

Manual de Instruções

Safe McQueen

Gerdau

Índice

1. Introdução

- 1.1. Solução
- 1.2. Arquitetura da Solução

2. Componentes e Recursos

- 2.1. Componentes de hardware
- 2.2. Componentes externos

3. Guia de Montagem

- 3.1 Montagem do circuito com DHT11;
- 3.2 Montagem do circuito com o MQ-135;
- 3.3 Conexão com a câmera e raspberry pi;
- 3.4 Conexão com os sensores e robô;

4. Guia de Instalação

4.1 Hardware;

Tabela de Conexões

Ambientação e alimentação elétrica

4.2 Software;

GitHub

VSCode

Arduino IDE

5. Guia de Operação

5.1 Consulta de medidas atuais;

5.2 Histórico;

] 5.1 Criação de uma nota rota;

6. Troubleshooting

1. Introdução

1.1. Solução

O objetivo do desenvolvimento é construir uma simulação de um robô capaz de se mover em ambientes de espaço confinado e áreas de difícil acesso. O robô terá a capacidade de coletar dados a partir dos sensores (principalmente de oxigênio e outros gases), também irá utilizar filmagens para apoiar na inspeção prévia da estrutura e no suporte na execução das atividades planejadas.

Para alcançar esse objetivo, foi adotada uma metodologia abrangente no desenvolvimento da solução, que envolve diferentes aspectos. Na parte da simulação existe o controle do robô e seus periféricos, esse tópico lida primordialmente com a comunicação entre os componentes e a simulação do robô. Além disso, foi utilizado o ROS2, que é um framework de robótica que permite a comunicação entre o sistema, consequentemente ele também permite a simulação do robô, o que permite testar como seria o comportamento do robô mesmo apenas com o ambiente virtual antes de sua implementação física.



Figura 1 - Logo da equipe SAFE MC QUEEN

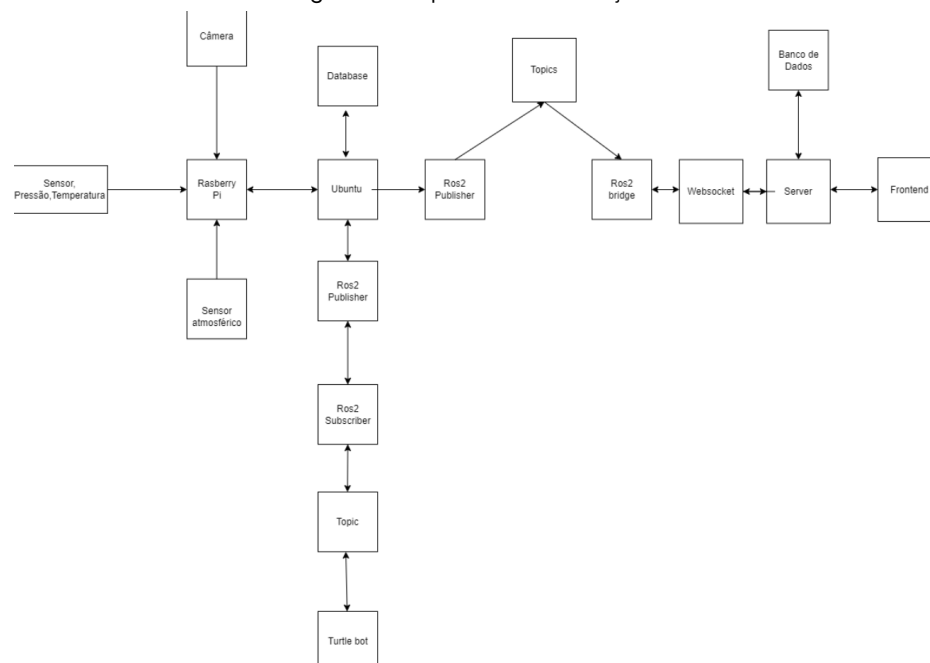
1.2. Arquitetura da Solução

A arquitetura do projeto é composta por duas partes distintas. A primeira parte envolve o sistema responsável pelo controle do robô e seus periféricos. Essa parte abrange a comunicação entre os diferentes componentes do sistema, bem como a simulação do robô. Para realizar essas funcionalidades, utiliza-se o ROS2, um framework de robótica que facilita a comunicação entre os componentes do sistema e permite a simulação do robô. Essa abordagem é vantajosa, pois possibilita o desenvolvimento e teste do sistema sem a necessidade de um robô físico.

A segunda parte da arquitetura consiste em uma solução web, que permite a visualização dos dados coletados pelo robô. Essa parte do sistema é desenvolvida utilizando o React, um framework de desenvolvimento web, o Flask, um framework de desenvolvimento web para Python, e o SQLite, um banco de dados embarcado. Para ser possível o funcionamento do projeto mesmo com a restrição de acesso constante à internet por parte do robô, foi implementado um banco de dados embarcado para armazenar os dados coletados pelo robô. Além disso, agora o robô envia os dados para o banco de dados apenas ao final da rota.

Adicionalmente, está sendo considerado o uso de uma bridge entre o ROS2 e o WebSocket, permitindo que o robô envie os dados para a aplicação web de forma mais eficiente. Essa integração proporcionará uma comunicação direta entre o sistema ROS2 e a aplicação web, possibilitando o monitoramento real dos dados coletados pelo robô. Diante desse cenário e levando em consideração a limitação de acesso constante à internet pelo robô, a introdução de um banco de dados embarcado para armazenamento dos dados coletados e a possível integração entre ROS2 e WebSocket foi primordial, e visa priorizar a coleta e visualização dos dados, contribuindo para a eficiência e funcionalidade do sistema como um todo.

Figura 2 - Arquitetura da Solução



2. Componentes e Recursos

2.1. Componentes de hardware

Componente	Especificações e links
<p>TurtleBot 3 Bot</p>  <p>Figura 3 - Robô</p>	<p>Modelo: TurtleBot 3 Burger Raspberry Pi</p> <p>Função principal: Se locomover nos espaços confinados com o acoplamento correto dos sensores periféricos para assim, concluir a inspeção de maneira completa e precisa.</p> <p>Quantidade: 1</p> <p>Onde encontrar: https://www.robotis.us/turtlebot-3-burger-us/</p>

<p>Mini câmera USB</p>  <p>Figura 3 - mini câmera USB</p>	<p>Modelo: DHT11</p> <p>Função principal: fazer a leitura do local para detectar imperfeições e possíveis rachaduras.</p> <p>Quantidade: 1</p> <p>Onde encontrar: https://www.amazon.com.br/OVEH </p>
<p>Sensor de temperatura e umidade</p>  <p>Figura 4 - DHT11</p>	<p>Modelo: DHT11</p> <p>Função principal: fazer a leitura da temperatura e umidade do local.</p> <p>Quantidade: 1</p> <p>Onde encontrar: https://www.baudaeletronica.com.br/produto/sensor-de-umidade-e-temperatura-dht11 </p>

<p>Sensor de gás</p>  <p>Figura 5 - MQ-135</p>	<p>Modelo: MQ-135</p> <p>Função principal: é capaz de detectar gases/vapores de GLP. (obs: verificar na Sprint 5 qual gás será identificado)</p> <p>Quantidade: 1</p> <p>Onde encontrar: https://www.baudaeletronica.com.br/produto/sensor-de-gas-mq-135</p>
<p>Cabo jumper macho-macho</p>  <p>Figura 7 - Protoboard ABS</p>	<p>Modelo: Jumper Macho x Macho</p> <p>Principal função: conectar os componentes e o microcontrolador.</p> <p>Quantidade: 12</p> <p>Onde encontrar: https://produto.mercadolivre.com.br/MLB-1734998537-40-cabos-jumper-20cm-macho-x-macho-para-arduino-protoboard-</p>

Protoboard



Figura 8 - Protoboard ABS

Modelo: Protoboard 170 furos

Material ABS (plástico)

Faixa de Temperatura: -20 a 80°C

Principal função: alocar os jumpers e os sensores.

Quantidade: 1

Onde encontrar:

https://www.piscaled.com.br/protoboard-170-pontos?utm_source=Site&utm_

2.2. Componentes externos

Para automatizar o monitoramento dos espaços confinados, o sistema projetado comunica-se com dispositivos externos através de conexão Wi-Fi e interface gráfica online. Assim, pode-se acessar os dados remotamente através de notebooks.

Para manutenção, é necessário ter posse de IDEs especializados, como o Arduino IDE para o firmware da placa Arduino UNO e um editor de Node.JS, HTML, CSS e Javascript como o VSCode ou WebStorm. Nesse sentido, faz-se necessário baixar ou importar as seguintes bibliotecas:

- Javascript: JQuery e Moment;
- Arduino IDE: EEPROM, Software, SPI, DHT sensor library;

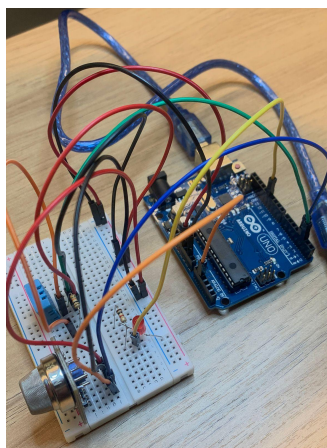
Para visualizar o banco de dados, recomenda-se o uso do software DB Browser ou equivalente com suporte para databases construídas em SQLite. Para testar endpoints, sugerimos o Postman. Para hospedar o servidor e banco de dados, recomendamos um host confiável com o Heroku, AWS ou Hostigator.

3. Guia de Montagem

Nossa montagem terá quatro etapas:

- Montagem do circuito com DHT11;
- Montagem do circuito com o MQ-135;
- Conexão com a câmera e raspberry pi;
- Conexão com os sensores e robô;

(descrever a montagem do circuito a seguir)



3.1 Montagem do circuito com o DHT11

(detalhamento, descrição, desenho do circuito)

3.2 Montagem do circuito com o MQ-135

(detalhamento, descrição, desenho do circuito)

3.3 Conexão com a câmera e o robô

(detalhamento, descrição, desenho do circuito)

3.4 Conexão com sensores e o robô

(detalhamento, descrição, desenho do circuito)

4. Guia de Instalação

4.1 Hardware

Primeiramente, assegure-se de que o circuito foi montado corretamente, seguindo minuciosamente as etapas do guia de montagem disponível na Seção 3. Ao final, o projeto deve se parecer com a imagem abaixo.

Figura .. - Componentes conectados à protoboard

Para auxiliar nessa comparação, verifique a tabela a seguir, que resume as conexões necessárias e certifique-se de que todas estão presentes em seu circuito. Note que as cores dos fios não necessariamente precisam ser as mesmas da tabela, isso foi só uma padronização que implementamos para facilitar a diferenciação de componentes e tipos de ligação

Tabela de conexões

Conexão entre os componentes	Cor dos jumpers
5V do Arduino com + da protoboard	Vermelho (macho / macho)
GND do Arduino com - da protoboard	Preto (macho / macho)

Tabela com descrição de todas as conexões feitas no circuito mostrado na foto anterior

Observações importantes:

- Observe que todos os **negativos** utilizados estão conectados por jumpers de cor **preta**;
- Observe que todos os **positivos** utilizados estão conectados por jumpers de cor **vermelha**;
- Observe quando usamos o **DHT1**, ele está conectado por um jumper de cor **laranja**;
- Observe quando usamos o **MQ135**, ele está conectado por um jumper de cor **azul** da placa;
- Observe quando usamos o **DHT11** ele está conectado por um jumper de cor **verde** da placa;

Ambientação e alimentação elétrica

1º passo:

2º passo:

3º passo:

4º passo:

4.2 Software

Clone do repositório no GitHub

VS Code

Para alterar o código da interface web, recomendamos utilizar o VS Code, uma IDE plenamente equipada com poderosas funções para facilitar o desenvolvimento de websites, tanto de back quanto de frontend. O download pode ser feito [aqui](#).

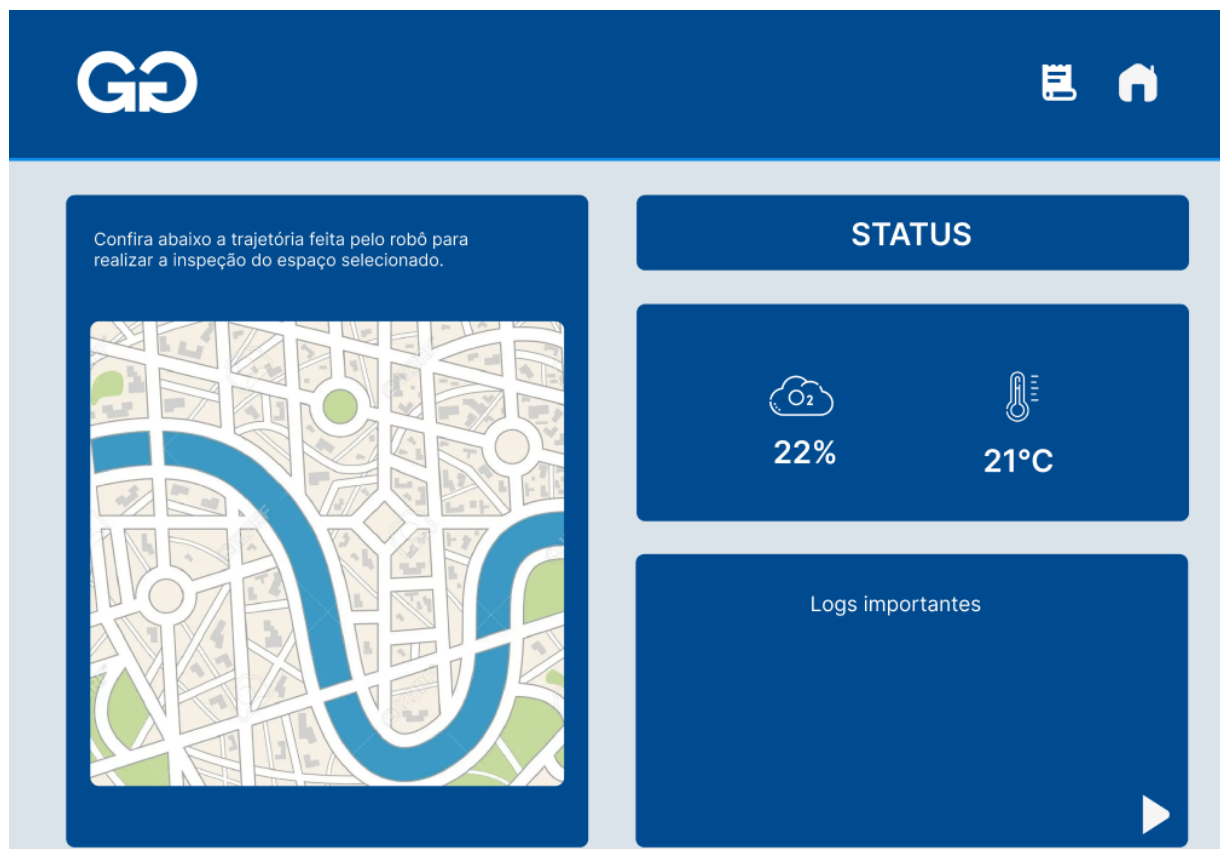
Arduino IDE

Para alterar o código do bloco central, recomendamos utilizar o Arduino IDE, otimizado para desenvolvimento de microcontroladores. Para baixá-lo, siga os passos [deste link](#). Depois, adicione suporte para o Arduino UNO através do gerenciador de placas.

5. Guia de Operação

Consulta de medições atuais

(detalhamento do front)



Consulta de medições históricas

(detalhamento do front)



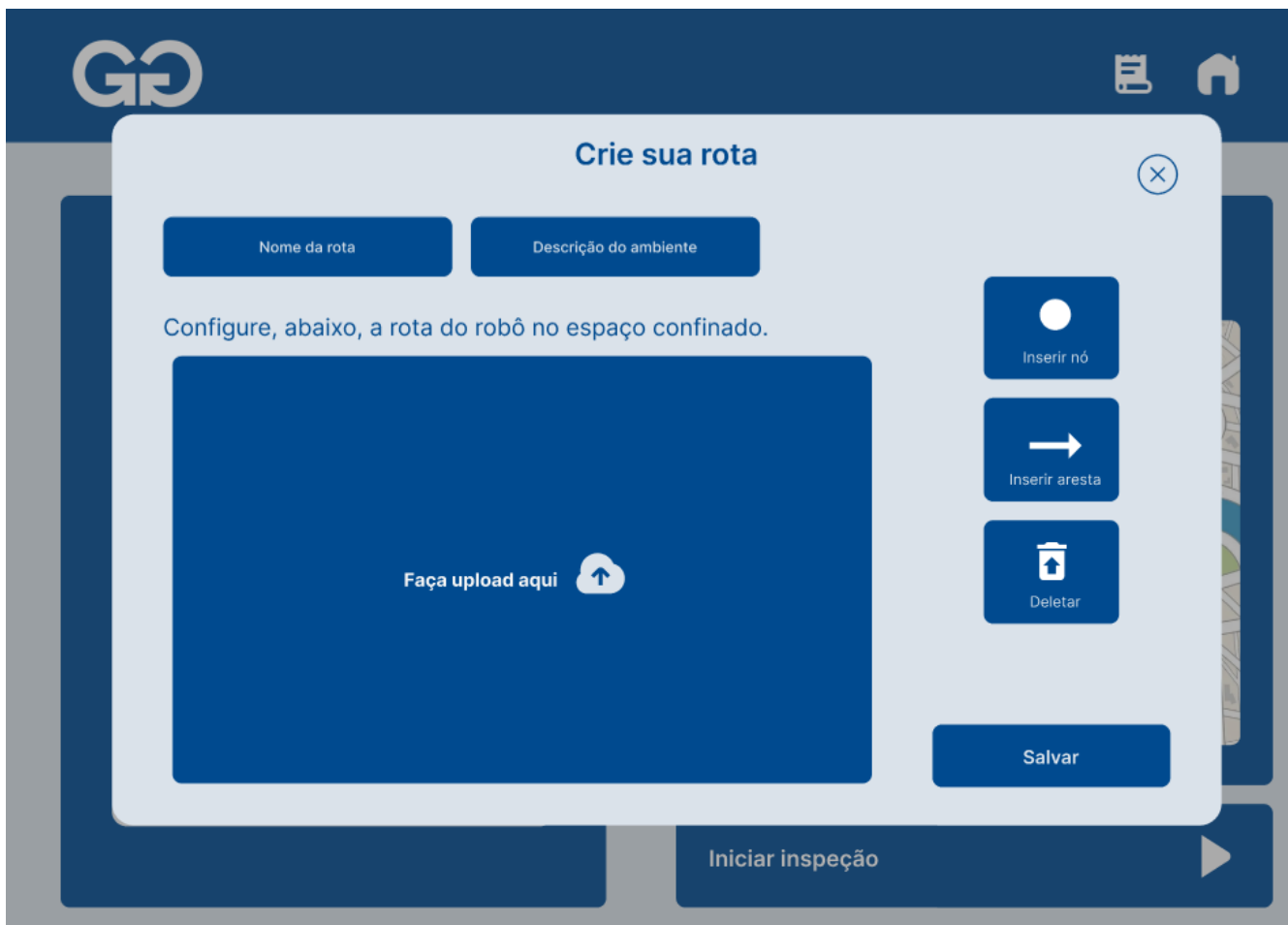
Histórico de inspeções

Pesquise por ID do espaço confinado

Inspeção	ID	Data
Inspeção 01	1KA88J102G	11/12/2023
Inspeção 02	OM9KL2S15	11/12/2023
Inspeção 03	K2WW0PJN3	11/12/2023
Inspeção 04	B6Z11L9M3	11/12/2023

Adicionar uma nota rota

(detalhamento do front)



The screenshot shows a web application interface with a dark blue header. On the left is the 'GO' logo, and on the right are icons for a calendar and a home page. A light blue modal window titled 'Crie sua rota' is centered on the screen. At the top right of the modal is a close button (an 'X' in a circle). Below the title, there are two input fields: 'Nome da rota' and 'Descrição do ambiente'. Below these fields is the instruction 'Configure, abaixo, a rota do robô no espaço confinado.' followed by a large blue rectangular area for route configuration. Inside this area, the text 'Faça upload aqui' is displayed next to a cloud upload icon. To the right of the configuration area are three buttons: 'Inserir nó' (with a white circle icon), 'Inserir aresta' (with a white arrow icon), and 'Deletar' (with a trash can icon). At the bottom right of the modal is a 'Salvar' button. Below the modal, a dark blue bar contains the text 'Iniciar inspeção' and a right-pointing arrow.

6. Troubleshooting

#	Problema	Possível solução
1	A carga da bateria estiver acabando	Antes da atividade ocorrer, é necessário a implementação do carregamento da bateria ao menos com 2 horas de antecedência. Caso essa averiguação for feita durante algum procedimento de inspeção é recomendado que haja uma interrupção das tarefas programadas.
2	LIDAR com a resíduos na frente	Como há a possibilidade de que nos ambientes confinados podem ou não ter algum tipo de resíduo é necessário fazer uma manutenção contínua, antes, durante e após a realização da detecção de objetos.
3	Perder alguma peça no caminho	Embora seja um risco de baixa probabilidade, existe a possibilidade de que alguma peça do robô, como parafusos ou componentes, nesse caso, a recomendação é que seja feita uma averiguação sobre a integridade física do robô antes e depois da vistoria.
4	Perda de conectividade	Esperar a inspeção a terminar por completo para depois verificar os dados obtidos ao final da inspeção.
6		

7		
8		
9		
10		
11		