTT também facilita a escalabilidade do sistema, permitindo adicionar mais dispositivos IoT no futuro sem comprometer a performance.
2. **Backend - .NET Microserviço:** O backend será responsável por consumir os dados do broker MQTT e processá-los. Ele também se comunica com o banco de dados PostgreSQL para armazenar e consultar os dados históricos. O uso de um microserviço .NET permite uma arquitetura escalável e modular, que pode ser facilmente expandida para incluir novos módulos ou funcionalidades.
3. **Banco de Dados - PostgreSQL:** O banco de dados PostgreSQL será utilizado para armazenar os dados históricos coletados pelos sensores. Ele garantirá a integridade e a consistência dos dados, além de oferecer consultas rápidas para os painéis de controle (dashboards) e a geração de relatórios.
4. **API RESTful:** A comunicação entre o backend e o frontend será feita por meio de APIs RESTful, que fornecerão os dados necessários para a visualização nos dashboards e permitirão interações entre o usuário e o sistema, como a exibição de gráficos e a configuração de alertas.

### Infraestrutura e Servidores

1. **Servidor Web:** A aplicação web será hospedada em servidores que garantirão alta disponibilidade e baixo tempo de resposta para os usuários. Esse servidor será responsável por renderizar o frontend com o Next.js e atender às requisições HTTP/HTTPS dos usuários.

Em conclusão, a conectividade da solução proposta foi projetada para garantir a comunicação eficiente e segura entre os dispositivos IoT, o backend e a aplicação web. O uso de Wi-Fi e MQTT oferece uma solução flexível e de baixo custo para transmissão de dados em tempo real. A integração entre o broker MQTT, o backend em .NET, o banco de dados PostgreSQL e o frontend desenvolvido com Next.js proporciona uma arquitetura robusta, escalável e capaz de atender a diferentes necessidades de monitoramento e manutenção de equipamentos industriais. O sistema foi pensado para suportar expansões futuras, permitindo a adição de novos dispositivos IoT e funcionalidades à medida que as necessidades do laboratório metalúrgico evoluam.

## 2. Guia de Montagem

### 2.1 Equipamentos e ferramentas necessárias

1. Chave de fenda
2. Cortadora laser
3. Acrílico
4. Parafuso

### 2.2 Componentes necessários

Quadro 1: Componentes necessários

Nome	Quantidade
Esp32 Tipo - C + Cabo	1
mpu6050 6-Axis Accel & Gyro Sensor	1
Cabo jumper	10

macho-macho	
Resistor	10
Botão momentâneo 4 pinos	4
DHT22	1
Micro Servo 9g	1
GY-BME280	2
Módulo Sensor de Distância Ultrassônico	1
Módulo Relé 1 Canal 5V	1



	1
Buzzer 12mm 5V Ativo Contínuo	
Potenciômetro Linear B1K L15	1
Capacitor Eletrolítico 10uF (10mF) 50V 105°C	1
Resistor Dependente de Luz 5mm	1
Display LCD 16X2 - BackLight Azul - I2C	1

Placa Arduino Uno R3 + Cabo USB	1
Led RGB 5mm	1
Sensor Fotoelétrico Óptico 0-100cm Difuso E3F-DS100	1
Multímetro Digital	1
Sensor de Corrente 30A	2
Módulo Acelerômetro 3 eixos	1

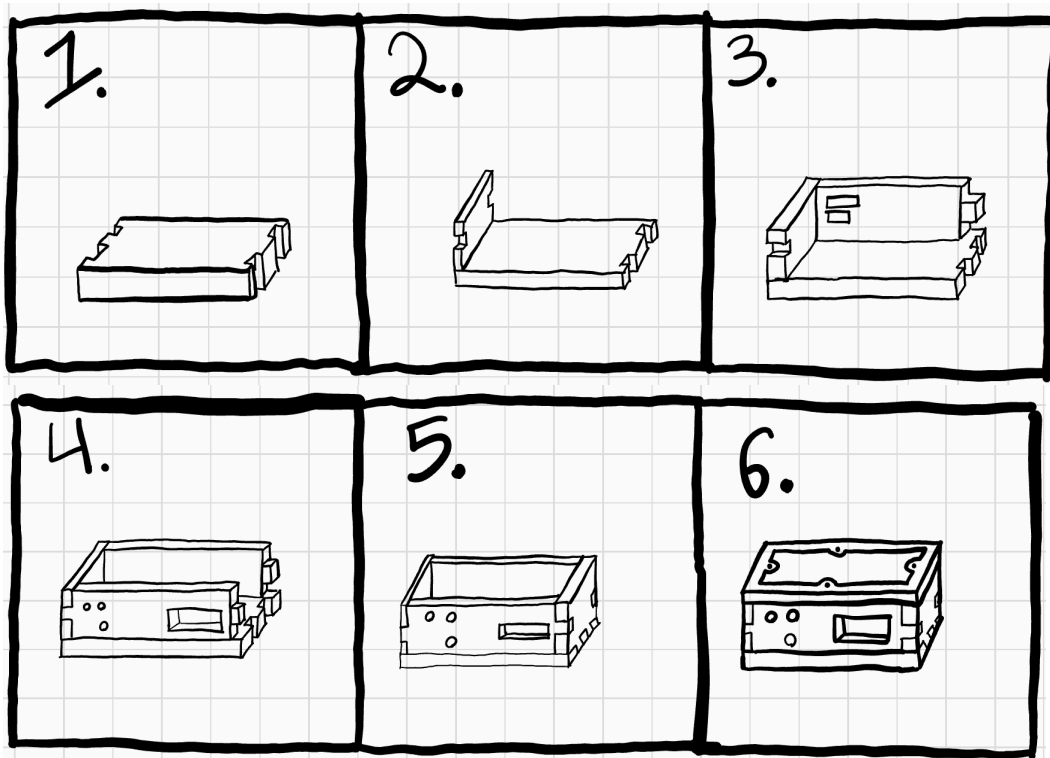
## 2.3 Passo a passo

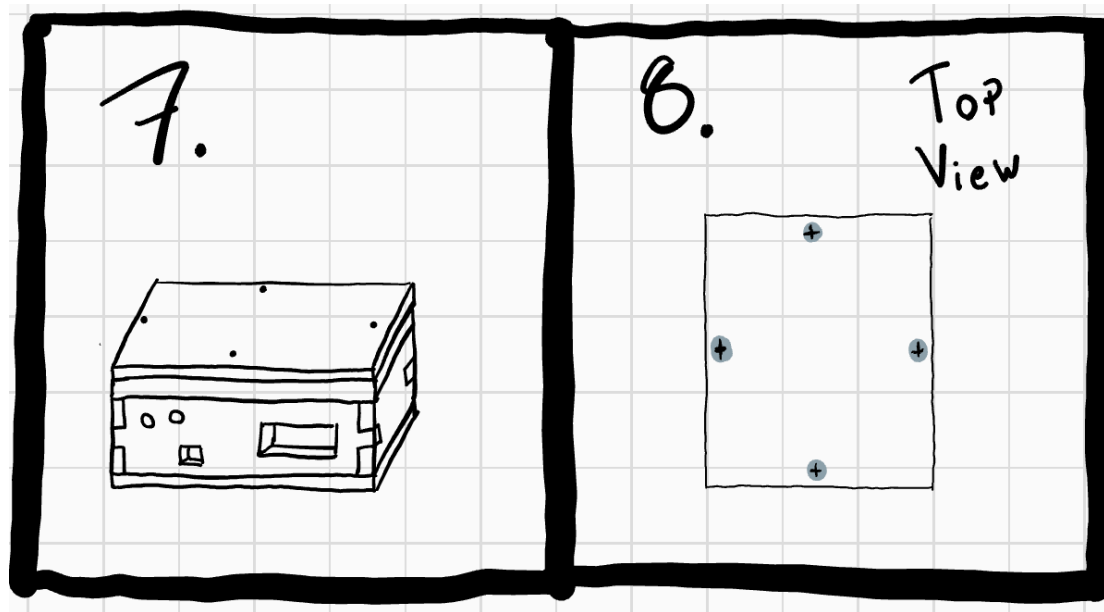
### 2.3.1 Montando a caixa

2.3.1.1 Baixe o arquivo CasePrensa.dxf

2.3.1.2 Faça o corte dele na corta a laser

2.3.1.3 Siga os paços abaixo para montar a caixa do hardware





## 2.3.2 Montando a Placa

### 2.3.2.1 Lista de Pinos

1. Led RGB(Vermelho): Pino 33
2. Led RGB(Verde): Pino 25
3. Led RGB(Azul): Pino 26
4. Led RGB(Terra): Pino GND
5. Display LCD(VCC): Pino 5V
6. Display LCD(GND): Pino GND
7. Display LCD(SCL): Pino 22
8. Display LCD(SDA): Pino 21
9. Botão: Pino 19
10. Botão(Terra): Pino GND
11. Interruptor: Pino 15
12. Interruptor(Terra): Pino GND
13. Led Vermelho: Pino 32
14. Led Vermelho(Terra): Pino GND
15. HC-SR04(Trig): Pino 4
16. HC-SR04(Echo): Pino 5
17. HC-SR04(Terra): Pino GND
18. HC-SR04(VCC): Pino 5V
19. Buzzer: Pino 18
20. Buzzer(Terra): Pino GND

### 2.3.2.2 Caminhos da Placa

1. O ESP32 deve ser conectado à placa de maneira com que sua entrada para fonte fique para fora da área da placa.
2. Os pinos listados acima devem todos serem conectados a pinos e soldados à placa.
3. Além dos pinos, é necessário conectar a Antena do ESP32

### 2.3.3 Juntando a Placa com a Caixa

\* Tome a imagem acima como referência de direção para a montagem da placa.

1. Cole a placa com o lado da solda virado para baixo, no chão dentro da caixa usando fita adesiva dupla face ou cola quente. Cole ela com a entrada do **ESP32** apontando para a parede menor da esquerda e o outro lado encostando na parede menor da direita.
2. Na parede da frente da caixa, encaixe e parafuse o **Display LCD** no buraco grande (confira se o sentido do display está correto), coloque o **Led RGB** e o **Led Vermelho** nos dois buracos que estão um do lado do outro e coloque a Antena no último buraco, desrosqueando o fio dela e passando de dentro para fora da caixa.
3. Desconecte cada um dos fios do Sensor HC-SR04 e os passe pelo buraco de cima da parede atrás da caixa. **Muito cuidado para não os confundir na hora de reconectar, os marque se possível!.**

4. Conecte a tomada do cabo fonte do **ESP32** em alguma tomada próxima da máquina. Passe o fio pelo buraco de baixo da parede de trás caixinha.
5. Conecte o cabo fonte do ESP32 nele e aguarde o sistema iniciar.

## 3. Guia de Instalação

Este guia apresenta um passo a passo detalhado para a instalação dos dispositivos IoT **Compressor** e **Prensa** no ambiente físico destinado, conectando-os à rede e configurando-os para o funcionamento pleno. O documento abrange tanto a instalação física dos dispositivos quanto a configuração dos softwares necessários. O objetivo é assegurar que qualquer pessoa com conhecimentos básicos possa reproduzir o ambiente de operação.

Antes de iniciar, certifique-se de que possui os seguintes itens:

- Guia de montagem foi realizado;
- Internet local acessível;
- Há tomada disponível;
- Computador/notebook disponível.

Após toda a montagem física, é necessário instalar o código-fonte da solução e carregá-lo no microcontrolador Esp32.

### 3.1 Instalação do Arduino IDE

Uma IDE (Integrated Development Environment) é uma ferramenta essencial para o desenvolvimento de projetos de dispositivos IoT, como o Compressor e a Prensa, pois integra um conjunto de recursos indispensáveis em um único ambiente. Com uma IDE, é possível escrever, compilar, depurar e carregar códigos diretamente nos microcontroladores, como o ESP32, facilitando o desenvolvimento e a implementação de funcionalidades. No caso do Arduino IDE, ele oferece uma interface amigável, suporte a diversas placas e bibliotecas específicas, além de simplificar a comunicação com os sensores e dispositivos conectados.

Portanto, devemos instalar o Arduino IDE.

#### 3.1.1 Download do Arduino IDE

- Acesse o site oficial do Arduino: [www.arduino.cc](http://www.arduino.cc).
- No menu superior, clique em "Software" e selecione "Downloads".
- Escolha a versão apropriada para o seu sistema operacional (Windows, macOS ou Linux).
- Clique em "Just Download" (ou contribua com uma doação, se preferir).

Figura x - Site oficial para download do Arduino IDE

7. Durante a instalação, será solicitado que você permita a instalação dos drivers do Arduino. Clique em **"Instalar"** quando aparecer o prompt.
8. Após finalizar, clique em **"Finish"**.

Obs: para outros sistemas operacionais é importante verificar como fazer a instalação de um arquivo executável.

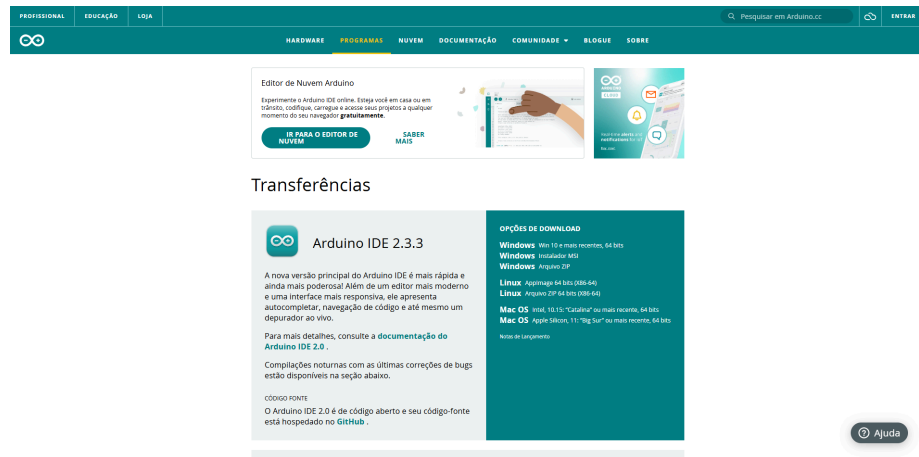
### 3.1.3 Instalação do ESP32

Para configurar o ESP32 no Arduino IDE, é necessário realizar algumas etapas para que a placa seja reconhecida e programada corretamente. Inicialmente, ao abrir o Arduino IDE, você deve acessar as preferências do software e adicionar o link do Gerenciador de Placas ESP32. Esse link permite que o IDE tenha acesso ao pacote necessário para suportar o ESP32. O link a ser adicionado é:

[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json).

Após adicionar o link, é necessário instalar o pacote do ESP32. Isso é feito por meio do Gerenciador de Placas, acessado pelo menu de ferramentas no Arduino IDE. No gerenciador, você deve buscar pelo nome "ESP32 by Espressif Systems" e proceder com a instalação.

Figura x - Imagem de "preferências" para instalar o Esp32



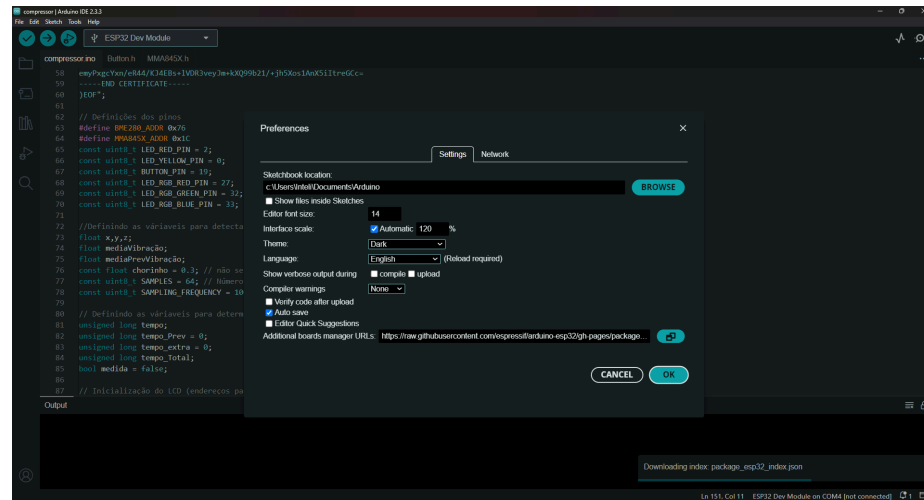
### 3.1.2 Instalação no Windows

1. Localize o arquivo baixado (ex.: `arduino-ide- <versão> -windows.exe`) e clique duas vezes para executar.
2. Na tela inicial, clique em **"Next"**.
3. Aceite os termos da licença e clique em **"Next"**.
4. Escolha os componentes a instalar. Certifique-se de que as opções para drivers e ferramentas estão marcadas.
5. Escolha o diretório de instalação ou mantenha o padrão.
6. Clique em **"Install"** e aguarde a conclusão do processo.

- **Wire.h** para comunicação I2C (LCD e sensores).
- **PubSubClient.h** para comunicação MQTT.
- **WiFi.h** para conexão à rede Wi-Fi.
- **Adafruit\_Sensor.h**
- **Adafruit\_BME280.h**
- **arduinoFFT.h**
- **"MMA845X.h"**

Com as bibliotecas corretamente instaladas no Arduino IDE, abra o arquivo do código-fonte correspondente ao protótipo (Compressor ou Prensa). Código disponível no repositório do github (<https://github.com/Inteli-College/2024-2B-T12-IN04-G04>).

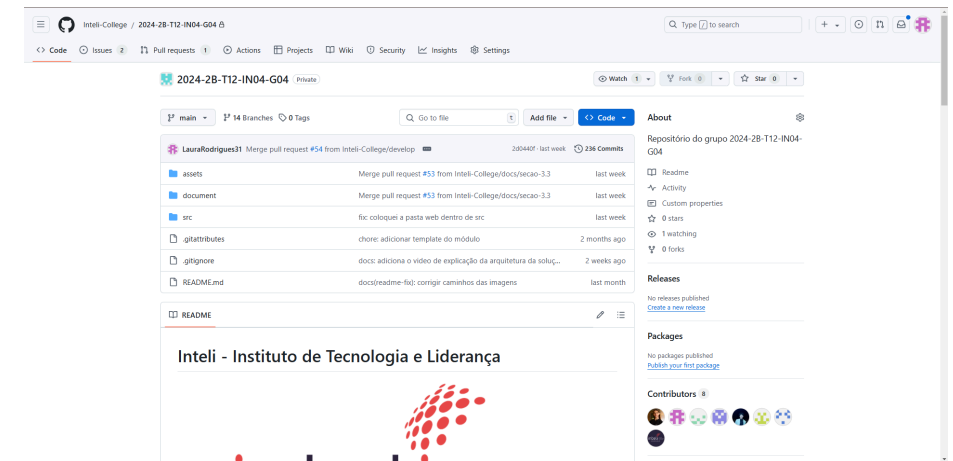
Figura x - Repositório do github



## 3.2 Instalação do Código-Fonte

Após configurar o Arduino IDE para o ESP32, é necessário instalar o código-fonte desenvolvido para o funcionamento dos protótipos Compressor e Prensa. O processo inclui a preparação do código, sua transferência para os dispositivos e a verificação do funcionamento.

Primeiramente, certifique-se de que o código-fonte está completo, incluindo as bibliotecas necessárias para o funcionamento dos sensores, comunicação MQTT e interface LCD. As bibliotecas principais que devem ser incluídas são:

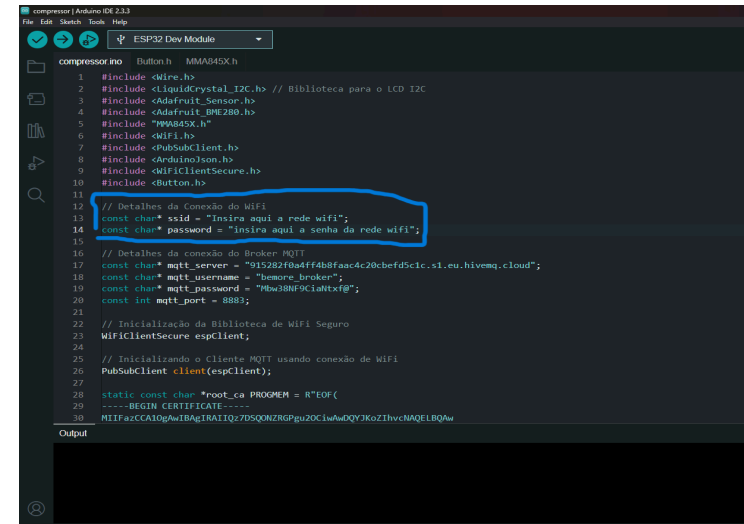




Clicando em "code" e em "Download zip" será baixado a pasta zip com todos os arquivos de códigos necessários para rodar os dois protótipos. Após o download da pasta zip, descompacte-a em um local de fácil acesso no seu computador, como a área de trabalho ou a pasta de documentos. Abra o Arduino IDE e, no menu superior, clique em **"File" (Arquivo)**, depois em **"Open" (Abrir)**. Navegue até a pasta descompactada e selecione o arquivo principal do código-fonte, que geralmente possui a extensão **.ino**. Certifique-se de que o arquivo aberto no Arduino IDE corresponde ao protótipo que deseja configurar, seja o Compressor ou a Prensa.

Antes de carregar o código no ESP32, configure os parâmetros do projeto, como o SSID e a senha da rede Wi-Fi, bem como os tópicos MQTT utilizados para publicação e assinatura.

figura x - código destacando o local das credenciais de internet



```

1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h> // Biblioteca para o LCD I2C
3 #include <Adafruit_Sensor.h>
4 #include <Adafruit_BME280.h>
5 #include "MP0545X.h"
6 #include <WiFi.h>
7 #include <PubSubClient.h>
8 #include <ArduinoJson.h>
9 #include <WiFiClientSecure.h>
10 #include <Button.h>
11
12 // Detalhes da conexão do WiFi
13 const char* ssid = "insira aqui a rede wifi";
14 const char* password = "insira aqui a senha da rede wifi";
15
16 // Detalhes da conexão do Broker MQTT
17 const char* mqtt_server = "915282f0d4ff4b8faac4c20cbef5c1c.v1.eu.hivemq.cloud";
18 const char* mqtt_username = "bemore_broker";
19 const char* mqtt_password = "b6u38Nf9Ciahtxf@";
20 const int mqtt_port = 1883;
21
22 // Inicialização da Biblioteca de WiFi Seguro
23 WiFiClientSecure espClient;
24
25 // Inicializando o Cliente MQTT usando conexão de WiFi
26 PubSubClient client(espClient);
27
28 static const char *root_ca PROGMEM = R"EOF(
29 -----BEGIN CERTIFICATE-----
30 MIIFazCCA30gAwIBAgIRAT1IQz7D5QONZRGpGuz2C1uWwQYJKoZIhvcNAQELBQAw
  
```

Após verificar as configurações, conecte o ESP32 ao computador por meio de um cabo USB e selecione a porta serial correta no Arduino IDE. Clique na opção "Upload" para transferir o código ao ESP32. Durante esse processo, o Arduino IDE compilará o código e carregará no dispositivo. Se a transferência for bem-sucedida, uma mensagem de conclusão será exibida na tela.

Com o código carregado, o próximo passo é testar o protótipo. Para isso, abra o Monitor Serial no Arduino IDE, configure a velocidade de comunicação correta (geralmente 115200 bauds) e observe as mensagens enviadas pelo ESP32. Essas mensagens devem incluir dados dos sensores, status da conexão Wi-Fi e confirmação da comunicação com o broker MQTT.

Se o ESP32 estiver se conectando corretamente ao broker e publicando os dados nos tópicos configurados, o sistema estará pronto para operação no ambiente físico, integrando-se ao restante do projeto. Caso contrário, será necessário revisar as configurações de Wi-Fi, tópicos MQTT e possíveis erros no código.

Este procedimento garante que os protótipos Compressor e Prensa estejam completamente operacionais, permitindo coleta de dados em tempo real e interação remota por meio do broker HiveMQ.

### 3.3 Ambiente Físico

Com os protótipos montados e os códigos-fonte devidamente carregados na memória do ESP32, é possível proceder com a instalação dos dispositivos no ambiente físico adequado. Este processo foi projetado para atender às necessidades práticas do parceiro de negócios, garantindo que os protótipos estejam em posições estratégicas para coleta e monitoramento de dados em tempo real.

#### Prensa

O protótipo da prensa deve ser acoplado próximo ao equipamento para monitorar sua operação. O sensor de proximidade deve ser

posicionado em uma região estratégica da prensa, no braço acima dela, garantindo que seja capaz de identificar a movimentação feita. O display LCD deve ser posicionado de maneira acessível, permitindo a leitura clara de informações operacionais. O LED do protótipo também serve como indicativo visual e deve estar visível ao operador ou responsável pela máquina. Na figura abaixo o que está na área vermelha é onde precisa ficar o sensor de proximidade. A área verde é a que se movimenta e o sensor detecta quantos centímetros.



## 4. Guia de Configuração

### 4.1 Conexão com Broker MQTT

A integração eficaz entre dispositivos em um sistema de Internet das Coisas (IoT) depende, em grande parte, de uma arquitetura de comunicação sólida. Nesse cenário, o Broker MQTT (Message Queuing Telemetry Transport) desempenha um papel de grande importância, atuando como o intermediário que possibilita a troca assíncrona de mensagens entre dispositivos conectados. Nesta seção, vamos explorar o que é um Broker MQTT, sua relevância, as vantagens do HiveMQ como broker e como configurar o cliente MQTT.

#### 4.1.1 Broker MQTT

Um Broker MQTT é um elemento na arquitetura de IoT que facilita a comunicação entre dispositivos através de um modelo de publicação e subscrição de tópicos.

## Funcionamento da Publicação e Subscrição

O modelo de publicação e subscrição é central para o funcionamento do MQTT:

- Publicação: Neste processo, um dispositivo (o publicador) envia mensagens para um tópico específico no broker. Essas mensagens podem incluir dados como leituras de sensores ou informações sobre o estado do dispositivo.
- Subscrição: Dispositivos que desejam receber essas mensagens (os subscritores) se inscrevem em tópicos específicos. Quando uma nova mensagem é publicada em um tópico ao qual um dispositivo está inscrito, o broker se encarrega de entregar essa mensagem ao dispositivo subscritor.

Esse modelo permite que os dispositivos se comuniquem sem necessidade de conhecer diretamente uns aos outros, resultando em uma arquitetura desacoplada que favorece a escalabilidade e a manutenção do sistema.

### 4.1.2 HiveMQ como Broker

Para este projeto específico, estamos utilizando o HiveMQ como nosso broker MQTT. O HiveMQ é uma solução robusta e escalável

desenvolvida para atender às exigências de sistemas IoT.

Aqui estão algumas das principais vantagens do HiveMQ:

- Escalabilidade: O HiveMQ é capaz de suportar milhões de dispositivos conectados ao mesmo tempo, tornando-o ideal para aplicações em grande escala.
- Desempenho: O broker oferece alta performance com baixa latência na entrega de mensagens, fundamental para aplicações que exigem comunicação em tempo real.
- Segurança: O HiveMQ proporciona suporte a autenticação forte e criptografia TLS/SSL, assegurando que as mensagens trocadas sejam seguras e protegidas contra interceptações.
- Extensibilidade: O HiveMQ permite a integração com outras ferramentas e serviços através de plugins, possibilitando personalizações conforme as necessidades específicas do projeto.

### 4.1.3. Conexão com o client MQTT com a dashboard.

A primeira etapa para a integração dos dados recebidos no broker com uma dashboard, foi elaborar um micro serviço capaz de transformar os dados recebidos num determinado tópico e criar INSERTS em um banco de dados com eles.

A configuração do cliente MQTT no microserviço é uma etapa central para a integração eficiente entre o sistema e o Broker MQTT, permitindo o processamento e armazenamento das mensagens enviadas pelos dispositivos conectados. O objetivo é garantir que os dados de operação das máquinas sejam recebidos, interpretados e armazenados em tempo real no banco de dados.

### 4.1.3.1. Definição da Conexão com o Broker MQTT

O primeiro passo na configuração do cliente é estabelecer uma conexão estável com o Broker MQTT, neste caso, o HiveMQ. A conexão é feita a partir do microserviço, que funciona como um consumidor das mensagens publicadas no tópico “prensa”.

- O endereço do broker, bem como a porta e as credenciais de autenticação, são configurados para garantir que o microserviço tenha acesso ao canal de comunicação.
- Após configurada, a conexão permanece ativa, e o cliente está pronto para ouvir e processar mensagens publicadas.

### 4.1.3.2. Subscrição ao Tópico de Interesse

O microserviço é configurado para subscrever-se ao tópico chamado “prensa”. Este tópico contém informações específicas sobre o funcionamento das máquinas do tipo prensa.

- Por que este tópico é importante?  
Ele centraliza as mensagens relevantes para o sistema, que incluem leituras como distância percorrida, qualidade do óleo, tipo de máquina, entre outras.
- Quando subscrito, o cliente automaticamente recebe as mensagens publicadas nesse tópico. Isso permite que o microserviço atue de forma proativa no processamento dos dados.

### 4.1.3.3. Processamento das Mensagens Recebidas

As mensagens recebidas pelo tópico são enviadas em formato JSON, um padrão leve e de fácil interpretação. Cada mensagem contém informações detalhadas sobre o estado e a operação de uma prensa.

- A estrutura dos dados recebidos inclui campos como ID da prensa, distância percorrida, qualidade do óleo, tempo de operação, entre outros.

- Assim que uma mensagem é recebida, o microserviço interpreta os dados e os transforma em um formato compatível com a estrutura do banco de dados.

#### 4.1.3.4. Armazenamento no Banco de Dados

Após processar as mensagens, o microserviço realiza a inserção desses dados na tabela prensaRunnings, que é utilizada para registrar informações em série temporal sobre as máquinas.

- A tabela é organizada para registrar o histórico completo de funcionamento, incluindo dados como o identificador da prensa, a distância percorrida e o horário do registro.
- Isso garante que todas as mensagens sejam registradas e possam ser utilizadas para análises futuras ou visualizações em dashboards.

Na imagem abaixo, é possível observar a estrutura da tabela prensaRunnings, que reflete os dados recebidos e armazenados

PrensaRunnings Insira uma expressão SQL para filtrar os resultados (use Ctrl+Espaço)

	RunningId	Prensald	DistanceTraveled	TimeStamp
1	2	7	2,992	2024-12-05 20:37:12.045 -0300
2	3	7	2,992	2024-12-06 00:05:52.597 -0300
3	4	7	4	2024-12-06 00:06:23.357 -0300
4	5	7	6	2024-12-06 00:06:54.754 -0300
5	6	7	8	2024-12-06 00:08:26.532 -0300
6	7	7	10	2024-12-06 00:13:24.929 -0300
7	8	7	12	2024-12-06 00:14:25.213 -0300
8	9	7	4	2024-12-06 00:23:31.034 -0300
9	10	7	6	2024-12-06 00:24:22.584 -0300
10	11	7	2,992	2024-12-06 08:31:58.249 -0300
11	12	7	8	2024-12-06 09:22:46.275 -0300
12	13	7	8	2024-12-06 09:22:54.467 -0300
13	14	7	10	2024-12-06 09:23:39.471 -0300
14	15	7	0	2024-12-06 10:00:12.197 -0300
15	16	7	0	2024-12-06 10:00:12.837 -0300
16	17	7	0	2024-12-06 10:00:13.653 -0300
17	18	7	0	2024-12-06 10:00:13.754 -0300
18	19	7	0	2024-12-06 10:00:14.590 -0300
19	20	7	0	2024-12-06 10:00:15.772 -0300
20	21	7	0	2024-12-06 10:00:16.613 -0300
21	22	7	0	2024-12-06 10:00:17.596 -0300
22	23	7	0	2024-12-06 10:00:18.584 -0300
23	24	7	0	2024-12-06 10:00:19.562 -0300

#### 4.1.3.5. Manutenção da Conexão e Tratamento de Erros

Para garantir a robustez do sistema, o cliente MQTT foi configurado para lidar com desconexões de forma automática. Caso ocorra

algum problema de rede ou o Broker fique temporariamente indisponível, o microsserviço tenta se reconectar periodicamente.

- Além disso, logs de erros são gerados para monitorar falhas, permitindo intervenções rápidas e garantindo a continuidade do sistema.

#### 4.1.3.6. Integração com o Frontend

Para que os dados armazenados sejam visualizados em tempo real, o microsserviço também colabora com o frontend do sistema. O frontend utiliza APIs para acessar os dados históricos no banco, mas também cria seu próprio cliente MQTT para receber atualizações em tempo real do tópico `prensa`.

- Isso garante que os gráficos na interface do usuário sejam interativos e reflitam tanto os dados históricos quanto os valores mais recentes publicados pelo Broker.

#### 4.1.3.7. Validação do Processo

Antes da implantação definitiva, o processo foi validado com testes práticos:

- Foram simuladas publicações de mensagens no tópico `prensa`.

- A funcionalidade foi conferida tanto no banco de dados quanto na exibição em dashboards.

### 4.2 Inicialização do Dashboard

Instruções para Iniciar o Dashboard

#### Instale as dependências necessárias:

No terminal, execute os comandos abaixo na pasta raiz do projeto:

```
npm install
```

```
npm install recharts
```

```
npm install prisma
```

1. Isso garantirá que todas as bibliotecas e ferramentas necessárias estejam disponíveis para o funcionamento do projeto.
2. **Acesse o caminho do arquivo:**  
Navegue até o diretório do projeto localizado em `src/web`.

#### Inicie o servidor de desenvolvimento:

Execute o comando:

```
npm run dev
```

3. Isso inicializará o dashboard na porta local **3000**.

O projeto foi desenvolvido utilizando o **Next.js**. Para quaisquer dúvidas relacionadas à inicialização, configuração ou aprimoramento do projeto, recomenda-se consultar a [documentação oficial do Next.js](#) ou entrar em contato

## 5. Guia de Operação

Este guia foi desenvolvido para assegurar o uso eficiente e correto do sistema, detalhando os fluxos de interação entre a interface do usuário e os dispositivos IoT conectados. Inclui orientações para monitorar sensores, acionar atuadores, acompanhar estados do sistema e corrigir imprecisões nas localizações registradas.

### 5.1 Visão Geral do Sistema

O sistema integra sensores e atuadores a uma interface web, permitindo monitoramento e controle em tempo real. Entre suas funcionalidades principais estão o monitoramento de sensores, que fornece uma visualização em tempo real dos dados coletados para identificar padrões e comportamentos operacionais; o controle de atuadores, que possibilita disparar ações específicas pela interface; a exibição dos estados do sistema, que indica o status atual dos dispositivos (como normal, alerta ou erro); e a funcionalidade de

ajuste de localizações, para corrigir discrepâncias nos dados de localização.

### 5.2 Navegação pela Interface

A interface web foi projetada para ser simples e eficiente, permitindo que o usuário tenha acesso rápido e direto às informações mais importantes sobre os dispositivos IoT conectados. O foco principal é oferecer uma experiência que priorize a transparência e a facilidade de uso, especialmente considerando que o sistema está integrado em uma rede interna (intranet), eliminando a necessidade de autenticação.

#### Estrutura da Página Inicial

Na página inicial, os principais componentes da interface são:

- Lista de dispositivos conectados (lado esquerdo):  
Esta lista apresenta todos os dispositivos monitorados pelo sistema, identificados de maneira única.
  - Ao lado de cada dispositivo, ícones podem ser adicionados futuramente para representar estados operacionais (ativo, inativo ou em manutenção).
  - Atualmente, cada dispositivo é exibido com um simples número ou identificador.
- Painel de informações detalhadas (lado direito):



- Quando um dispositivo é selecionado, o painel exibe dados em tempo real e informações resumidas, organizados em três seções principais:
  1. Gráficos de funcionamento ao vivo: Representam o comportamento operacional do dispositivo.
  2. Indicadores de desempenho: Informações como falhas acumuladas no dia, distância máxima alcançada, e data da última manutenção.
  3. Configurações ajustáveis: Campos para personalizar limites operacionais, como a altura máxima ou o tempo de funcionamento permitido.

## Funcionalidades Essenciais

A navegação pela interface é intuitiva e centrada no clique de dispositivos para explorar detalhes operacionais:

1. Seleção de dispositivos: Clicando em qualquer item da lista, o painel à direita se atualiza automaticamente para exibir as informações correspondentes.
2. Exibição gráfica:
  - Os gráficos mostram dados como a variação de distância percorrida pelo tempo, permitindo identificar padrões operacionais ao longo do dia.

- Esses gráficos são gerados em tempo real, com os dados sendo atualizados iterativamente, mas se limitam ao dia atual.
- 3. Configuração de parâmetros:
  - O campo azul na parte inferior permite ajustar valores como altura máxima ou tempo limite de operação.
  - Essas alterações são aplicadas diretamente ao dispositivo e são salvas no banco de dados para consultas históricas.



## 5.3 Fluxos de Operação

Nesta seção, os fluxos de operação detalham como os usuários interagem com o sistema para executar tarefas específicas, como

monitoramento de sensores, ajuste de parâmetros e gerenciamento de alertas.

### 5.3.1 Leituras de Dados dos Sensores

#### Acessando a Tela de Monitoramento

1. Na interface inicial, identifique o dispositivo que deseja monitorar na lista à esquerda.
2. Clique no dispositivo para abrir o painel detalhado, onde os dados dos sensores conectados ao aparelho serão exibidos.
3. Os dados são organizados em:
  - Gráficos em tempo real: Exibem o comportamento do sensor ao longo do dia.
  - Indicadores importantes: Mostram valores como a distância máxima atingida, falhas acumuladas e data da última manutenção.

Os gráficos fornecem uma visão clara do desempenho do sensor em tempo real. Eles incluem:

- Linhas de variação contínua: Representam dados em constante atualização.
- Pontos críticos destacados: Indicam eventos fora do padrão, como alturas extremas ou tempos muito longos de operação.

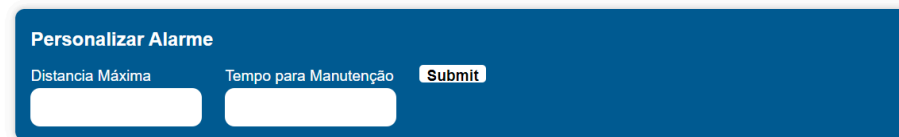
Esses gráficos ajudam a identificar:

- Momentos de pico operacional, como quando uma prensa realiza ciclos mais profundos.
- Possíveis falhas ou interrupções nos sensores.

#### Personalização e Configurações

O painel azul permite ajustes manuais nos parâmetros de funcionamento:

- Altura máxima permitida: Define o limite superior de operação.
- Tempo máximo de uso contínuo: Permite que o usuário estabeleça um tempo seguro de operação.



Embora essas configurações ainda não emitam alertas automáticos em caso de violação, essa funcionalidade está em desenvolvimento e será implementada em breve.

### 5.3.2 Notificação e Gerenciamento de Erros

Em situações de falha, é considerado para a plataforma como requisito funcional, que ainda não está implementado, um sistema de alerta ao usuário imediatamente.

- Alerta visual: Uma mensagem vermelha aparecerá na tela principal, destacando o sensor problemático e o tipo de erro (ver Imagem 5).
- Mensagens no painel:
  - E01: Sensor desconectado.
  - E02: Leitura inválida ou fora do intervalo esperado.

Essas mensagens ajudam o usuário a identificar rapidamente os problemas e tomar as ações necessárias para restaurar o funcionamento.

### 5.3.3 Operação da Case

Ao receber a case do dispositivo, siga os passos abaixo para colocá-la em operação:

#### 1. Posicionamento:

- Posicionar o dispositivo deve ser sob a prensa com o sensor de distância (HC-SR04) apontado para a prensa, para que assim seja possível a coleta da variação de distância
- Escolha um local estável e próximo ao equipamento que será monitorado.
- Certifique-se de que a área ao redor esteja livre de vibrações excessivas ou condições que possam interferir no funcionamento do dispositivo.

#### 2. Conexão à Energia:

- Conecte o cabo de energia da case a uma tomada elétrica adequada.
- Verifique se os LEDs indicadores ou o display acendem, confirmando que o dispositivo está ligado.

#### 3. Conexão com Sensores:

- Certifique-se de que os sensores externos estejam corretamente conectados à case, conforme orientações no **Guia de Montagem**.

#### 4. Teste Inicial:

- Ligue o dispositivo e aguarde o sistema inicializar.
- Observe no display as mensagens ou indicadores de status que confirmam o funcionamento adequado.

#### 5. Conexão ao Dashboard:

- Acesse o dashboard no navegador digitando o endereço configurado (ex.: `localhost:3000` ou outro IP definido).
- Certifique-se de que os dados dos sensores conectados estão sendo exibidos corretamente na interface.

*Recomenda-se manter a case em local seguro e evitar exposição direta a elementos como água ou calor excessivo.*

## 6. Como descartar este dispositivo

O processo de descarte responsável é essencial para mitigar impactos ambientais negativos, como a contaminação do solo e da água, e para alinhar-se com as regulamentações ambientais. Nesse contexto, a elaboração do MVP para o projeto do IPT envolveu a seleção cuidadosa de materiais cujo descarte requer atenção especial, tanto por razões de sustentabilidade quanto por obrigações legais, conforme orienta a **Política Nacional de Resíduos Sólidos (Lei nº 12.305/2010)** do Brasil.

O descarte adequado de componentes eletrônicos e baterias é especialmente crucial. Esses materiais podem conter substâncias nocivas que, se não forem manuseadas corretamente, podem causar danos significativos ao meio ambiente e à saúde pública. Além disso, a reciclagem desses itens é vital para reduzir a demanda por recursos naturais, promovendo os princípios da economia circular e da sustentabilidade.

Visando minimizar o impacto ambiental de nossa solução, elaboramos a **Tabela 2: Materiais e Diretrizes de Descarte**, que auxilia no descarte adequado dos componentes utilizados. Essa tabela detalha cada componente, os materiais que o compõem, os

métodos recomendados de descarte e a vida útil estimada, servindo como um guia prático para o manejo responsável dos resíduos gerados.

**Quadro 2: Materiais e Diretrizes de Descarte**

Componente	Materiais	Método de Descarte	Vida Útil Estimada
<b>Microcontrolador ESP32</b>	Silício, fibra de vidro, cobre	1. Levar a um centro de coleta seletiva para eletrônicos.  2. Desmontagem e separação de componentes para reciclagem especializada.	5-10 anos, dependendo do cuidado no manuseio e condições ambientais.

<b>Display LCD 16x2 com módulo I2C</b>	Vidro, plástico, metais	1. Entregar em pontos de coleta de eletrônicos. 2. Desmontagem para reciclagem de vidro e metais.	5-8 anos, com uso adequado e proteção contra impactos e umidade.
<b>Bateria de Lítio ICR18650</b>	Lítio, metais, plástico	1. Descartar em pontos de coleta específicos para baterias. 2. Processo de neutralização de lítio e	2-3 anos, com uso regular e recargas adequadas.

		reciclagem de metais.	
<b>LED RGB</b>	Vidro, metal, semicondutores	1. Descartar em pontos de coleta de eletrônicos. 2. Desmontagem e reciclagem separada de vidro, metais e semicondutores.	Mais de 50.000 horas de uso se operado em condições ideais e sem sobrecargas elétricas.
<b>Fios Jumper Macho-Macho e Macho-Fêmea Sensor RFID</b>	Plástico, cobre	1. Descartar em pontos de coleta de resíduos eletrônicos. 2. Separação de cobre e plástico para	3-5 anos, dependendo do cuidado no manuseio e armazenamento.

		reciclagem.	
<b>Antena WiFi</b>	Metal, plástico	1. Descartar em pontos de coleta de resíduos mistos ou eletrônicos. 2. Separação e reciclagem de metal e plástico.	4-6 anos, com uso e armazenamento adequados.
<b>Case de Acrílico</b>	Plásticos, acrílico.	1. Levar a pontos de reciclagem que aceitem acrílico. 2. Avaliar possibilidade de reuso antes do descarte.	Vida útil variável, dependente do manuseio e armazenamento da case.

<b>Protoboard</b>	Plástico, metal	1. Descartar em pontos de coleta de resíduos eletrônicos. 2. Desmontagem e separação de metal e plástico para reciclagem apropriada.	Vida útil de 5-10 anos, dependendo do cuidado no uso e armazenamento.

Para auxiliar ainda mais no descarte adequado, recomendamos consultar a **Associação Brasileira de Reciclagem de Eletroeletrônicos e Eletrodomésticos (ABREE)**, uma organização que apoia e trabalha com iniciativas de coleta e reciclagem desses materiais. Mais informações podem ser obtidas em [www.abree.org.br](http://www.abree.org.br).

Realizar o descarte correto dos itens utilizados é de extrema importância para proteger o meio ambiente, reduzir os riscos à saúde pública e promover a sustentabilidade. Adotar práticas

adequadas contribui para a conservação dos recursos naturais e incentiva a transição para uma economia circular, beneficiando toda a sociedade. Com o cumprimento das orientações apresentadas, podemos minimizar impactos negativos e maximizar os benefícios de uma gestão de resíduos responsável do projeto.

## 7. Troubleshooting

O nosso sistema foi projetado para ser simples de operar e de dar manutenção. Caso seja identificado algum problema, muitas vezes a solução pode ser rápida e fácil, como verificar se o dispositivo está ligado, observar os LEDs ou até reiniciar o dispositivo. No entanto, para problemas mais complexos ou técnicos, como falhas nos sensores ou na comunicação do sistema, é necessário acionar uma equipe técnica especializada. Eles serão responsáveis por solucionar problemas que exigem conhecimentos mais específicos, como verificar o funcionamento dos sensores, as conexões internas do dispositivo ou ajustes no código-fonte do sistema.

A seguir, listamos os problemas mais comuns que podem ser encontrados, com instruções de como proceder..

Tabela 3: Possíveis problemas e soluções.

#	Problema	Possível solução
1	Dispositivo não liga	Verifique se o dispositivo está conectado à tomada ou se a bateria está carregada. Se a bateria estiver descarregada, tente trocar por uma nova. Se o dispositivo ainda não ligar, entre em contato com a equipe técnica para verificar se há algum defeito interno.
2	Display LCD não exibe informações	Verifique se o dispositivo está ligado. Caso o display continue em branco ou não mostre as informações corretamente, tente reiniciar o dispositivo (desligando e ligando novamente). Se o problema continuar, a equipe técnica precisará verificar as conexões internas do dispositivo.
3	LEDs não acendem ou estão em uma cor estranha	Verifique a cor do LED: Se o LED estiver vermelho ou piscando, pode ser um sinal de que algo não está funcionando

		corretamente. Tente reiniciar o dispositivo. Se o problema persistir, entre em contato com a equipe técnica para investigar falhas nos sensores ou na comunicação interna.
4	Dispositivo sem conexão com a rede (problemas com a internet)	Verifique se a rede Wi-Fi está funcionando corretamente. Caso o sinal da rede esteja fraco, tente mover o dispositivo para um local mais próximo do roteador. Se a conexão não for restabelecida, a equipe técnica precisa verificar se o dispositivo está configurado corretamente para acessar a rede.
5	Aplicação não está mostrando dados atualizados	Verifique se o dispositivo está ligado e se o display indica que ele está funcionando corretamente. Se a aplicação não estiver exibindo dados novos, tente sair e entrar novamente no sistema. Se o problema continuar, entre em contato com a equipe técnica para investigar se há falhas na

	comunicação do sistema.
--	-------------------------

6	Erros ao tentar acessar o aplicativo (erro de login ou de tela em branco)	Verifique sua conexão com a internet e tente recarregar a página da aplicação. Se o problema persistir, tente limpar o cache do navegador ou reiniciar o dispositivo que está utilizando para acessar o aplicativo. Caso o erro continue, avise a equipe técnica para verificar se há problemas no sistema ou no servidor.
7	Os dados no aplicativo estão inconsistentes ou com falhas (valores errados)	Observe os dados no display do dispositivo e compare com o que aparece no aplicativo. Se houver grandes discrepâncias, reinicie o dispositivo e veja se o erro é corrigido. Se isso não resolver, entre em contato com a equipe técnica para revisar o sistema e verificar se os dados estão sendo processados corretamente.



8	Sensores não estão funcionando corretamente	Verifique no display se os dados do sensor estão sendo exibidos corretamente. Se você perceber valores estranhos ou inconsistentes, tente reiniciar o dispositivo. Caso o problema persista, entre em contato com a equipe técnica para que ela verifique as conexões do sensor ou, se necessário, substitua o sensor.

10	O sistema apresenta demora para carregar os dados no dashboard	Verifique se sua conexão à internet está estável. Se a conexão estiver funcionando corretamente, mas o problema persistir, entre em contato com a equipe técnica para verificar se o servidor ou o sistema está funcionando corretamente.
----	--	---

11	Alerta de falha no sistema de monitoramento não é recebido	Verifique se o dispositivo está corretamente conectado à rede e funcionando. Caso o alerta de falha não seja recebido, tente reiniciar o dispositivo. Se o problema persistir, entre em contato com a equipe técnica para revisar a configuração do sistema de alertas.
----	--	---

A manutenção diária do sistema é simples e envolve a verificação de alguns componentes básicos, como os LEDs, o display e o funcionamento do dispositivo. Para problemas mais complexos, que envolvem a comunicação do sistema ou falhas nos sensores, é essencial acionar a equipe técnica. Eles possuem as ferramentas e o conhecimento necessários para diagnosticar e corrigir problemas mais profundos, garantindo que o sistema continue funcionando corretamente. Não hesite em entrar em contato com a equipe técnica sempre que necessário.