

# Manual de Instruções

**TurtleBee**  
**Gerda**

## Controle do Documento

### Histórico de revisões

Data	Autor	Versão	Resumo da atividade
07/06	Giovanna Rodrigues	1 e 2	Preenchimentos das seções 1 e 2.
11/06	Patricia Honorato	3 e 4	Preenchimento de toda a seção 3 e 4.
11/06	Patricia Honorato	1 e 2	Edição do texto descritivo da solução, identificação das tabelas e texto de descrição dos requisitos de conectividade.
20/06	Patricia Honorato	4	Preenchimento das seções: 4.3 Guia de prevenção e mitigação de acidentes, 4.1 Riscos Operacionais e 4.2 Adequação do local de operação do robô.

# Índice

<b>1. Introdução</b>	<b>4</b>
1.1. Solução	4
1.2. Arquitetura da Solução	5
<b>2. Componentes e Recursos</b>	<b>7</b>
2.1. Componentes de hardware	7
2.2. Componentes externos	8
2.3. Requisitos de conectividade	10
<b>3. Guia de Montagem</b>	<b>11</b>
3.1 Guia de montagem e inicialização do sistema	11
3.1.2 Montagem do TurtleBot3	11
3.1.3 Inicialização do Sistema	11
3.1.3.1 Banco de Dados	11
3.1.3.2 Backend	12
3.1.3.3 Frontend	13
3.1.3.4 Comunicação	14
3.2 Instruções operacionais	14
3.2.1 Operando o TurtleBot3	14
3.2.2 Precauções de Segurança ao Operar o TurtleBot3	15
3.3 Guia de manutenção e soluções para os problemas mais comuns	16
3.3.1 Manutenção do TurtleBot3	16
3.3.2 Soluções para Problemas Comuns	16
<b>4. Considerações de segurança</b>	<b>18</b>

4.1 Riscos Operacionais	18
4.2 Adequação do local de operação do robô	18
4.3 Guia de prevenção e mitigação de acidentes	18
4.3.1 Proteção contra comandos indesejados	18
4.3.2 Prevenção de Acidentes com Obstáculos	19
4.3.3 Mapeamento e Análise de Riscos	20
4.3.4 Validando a Eficiência dos Sistemas de Segurança	21
<b>5. Créditos</b>	<b>22</b>

# 1. Introdução

## 1.1. Solução

A solução proposta é um sistema integrado de monitoramento de ambientes industriais, que utiliza um robô modelo TurtleBot 3 Burger. Este robô é equipado com um sensor de gases voláteis MQ2 e uma webcam, permitindo a detecção de gases potencialmente perigosos e a visualização em tempo real do ambiente.

O TurtleBot3 é controlado remotamente através de uma interface web amigável, desenvolvida com Next.js e React.js, que permite aos usuários interagir com o robô e solicitar ações específicas. A interface web se comunica com o backend, construído com Docker e Python (Sanic), que gerencia a comunicação com o TurtleBot3 através de uma rede ROS2.

O sistema também inclui um banco de dados MySQL, construído com o ORM Prisma e hospedado na AWS, que armazena informações e dados relevantes para o projeto. Isso permite o acompanhamento e a análise das condições do ambiente industrial ao longo do tempo.

Com este sistema, é possível monitorar o ambiente industrial sem a necessidade de um operador estar fisicamente presente, aumentando a segurança e minimizando a exposição a possíveis riscos. Além disso, o sistema fornece dados acurados e uma visão

detalhada da área, permitindo uma resposta rápida a qualquer mudança nas condições do ambiente.

## 1.2. Arquitetura da Solução

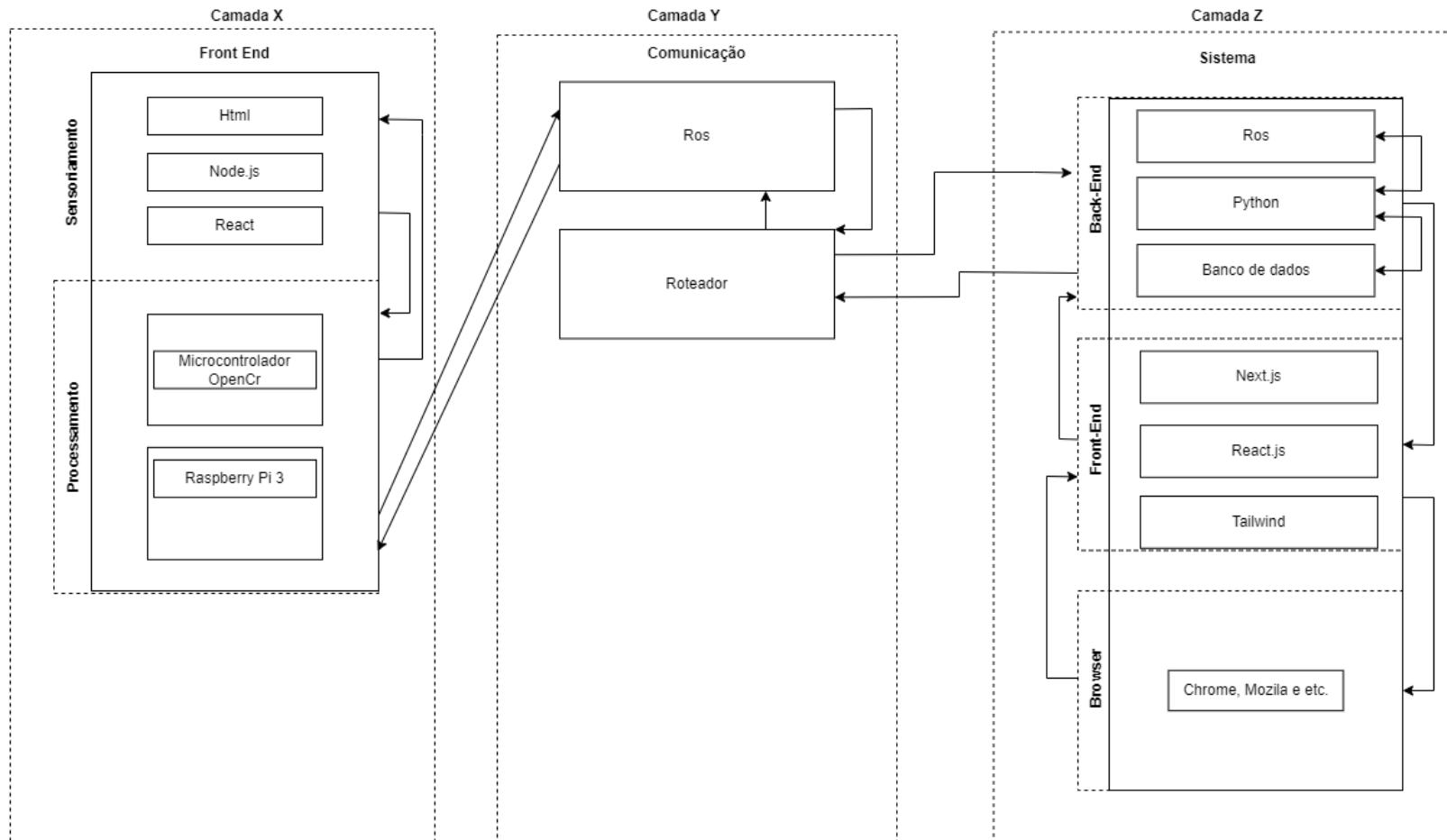


Figura 1 -Diagrama de Blocos (Fonte: Autores, 2023)

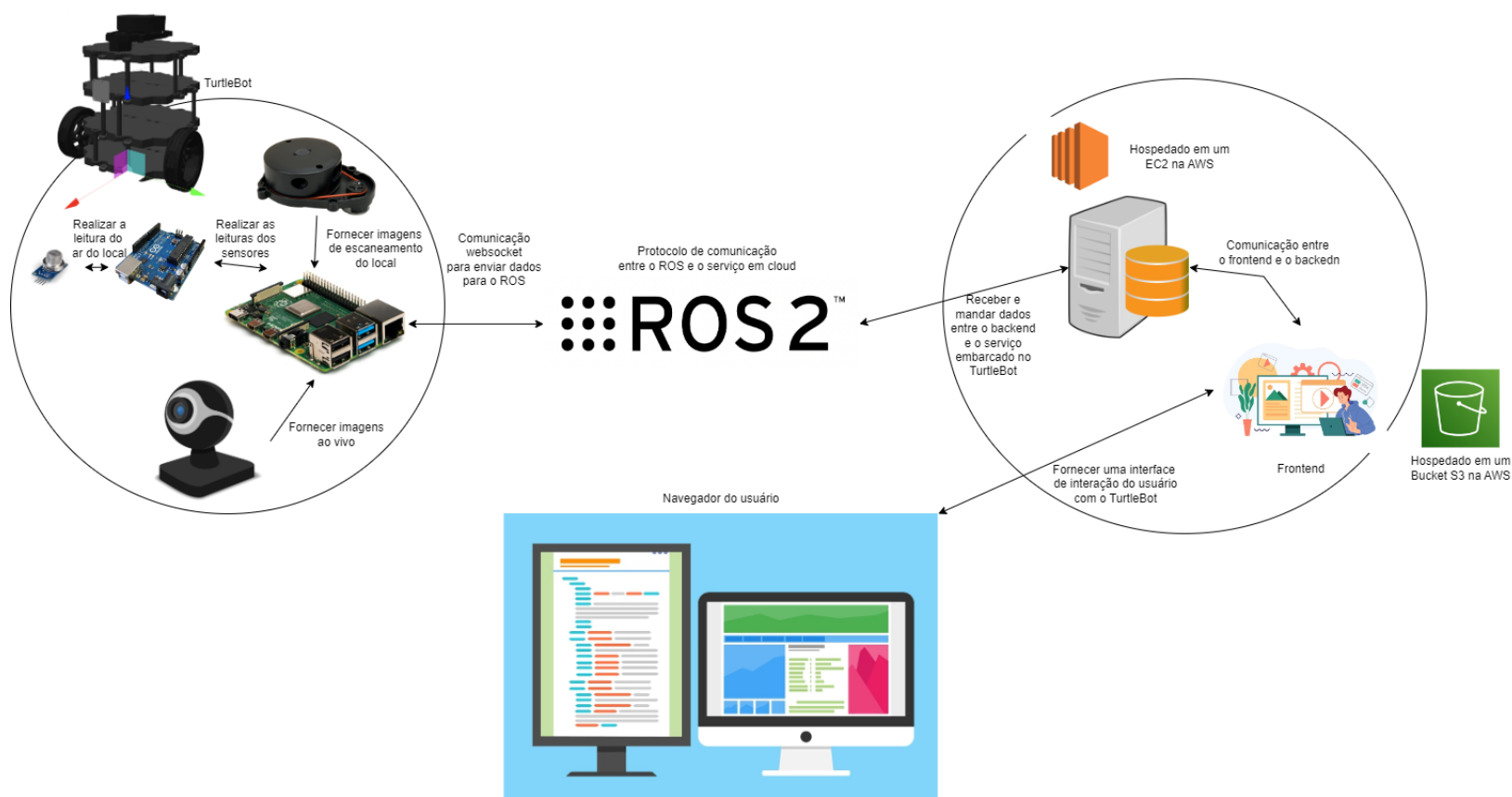






Figura 2 - Diagrama da arquitetura da solução do projeto

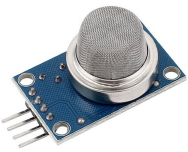

(Fonte: Autores, 2023)

## 2. Componentes e Recursos

### 2.1. Componentes de hardware

Representação	Categoria	Componente	Descrição de uso
	Movimentação	TurtleBot3 burger	O robô é responsável pela movimentação, possuindo um conjunto de duas rodas e um motor. Além de ser equipado com os demais componentes de hardware mencionados a seguir.
	Computador embarcado	Raspberry Pi 3	Se comunica com o backend e processa informações de todos os componentes.
	Entrada	Sensor Lidar 360°	Esse sensor é capaz de detectar objetos ao seu redor por meio de um feixe de luz, assim, será usado para evitar a colisão do robô com possíveis obstáculos.
	Microcontrolador	Arduino UNO	Este microcontrolador é usado na leitura dos dados providos pelo sensor de gás.



	Entrada	Sensor de gás MQ2	Sensor responsável por detectar gases voláteis no ambiente isolado.
	Entrada	Webcam	A webcam é usada para a captação de imagens que serão exibidas ao vivo no frontend.

**Tabela 1 -Componentes do hardware**

(Fonte: Autores, 2023)

## 2.2. Componentes externos

Componente	Categoria	Versão/Serviço	Descrição de uso
Laptop ou Desktop	Dispositivo	Windows 7+ ou posterior MacOS 10.6 Snow leopard ou posterior Linux OS	Utilização da plataforma Web.
Banco de Dados - MySQL	Serviços Cloud	Amazon RDS	Armazenamento das informações, como quais robôs e rotas foram adicionados e quais análises já foram realizadas.

Hospedagem Backend	Serviços Cloud	Amazon EC2	Servidor para a execução de serviços backend requeridos pela aplicação.
Hospedagem Plataforma	Serviços Cloud	Amazon S3	Interface gráfica para uma melhor comunicação com o usuário.
Edição de Código Arduino	Software	Arduino IDE	Edição e compilação do código pertencente ao Arduino.
Edição de Código Backend/Frontend	Software	Visual Studio Code v1.78	Edição e configuração das plataformas backend e frontend da aplicação.
Instalação de Packages para desenvolvimento Frontend/Backend	Software	Package Installer for Python v23.1.2	Instalação de pacotes python para o desenvolvimento das plataformas.
SQL Client Software Application	Software	DBeaver v23.1.0	Manipulação e modificação do banco de dados.
REST Client Testing	Software	Insomnia v2023.2.2 Postman v10.14	Simulação de requisições a rotas do backend.
WSL	Software	1.2.5.0	Permite executar um ambiente Linux em um dispositivo do sistema operacional Windows.
Ubuntu	Sistema operacional	20.04 LTS	Sistema operacional de distribuição Linux.

**Tabela 2 - componentes externos (Fonte: Autores, 2023)**

## 2.3. Requisitos de conectividade

Para garantir a operação eficiente e eficaz do TurtleBot3 e sua interface web, é necessário atender a vários requisitos de conectividade. Esses requisitos envolvem o uso de vários protocolos de rede e especificações de software, conforme detalhado na tabela abaixo:

Serviço	Categoria	Descrição de uso
TCP/IP	Protocolo de Rede	Protocolo de controle de transmissão, viabiliza a troca de informação entre dispositivo Origem e Destino.
Internet Protocol Versão 4 - IPv4	Protocolo de Rede	Endereço único para identificação do dispositivo na internet ou em uma rede local.
Media Access Control Protocol - MAC	Protocolo de Rede	Permite a conexão de múltiplos dispositivos na internet ou em uma rede local.
HyperText Transfer Protocol Secure - HTTPS	Protocolo de Rede	Protocolo que estabelece uma conexão entre o cliente e o servidor remoto.
Sanic	Especificações	Framework para Python que permite a construção de um servidor web assíncrono.
Prisma Node Package	Especificações	É uma ORM open-source usada na manipulação do banco de dados.
ROS	Especificações	Framework de software utilizado no desenvolvimento de softwares para robôs.

**Tabela 3 - Requisitos de conectividade (Fonte: autor, 2023)**

## 3. Guia de Montagem

### 3.1 Guia de montagem e inicialização do sistema

Neste tópico, serão apresentadas instruções detalhadas sobre como montar o TurtleBot3 e inicializar o sistema. Serão descritos os componentes do sistema do projeto e como deve ser realizada a configuração de cada parte.

#### 3.1.2 Montagem do TurtleBot3

A montagem do TurtleBot3 é fornecida diretamente pelo fabricante e possui os seguintes componentes: Microcontrolador OpenCR, Sensor de gás MQ2, Raspberry Pi 3, Webcam e Sensor Lidar 360°. Para sua correta instalação, os passos a serem feitos estão presentes no manual disponibilizado pelo fabricante. Para ligar o robô é necessário empurrar o botão presente na sua parte frontal, próximo a bateria. Após a execução sonora indicando que ele está ligado, ele está pronto para se conectar à rede.

Para a configuração do software do TurtleBot3 e conexão na rede do usuário, é necessário a instalação de diferentes bibliotecas e pacotes, conforme documentado no material didático do Prof.

Nicola do Inteli, disponível no seguinte link: [Guia de Configuração do TurtleBot3](#). Este guia oferece uma orientação passo a passo para o processo de instalação, assegurando que o TurtleBot3 funcione corretamente em concordância com seus componentes.

#### 3.1.3 Inicialização do Sistema

A inicialização do sistema envolve a configuração do banco de dados, do backend e do frontend. Nos tópicos a seguir serão descritos as ações necessárias para acesso ao sistema.

##### 3.1.3.1 Banco de Dados

O banco de dados foi construído usando o ORM Prisma e MySQL. Foi feito o deploy do banco de dados na AWS, especificamente no servidor RDS. Grupos de segurança foram criados para proteger o banco de dados, garantindo que apenas os membros da equipe tenham acesso a ele.

Para inicializar o banco de dados, você deve acessar o servidor RDS na AWS e iniciar o serviço do banco de dados. Os passos estão descritos abaixo:

## 1. Acesso à AWS:

Primeiro, você precisará acessar o console da AWS. Nessa etapa será necessário inserir as credenciais de acesso (ID de acesso e chave de acesso).

## 2. Navegação até o RDS:

No painel de serviços da AWS, procure pelo serviço RDS (Relational Database Service). Clique para acessar o console do RDS.

## 3. Seleção da Instância de Banco de Dados:

No console do RDS, você verá uma lista de instâncias de banco de dados. Localize a instância de banco de dados chamada `bd-gerdau.csqbki0w74l.us-east-1.rds.amazonaws.com` e clique nela.

## 4. Inicialização da Instância de Banco de Dados:

Com a instância de banco de dados selecionada, você terá a opção de iniciar o serviço do banco de dados. Clique em 'Iniciar' para ativar o banco de dados.

As credenciais de acesso à AWS e à instância do banco de dados são as seguintes:

**AWS\_ACCESS\_KEY\_ID**="ASIAZRYMG3A4O2H4IIF6"

**AWS\_SECRET\_ACCESS\_KEY\_ID**="Oq/EXgvSl+rkSuwQ9mWzXp  
nVDpHfRTzGMN7SMc4Jaws\_session\_token=FwoGZXIVYXdzEP  
7////////wEaDGoXeg2qyOSQiEDfayLNAYC8+wPileP71XIGxazfgf  
W84qalqUfcwj3yp4DHHUt+G9QNpjkj9qfoOtP/KLzpagvKvWY21  
dfO7X1Nfv1rhN+TnKaG6o/rzdJ0zpaXJIHtQU/fLTU9DYKdykx/xaQ  
aRuYWfmcP0Nhz5oWfKpjZxPfOzf6TAAhx2BpctfrlebwJ+IGXyz6  
FNowXABYirQqixAx04Oheo3EfE8AWN3L1HajExN+5O58URZ1C  
W1cEbz/eh6FzKG8BQpz0pezerESkleMgGNoVXXhw8yj2Jvsotoj  
5owYyLU2ASJebJ9mIN7AygkO4mdFMgOAuatziJi8u47jjoeneO  
fwW0L7okOWaQSVsaQ=="

**AWS\_REGION**="us-east-1"

**DATABASE\_URL**="mysql://admin:admin123@bd-gerdau.csqbki  
c0w74l.us-east-1.rds.amazonaws.com:3306/projeto\_gerdau?co  
nnection\_limit=5"

### 3.1.3.2 Backend

O backend foi projetado e construído utilizando uma série de tecnologias poderosas e robustas:

- **Docker:** Permite a criação e gerenciamento de contêineres que encapsulam o serviço do backend, facilitando assim o deploy e a portabilidade em qualquer serviço cloud. Sua instalação é realizada seguindo os procedimentos presentes em: <https://docs.docker.com/engine/install/>

O arquivo docker está disponível na **pasta src/backend** desse projeto.

- **Python (Sanic):** Este framework web foi utilizado para a criação da API e para gerenciar a lógica do servidor. Isso estabelece a comunicação entre o usuário e as atividades do TurtleBot. A instalação é realizada seguindo os procedimentos presentes em: <https://www.python.org/downloads/>
- **Banco de Dados SQL:** Esta tecnologia é responsável por armazenar todas as informações e dados relevantes para o funcionamento e a operação do TurtleBot. A instalação é feita seguindo os procedimentos presentes em: <https://dev.mysql.com/doc/refman/8.0/en/installing.html>

O backend está hospedado em um serviço cloud da AWS e utiliza a rede ROS2 para comunicação bidirecional com o TurtleBot3.

Para inicializar o backend, você deve acessar o serviço cloud onde o backend está hospedado e iniciar o serviço. Após essa etapa, é necessário inicializar o docker e realizar testes de conectividade, conforme descrito abaixo:

## 1. Acesso à Instância:

Uma vez iniciada a instância do banco de dados da AWS, você pode se conectar a ela via SSH. No console do EC2, selecione a instância, clique em 'Conectar' e siga as instruções fornecidas.

## 2. Inicialização do Docker:

Na instância, navegue até o diretório que contém o arquivo docker-compose.yml. Inicialize o Docker usando o comando `docker-compose up -d`. Este comando iniciará todos os serviços listados no arquivo docker-compose.yml.

## 3. Verificação de Status:

Para verificar se o backend está funcionando corretamente, use o comando `docker ps`. Este comando listará todos os contêineres do Docker atualmente em execução e seu status. Certifique-se de que o contêiner do backend está listado e seu status é 'Up'.

## 4. Teste da Conexão com o TurtleBot3:

Para verificar se o backend pode se comunicar com o TurtleBot através da rede ROS2, executando um comando do ROS2. Certifique-se de que o TurtleBot3 está ligado e conectado à mesma rede que o backend.

### 3.1.3.3 Frontend

O frontend foi desenvolvido com Next.js e React.js, e é hospedado na AWS. A interface permite que os usuários interajam com o TurtleBot e solicitem ações específicas.

Para inicializar o frontend, você deve acessar a pasta `src/frontend/` do repositório do grupo 4 presente nesse link do [GitHub](#).

E após, executar o comando:

```
npm run dev
```

E acessar o link:

<http://localhost:3000/>

Assim é possível visualizar o que foi implementado no frontend desse projeto.

**O acesso da interface web pela AWS é feito pelo seguinte link:**

<https://turtlecontroller.s3.amazonaws.com/index.html>

### 3.1.2.3.4 Comunicação

Para verificar se o sistema está funcionando corretamente, você deve verificar se todos os componentes estão funcionando e se o TurtleBot está se comunicando corretamente com o backend e o frontend. A interface web indica o estado atual do TurtleBot e permite que o usuário realize ações.

## 3.2 Instruções operacionais

Nesse tópico serão descritas as etapas a serem realizadas para operar o robô TurtleBot3 via interface web.

### 3.2.1 Operando o TurtleBot3

O TurtleBot3 é controlado através da interface web, que se comunica com o backend para enviar comandos ao robô. Abaixo estão detalhadas o conjunto de ações a serem realizadas:

#### 1. Acesso a interface web através do serviço cloud onde o frontend está hospedado.

Na interface web, o usuário visualizar várias opções para controlar o TurtleBot3. Isso inclui mover o robô, usar a webcam e o sensor Lidar 360°, e monitorar os dados do sensor de gás MQ2.

Selecione a ação que você deseja que o TurtleBot3 execute. O comando será enviado ao backend, que por sua vez enviará o comando ao TurtleBot3.

Você pode monitorar o estado do TurtleBot3 através da interface web. Isso inclui a posição atual do robô, as imagens da webcam e do sensor Lidar 360°, e os dados do sensor de gás MQ2.

#### 2. Usando a Interface Web

Abaixo estão listados as tarefas que podem ser feitas pelo usuário:

- **Emissão de relatório:** A interface permite que os usuários gerem relatórios sobre as atividades do TurtleBot3. Para emitir um relatório, vá para a seção de relatórios e siga as instruções na tela.
- **Gerenciamento de análises:** A interface permite que os usuários visualizem, editem, excluam e cadastrem análises. Para gerenciar análises, vá para a seção de análises e siga as instruções na tela.
- **Teste dos sensores:** A interface permite que os usuários solicitem um teste dos sensores do TurtleBot3. Para solicitar um teste, vá para a seção de testes de sensores e siga as instruções na tela.
- **Acompanhamento em tempo real:** A interface permite que os usuários acompanhem em tempo real as atividades do TurtleBot3. Para acompanhar em tempo real, vá para a seção de acompanhamento em tempo real e siga as instruções na tela.

### 3. Interpretando os Dados dos Sensores

Os dados dos sensores do TurtleBot3 são exibidos na interface web. Isso inclui os dados do sensor de gás MQ2, as imagens da webcam e do sensor Lidar 360°. Para interpretar os dados dos sensores, é necessário que o usuário entenda o que cada sensor faz e como os dados são apresentados na interface web. Esses parâmetros serão estabelecidos pelo próprio usuário, que determinará quais condições são normais, seguras ou de risco.

### 3.2.2 Precauções de Segurança ao Operar o TurtleBot3

Ao operar o TurtleBot3, é necessário se atentar às condições do ambiente e do robô. Abaixo estão detalhadas algumas dessas situações:

- Realize o monitoramento da tensão da bateria: não utilize o robô quando os níveis de energia estiverem abaixo do normal. Para essa situação o robô emitirá um alerta sonoro para que o operador tome uma ação de parada do sistema.
- Certifique-se de que o TurtleBot3 está em uma área segura antes de iniciar qualquer operação, que não causará danos em seus componentes.
- Utilize o TurtleBot3 em áreas com gases perigosos ou inflamáveis apenas se o robô estiver equipado com sensores.
- Sempre monitore o estado do TurtleBot3 através da interface web durante a operação.
- Em caso de qualquer problema ou comportamento inesperado, pare a operação do robô.



## 3.3 Guia de manutenção e soluções para os problemas mais comuns

Abaixo serão fornecidas informações de manutenção do TurtleBot3 e para solucionar problemas comuns que podem ocorrer durante a operação e execução do sistema.

### 3.3.1 Manutenção do TurtleBot3

A manutenção regular do TurtleBot3 é essencial para garantir seu funcionamento adequado. Aqui estão algumas diretrizes básicas para a manutenção do TurtleBot3:

#### 1. Limpeza:

É necessário manter o TurtleBot3 limpo para assegurar uma melhor performance dos sensores e funcionamento dos componentes. Para isso, pode ser utilizado um pano seco para limpar a poeira e a sujeira do robô. Não pode-se utilizar produtos de limpeza ou água, pois podem danificar os componentes eletrônicos.

#### 2. Inspeção dos componentes:

Verifique regularmente os componentes do TurtleBot3, incluindo o Microcontrolador OpenCR, o Sensor de gás MQ2, o Raspberry Pi 3, a Webcam e o Sensor Lidar 360°. Se você notar qualquer dano ou desgaste, substitua o componente danificado. Em grande parte dos casos o próprio sistema da interface web indicará que o componente não está mais funcionando.

#### 3. Atualização de software:

É necessário manter o software do TurtleBot3 atualizado para garantir que ele tenha as últimas correções de bugs e melhorias de desempenho. Isso inclui o software do backend e do frontend, bem como o firmware do TurtleBot3.

### 3.3.2 Soluções para Problemas Comuns

#### 1. Problemas de conexão com o backend:

Se o TurtleBot3 não conseguir se conectar ao backend, verifique se o backend está funcionando corretamente e se o TurtleBot3 está conectado à rede correta. Se o problema persistir, tente reiniciar o TurtleBot3 e o backend.

## **2. Problemas com os sensores:**

Se um sensor não estiver funcionando corretamente, verifique se o sensor está limpo e livre de obstruções. Se o sensor ainda não funcionar, pode ser necessário substituí-lo.

## **3. Problemas com a interface web:**

Se a interface web não estiver funcionando corretamente, tente atualizar a página ou usar um navegador diferente. Se o problema persistir, pode haver um problema com o frontend, e você deve entrar em contato com o suporte técnico.

## **4. Assistência adicional:**

Se você não conseguir resolver um problema por conta própria, entre em contato com o suporte técnico de alguma das tecnologias utilizadas nesse projeto. Certifique-se de fornecer o máximo de detalhes possível sobre o problema, incluindo quaisquer mensagens de erro que você possa ter recebido.

## 4. Considerações de segurança

### 4.1 Riscos Operacionais

Nesse projeto os riscos operacionais abrangem desde falhas de componentes devido à exposição à umidade, desgaste mecânico, falha de software até interrupções de comunicação e problemas de fonte de energia. Além disso, a exposição a condições de ambientes confinados e danos nos dutos durante a operações são riscos tangíveis a este sistema.

Ademais, é imperativo considerar a exposição potencial do robô a substâncias químicas durante a operação. Visando-se mitigar esses riscos, podem-se ser desenvolvidas estratégias que incluem manutenções preventivas regulares, uso de equipamento apropriados, aplicação de protocolos de segurança rigorosos e treinamentos frequentes da equipe, além da instalação de sistemas de detecção e prevenção de falhas.

### 4.2 Adequação do local de operação do robô

A análise do ambiente em que o robô irá operar é crucial para garantir a segurança do processo. Nesse projeto, diversos fatores são levados em consideração, como um espaço limitado para a

operação, presença de obstáculos físicos nas tubulações, substâncias químicas ou condições ambientais, como a temperatura e umidade do ar.

A presença de rotas de emergência na ocasião de situações adversas são fundamentais para a segurança da operação. Ao assegurar a adequação do local de operação se minimiza a probabilidade de riscos operacionais e aumenta a performance e eficiência do robô ao longo desse processo.

### 4.3 Guia de prevenção e mitigação de acidentes

Os sistemas de segurança aplicados a este projeto visam realizar a prevenção de acidentes no backend e frontend desse projeto, incluindo proteção contra comandos irregulares e a segurança da bateria do robô.

#### 4.3.1 Proteção contra comandos indesejados

No frontend e backend do projeto foram implementados as seguintes funcionalidades para contenção do acionamento de comandos indesejados:

- A. **Dupla verificação de etapas:** Nesse método, é requisitado que o usuário confirme as ações antes da execução, evitando comandos acidentais.

B. **Informações de feedback em tempo real:** Fornece atualizações imediatas sobre as ações do robô, permitindo a intervenção rápida.

- C. **Validação de entrada de dados:** Permite que apenas dados válidos e seguros sejam aceitos pelo sistema, evitando dados maliciosos.
- D. **Autenticação para todas as rotas do back-end:** As rotas do back-end estão protegidas por autenticação, exigindo um token de acesso para garantir que apenas usuários autorizados tenham acesso.
- E. **Verificação da conexão do robô com o servidor:** Com esse método, garante-se que o robô esteja conectado ao servidor antes que comandos sejam enviados, evitando falhas de comunicação.
- F. **Checagem de comunicação direta com o robô:** Permite verificar se o robô está respondendo conforme esperado ou não, identificando-se possíveis falhas no sistema.

### 4.3.2 Prevenção de Acidentes com Obstáculos

Para evitar-se obstáculos na trajetória do robô, foram implementados os seguintes recursos para mapeamento e navegação:

- A. **Limite de proximidade com obstáculos:** O robô é configurado para ter um limite de proximidade com todos os obstáculos detectados nos dutos de ventilação, isso permite que o robô minimize a possibilidade de colisões e acidentes.
- B. **Definição de rota de emergência:** Em casos de falhas inesperadas ou outras situações, o robô consegue seguir uma rota pré-definida que o encaminhará para um local seguro.
- C. **Algoritmo de prevenção de colisões:** No backend do projeto existe um algoritmo que permite que sejam projetadas e calculadas as rotas mais eficientes para o robô, levando-se em consideração a presença de obstáculos no caminho.
- D. **Planejamento da trajetória e otimização de rotas:** O robô faz uso do algoritmo de busca em profundidade para o planejamento das trajetórias. Esse algoritmo percorre os nós de um grafo (representando o sistema de ventilação,

### 4.3.3 Mapeamento e Análise de Riscos

E. **Sistema de visão computacional para análise de rachaduras nas paredes das tubulações:** Esse sistema recebe imagens da câmera que está acoplada no robô e em seguida as processa por meio do modelo de detecção YOLOv8, e fornece para o usuário diagnósticos sobre as paredes da tubulação, informando se existem rachaduras. Essa funcionalidade permite que sejam identificadas

- Risco de exposição a substâncias perigosas: muito alto (90%)
- Falha dos componentes devido à exposição à umidade: baixo (30%)
- Desvio de rota causado por interferências externas: moderado (70%)
- Falhas mecânicas: alto (70%)
- Falha dos componentes após contato com a água: moderado (50%)
- Degradação da câmera acoplada ao robô: muito baixo (30%)
- Falha do sistema de gerenciamento do robô: baixo (30%)
- Risco do robô cair em ambientes confinados: moderado (30%)
- Corrosão do chassi do robô: alto (30%)
- Possibilidade de explosão dentro da tubulação devido à bateria: muito baixo (10%)
- Falhas causadas por erro humano: moderado (10%)

- Bateria se esgotando devido a operar com menos de 30% de carga: muito alto (10%)

Essa análise permite uma compreensão dos potenciais riscos que o robô enfrentará durante as operações, possibilitando que sejam implementadas as medidas preventivas descritas neste manual ou a elaboração de outras.

#### 4.3.4 Validando a Eficiência dos Sistemas de Segurança

A verificação e validação da eficácia dos sistemas de segurança, são responsáveis por assegurar que o robô esteja funcionando conforme projetado e fornecem uma robusta defesa contra potenciais ameaças à operação. Abaixo serão apresentadas os métodos avaliativos que compõem o presente projeto:

- Monitoramento da tensão da bateria:** O monitoramento da tensão da bateria é primordial para a operação segura e eficiente do TurtleBot. No robô esse recurso está implementado no formato de um alarme sonoro que é emitido quando a bateria está prestes a se esgotar. Este sistema de alarme é um exemplo de uma medida de proteção contra sobrecarga da bateria. A sobrecarga pode danificar a bateria, reduzindo sua capacidade e vida útil.
- Testes em ambientes confinados controlados:** A realização de testes em ambientes similares aos de operação permite avaliar o desempenho do robô em condições práticas e reais,

garantindo assim a eficácia dos sistemas de segurança em um ambiente operacional genuíno.

- Teste e validação do sistema de otimização de rotas:** A eficiência desse sistema pode ser validada através de simulações e testes em ambientes confinados controlados, que buscam verificar a precisão do algoritmo em diversos cenários e condições. Ao garantir que o robô possa navegar com segurança e eficiência, reduz-se a probabilidade de acidentes, aumentando a confiabilidade do robô em suas operações.
- Manutenção preventiva:** A realização de manutenções preventivas periódicas permitem detectar e corrigir falhas antes que ocorra falhas críticas no sistema.
- Treinamento da equipe:** O treinamento da equipe responsável por manusear o robô é uma terna importante para assegurar que o sistema será executado corretamente e serão acionadas ações corretivas quando necessárias, para garantir sua funcionalidade.

## 5. Créditos

### Equipe estudantes:

[Felipe Campos](#)

[Giovanna Rodrigues](#)

[Gustavo Oliveira](#)

[Henrique Lemos](#)

[João Pedro Carazzato](#)

[Luiz Borges](#)

[Patricia Honorato](#)

### Equipe professores:

Professor de Programação - [Rodrigo Mangoni Nicola](#)

Professor Orientador - [Murilo Zanini de Carvalho](#)