

RoboCup Brasil TDP

1º Fernanda de Lima
Gonçalves
SESI-SP
Franca, Brasil
fernandalima99995788
@gmail.com

2º José Victor Faccirolli
SESI-SP
Franca, Brasil
jose.faccirolli08@gmail.
com

3º Lívia Silva Comparini
SESI-SP
Franca, Brasil
liviacomparini19@gmail.
.com

4º Valentina Franco de
Camargo Castro
SESI-SP
Franca, Brasil
valentinacfcamargocastro
@gmail.com

Resumo — Este documento apresenta nosso trabalho para a competição RoboCupJunior Rescue Maze, com o objetivo de compartilhar nossas inovações e contribuir para a comunidade global de robótica. Detalhamos os processos de modelagem 3D, montagem, programação, documentação e refinamento dos detalhes do projeto, incluindo testes, erros e validações. Nossas inovações incluem um sistema de navegação baseado em grafos, sensores para identificação das cores do piso, um lançador de kits de resgate utilizando um mecanismo biela-manivela e rodas de silicone. Ao definir as tarefas no núcleo 0 do ESP32 como prioridade 1, aumentamos significativamente a velocidade do programa. Além disso, um segundo núcleo do ESP implementa um sistema de interrupção, garantindo que o robô não perca seu mapa. Por meio deste trabalho, aspiramos a inspirar novos desenvolvimentos e colaborações na área de robótica.

Palavras-chave — Labirinto, RoboCupJunior-Brasil, Robótica-Educacional, Grafos, Edge-Impulse.

I. VISÃO GERAL

Sendo esta a primeira participação de parte da equipe e a terceira de uma das integrantes, reduzimos o tamanho do robô em relação à temporada anterior, incorporando o encaixe interno das rodas para torná-lo ainda mais compacto. A estrutura conta com acrílico transparente, que facilita a visualização dos circuitos, uma placa-mãe modular que concentra os principais componentes, peças 3D em ABS e rodas de silicone reforçadas, garantindo resistência e maior aderência durante o percurso.

A navegação autônoma é realizada em C++ por meio de um sistema baseado em grafos (vértices e arestas), que constrói mapas do ambiente e utiliza o algoritmo BFS para retornar ao início quando necessário. Sensores a laser identificam as paredes do labirinto, fornecendo dados para a tomada de decisão.

Na detecção de vítimas, utilizamos duas câmeras OpenMV H7 Plus. As letras são classificadas com um modelo MobileNetV2 treinado no Edge Impulse, empregando camadas como Separable Convolution, Linear Bottleneck e Softmax, o que garante alta precisão e baixo custo computacional. Já para vítimas coloridas, a análise ocorre no espaço LAB, seguida de verificação de forma e área. As informações são transmitidas à ESP32 via I2C, resultando em mais de 80% de confiança nos testes.

Os avanços implementados proporcionaram maior estabilidade e velocidade, permitindo que o robô navegue com eficiência no labirinto e alcance desempenho superior em relação à versão anterior.

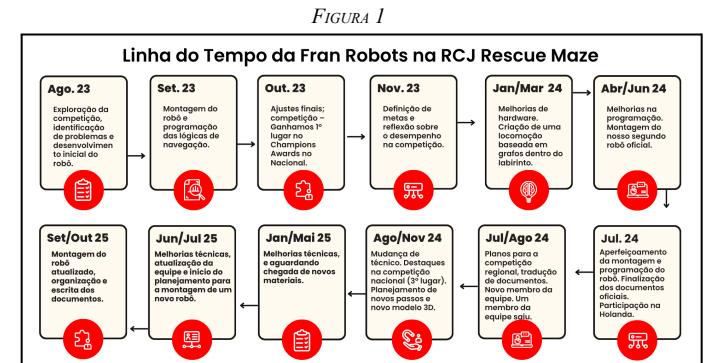
II. PLANEJAMENTO GERAL DO PROJETO

O principal objetivo da nossa equipe nesta temporada é alcançar maior robustez em comparação à anterior, buscando tornar o robô mais confiável. Para isso, estabelecemos metas

como remodelar sua estrutura, reduzindo o tamanho para facilitar a locomoção em becos estreitos, além de adotar novos motores que contribuam para maior velocidade e eficiência. Também pretendemos garantir que o design seja adaptável para futuras fases, mantendo a consistência do desempenho.

Como parte do nosso planejamento, desenvolvemos uma linha do tempo (Figura 1) que registra a evolução do robô e as principais etapas do projeto ao longo da temporada. Essa linha do tempo nos permite visualizar de forma clara os avanços técnicos, as melhorias implementadas e os momentos de aprendizado da equipe, servindo como uma ferramenta de acompanhamento e reflexão sobre nosso progresso.

Paralelamente, temos como meta desenvolver o conhecimento dos novos integrantes, promovendo aprendizado e integração na equipe.



III. ESTRUTURA DO ROBÔ E ELETRÔNICA

A estrutura do robô foi desenvolvida para ser mais compacta e otimizada em comparação à versão anterior, aproveitando melhor o espaço interno e facilitando a manutenção. Ela é composta por placas de acrílico transparente, que permitem a visualização dos circuitos e componentes, e acrílico preto na base, utilizado para evitar reflexos que poderiam interferir na placa de refletância. O design mecânico foi elaborado no SolidWorks, com simulação 3D realizada antes da fabricação. Todas as peças foram produzidas internamente, utilizando corte a laser para os componentes 2D e impressão 3D em filamento ABS para as partes estruturais, garantindo durabilidade e precisão nos encaixes. A placa-mãe, localizada na parte superior do robô, concentra os principais módulos eletrônicos, contribuindo para uma organização prática e eficiente do sistema.

A. Mecânica e Manufatura

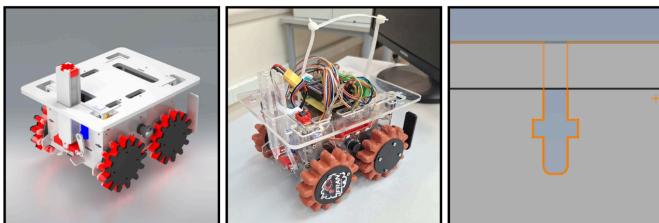
1. Estrutura Principal

O robô foi projetado no SolidWorks (Figura 2a, 2b), priorizando robustez, modularidade e dimensões reduzidas.

Sua estrutura possibilita montagem e desmontagem ágil por meio de junções em “T” (Figura 2c), otimizando a manutenção. Foram distribuídos oito sensores laterais para detecção de paredes e duas câmeras, posicionadas nas laterais, para identificação de vítimas, assegurando ampla cobertura do ambiente e maior confiabilidade operacional.

FIGURA 2

(A) MODELO 3D DO ROBÔ; (B) ROBÔ REAL; (C) JUNÇÃO EM T.



2. Rodas, Motores e Encoder

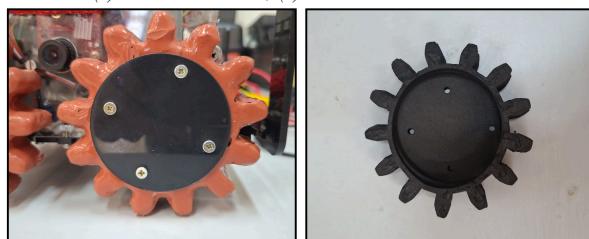
O robô é movimentado por quatro motores equipados com encoders. Atualmente, apenas um encoder é utilizado para odometria, enquanto os demais permanecem conectados e disponíveis para ativação futura, se necessário. Diferentes tipos de rodas foram testados durante o desenvolvimento, sendo adotada uma roda personalizada que demonstrou maior eficiência e confiabilidade durante os testes.

3. Rodas de Silicone

As rodas do robô (Figura 3a) foram desenvolvidas para proporcionar alta aderência, estabilidade e confiabilidade durante a navegação pelo labirinto. Cada roda é composta por um núcleo de EVA (Figura 3b), responsável por absorver impactos e manter a leveza, coberto por uma camada externa de silicone que garante atrito ideal em diferentes superfícies. O encaixe do motor é impresso em 3D, permitindo integração precisa com o eixo de tração, e a fixação da roda é feita por meio de parafusos e porcas, facilitando manutenção e substituição rápida. O design modular das rodas possibilita ajustes no alinhamento e na tensão, além de proteger o disparador de kits durante deslocamentos em rampas e superfícies irregulares. Essa configuração demonstra eficiência e confiabilidade superiores, mantendo o robô estável e preciso durante toda a competição.

FIGURA 3

(A) RODA COMPLETA; (B) NÚCLEO DE EVA DA RODA.

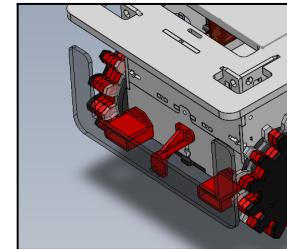


4. Para-choque tátil

O robô é equipado com um para-choque tátil (Figura 4) composto por duas unidades de micro-switches, substituindo as peças frontais passivas utilizadas anteriormente. O corpo do para-choque é confeccionado em acrílico, enquanto os suportes dos micro-switches são impressos em ABS, garantindo durabilidade e resistência. Este dispositivo não

apenas previne tombos frontais devido ao peso dos motores, mas também detecta obstáculos e direciona o robô até que não haja mais colisões, contribuindo para maior precisão na navegação. O para-choque é fixo, mas pode ser removido facilmente quando necessário, facilitando manutenção e ajustes.

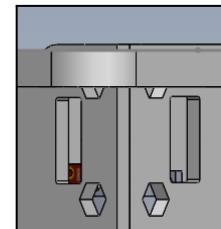
FIGURA 4
PARA-CHOQUE TÁTIL



5. Módulos

O robô possui modularidade em oito sensores de distância, duas câmeras laterais e nas partes superior, inferior e frontal inferior do chassis. Componentes da placa-mãe, como multiplexador, ESP32 e GY-BNO055, podem ser removidos facilmente via conectores de barra-pinada, enquanto os sensores a laser são fixados por parafusos. Essa configuração permite manutenção e substituição rápidas, sem desmontagem completa do robô.

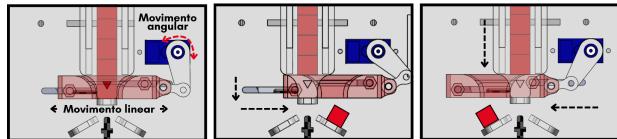
FIGURA 5
SUPORTE DOS SENSORES DE DISTÂNCIA.



6. Mecanismo de Liberação dos Kits de Resgate

Melhoramos o mecanismo de disparo de kits de resgate do nosso robô, passando de um design unidirecional para um bidirecional, aumentando a precisão e eficiência. O mecanismo de biela-manivela (inspirado em uma ideia anterior de disparo por mola), é acionado por um servo motor que desliza, por meio de um êmbolo, suavemente os kits para ambos os lados sem que haja a necessidade de curvas do robô (Figura 6). Sua posição na traseira do robô evita que as rodas atropelam os kits. Possui paredes laterais limitantes que impedem os kits de baterem nas irregularidades da roda e terem seu escape alterado, também foram adicionados abre-longo na manivela e biela do mecanismo que diminuem que travamentos ocasionais. Diversas versões foram criadas e registradas para aumentar a confiabilidade e precisão, prevenindo que os kits travem ou fiquem fora da área pontuável. Alguns registros estão na seção de mecânica do nosso GitHub.

FIGURA 6
SEQUÊNCIA DE OPERAÇÃO DO MECANISMO DE BIELA MANIVELA.



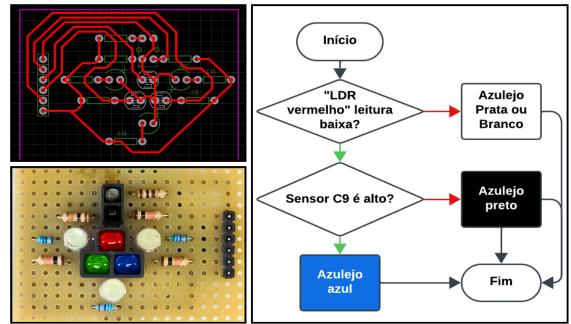
7. Materiais

- Placas de acrílico:** O acrílico transparente proporciona visibilidade dos circuitos internos, auxiliando em diagnósticos, enquanto o acrílico preto, na parte inferior da tampa do chassi, previne a interferência nos sensores de cor da placa de refletância causada pela iluminação interna.
- Filamento ABS:** O filamento de ABS possui baixa tolerância durante a impressão, durabilidade e resistência. Suas propriedades bem conhecidas facilitaram a confecção 3D e a simulação com CAE.

B. Design Eletrônico e Manufatura

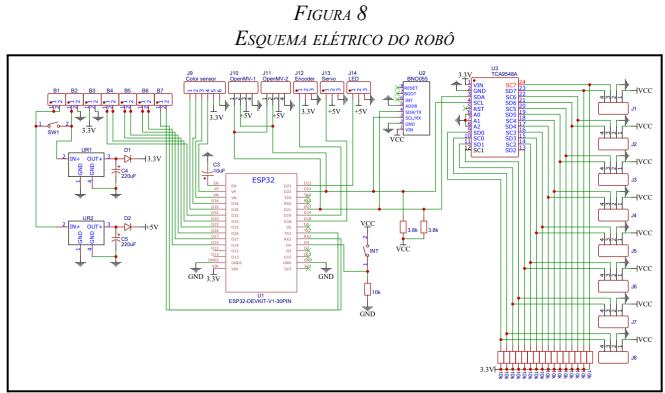
- I2C:** O robô utiliza oito sensores de distância infravermelhos VL53L0X conectados via I2C, integrados a um multiplexador PCA9548A de oito canais que consolida os sensores e o giroscópio GY-BNO055 em um único barramento, prevenindo interferências. O GY-BNO055 atua tanto como giroscópio quanto para rastrear a inclinação do robô, garantindo medições precisas de posição e orientação.
 - Câmeras:** Duas câmeras OpenMV H7 Plus, com processamento integrado, são utilizadas para identificação de vítimas. Elas são fixadas lateralmente em suportes de ABS, mantendo-se dentro da altura máxima de 7 cm exigida pelas regras. Os resultados do processamento são transmitidos via I2C para a ESP32, permitindo fácil substituição e manutenção.
 - Placa de refletância:** Uma placa de circuito removível (Figura 7a) feita em placa ilhada, é posicionada no nível mais baixo do robô, agora, internamente. Ela contém um sensor emissor e receptor de infravermelho (C9L3), LEDs brancos de alta luminosidade de 5mm e LDRs com filtros de gelatina para cores vermelha, verde e azul. O sensor infravermelho detecta rapidamente a luminosidade da superfície e os filtros de gelatina atuam como barreiras ópticas RGB, simplificando a separação de cores no software, como mostrado no fluxograma (Figura 7b).
- Fisicamente, a luz consegue passar apenas pelo filtro de seu espectro luminoso, assim, quando deparado com uma superfície inteiramente azul a reflexão da luz branca dos LEDs adquiri essa característica e chega somente ao LDR coberto por esse filtro, gerando uma leitura de nível alto, enquanto os outros se mantêm com uma leitura de nível baixo. Com os três espectros RGB e uma leitura analógica, estamos preparados para cores conhecidas e potenciais surpresas do desafio Super Teams e Técnico. Sua instalação interna agora permite o robô passar diretamente por cima de redutores paralelos às laterais do mesmo, solucionando problemáticas enfrentadas na fase mundial da última temporada.

FIGURA 7
(A) ROTEAMENTO DA PLACA DE REFLETÂNCIA E SUA FOTO; (B) FLUXOGRAMA DA LÓGICA DE IDENTIFICAÇÃO DE CORES.



1. EasyEDA

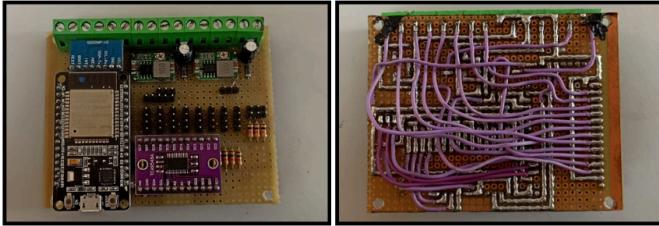
Para criar esquemáticos, diagramas elétricos e rastreamento de circuitos, usamos o software de design eletrônico gratuito EasyEDA. Ele previniu erros de produção e economizou tempo e recursos. Conforme necessário, os circuitos foram modificados fisicamente e depois atualizados online, mantendo a documentação atualizada.(Figura 8)



2. Controladores

- ESP32:** O núcleo do robô, um microcontrolador ESP32, gerencia movimentos e decisões, processando dados dos sensores para distância, cor, orientação e classificação de vítimas. Foi escolhida por sua capacidade de multitarefa, velocidade, memória e tamanho compacto, quando comparado com um Arduino. Um capacitor eletrolítico de baixa capacitância entre os pinos "EN" e "GND" permite a entrada automática no modo de programação do chip.
- OpenMV H7 Plus:** As câmeras OpenMV H7 Plus foram selecionadas por possuírem processamento integrado, baixo consumo de energia, facilidade de integração via I2C, dimensões compactas e confiabilidade. Elas realizam o processamento das imagens para identificação das vítimas e enviam os resultados diretamente à ESP32, que executa a lógica de resgate.
- Placa mãe:** A placa central (Figura 9) agrupa os principais componentes, permitindo a substituição eficiente do "cérebro" do robô. A fiação interna e as trilhas de solda facilitam as conexões com bornes e conectores posicionados estratégicamente, minimizando os comprimentos dos fios e garantindo substituições rápidas em caso de falhas ou defeitos.

FIGURA 9
FRENTE E VERSO DA PLACA MÃE



3. Sistemas de Alimentação

- Alimentação:** O robô é alimentado por uma única bateria LiPo de 11.1V 2,2 Ah. Dois reguladores Step-Down fornecem 5 V e 3,3 V, atendendo aos diferentes componentes eletrônicos e sensores do robô. Ambos os reguladores possuem capacitores de $220\ \mu\text{F}$ em suas saídas, funcionando como filtros passa-baixa para reduzir ruídos e garantir estabilidade na alimentação. Um display voltímetro exibe os níveis da bateria, fornecendo informações apenas de forma informativa para evitar uso excessivo durante os treinos.
- Motores e Drivers:** O sistema de propulsão é composto por quatro motores TEK8 de 12 V e 60 RPM, com encoder integrado e caixa de redução. Cada dois motores são controlados por uma ponte H L298N, garantindo navegação eficiente e confiável. As travas físicas integradas aos motores evitam rotações indesejadas durante o desligamento, aumentando a precisão em diferentes condições de movimento. A tensão nominal de 12 V permite que os motores alcancem a velocidade projetada, enquanto a configuração elétrica optimiza a eficiência e preserva a vida útil dos componentes.

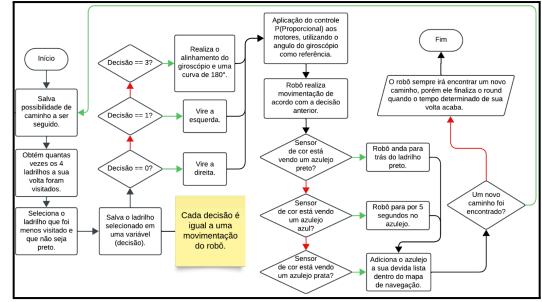
IV. SOFTWARE

O robô utiliza C++ para navegação e MicroPython nas câmeras OpenMV H7 Plus, combinando controle de movimento com identificação de vítimas. Alfanuméricas são reconhecidas por MobileNetV2, enquanto vítimas coloridas usam processamento de imagens em LAB. Os dados são enviados à ESP32 via I2C, com confiança superior a 80% na maioria dos casos (Figura 11).

A. Navegação no Labirinto

O mapa é atualizado a cada novo ladrilho visitado. Isso permite que o robô tome decisões ao escolher o próximo caminho, priorizando áreas não exploradas. Também focamos em movimentos precisos, enfatizando alinhamentos, cálculos de movimento e controles PD. A lógica pode ser dividida em duas etapas, representadas pelos fluxogramas na Figura 10.

FIGURA 10
ANALISE DOS LADRILHOS DE COR; PROCESSO DE DECISÃO; TOMADA DE ATITUDE.



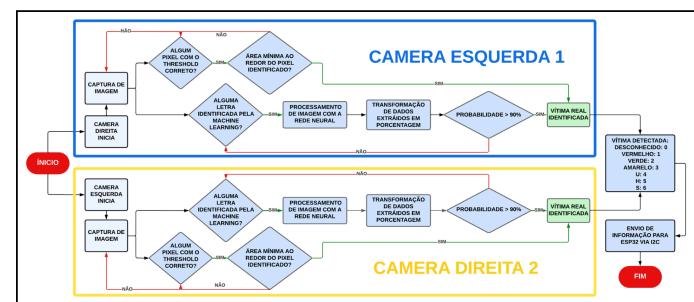
B. Identificação de Imagem

Duas câmeras OpenMV H7 Plus capturam imagens para identificação de vítimas. Letras são classificadas por um modelo MobileNetV2 treinado no Edge Impulse. As imagens são redimensionadas para 96×96 pixels, convertidas para tons de cinza e processadas pela rede neural, que aplica:

- Separable Convolution:** Detecta padrões como bordas e curvas.
- Linear Bottleneck:** Extrai características complexas mantendo baixo custo computacional.
- Pooling:** Reduz risco de overfitting e otimiza recursos.
- Flatten:** Converte os dados em um vetor unidimensional.
- Dense:** Calcula médias ponderadas das características para classificação.
- Softmax:** Realiza a decisão final, atribuindo probabilidades a cada classe.

Para vítimas coloridas, é utilizada limiarização no espaço LAB, seguida de verificação de forma e área para confirmar regiões válidas. Todos os dados de detecção, alfanuméricos e coloridos, são enviados à ESP32 via I2C, com confiança superior a 80% na maioria dos casos (Figura 11).

FIGURA 11
FLUXOGRAMA DO ALGORITMO GERAL DE PROCESSAMENTO DE IMAGEM.



V. Inovações

A. Inovações Mecânicas e Elétricas:

- Roda de Silicone:** O robô utiliza rodas compostas de EVA com uma camada externa de silicone, proporcionando maior aderência em rampas e curvas, garantindo deslocamento mais estável e eficiente.
- Formato Chanfrado do Último Andar:** Os chanfros na extremidade traseira evitam que o robô colida com as paredes especialmente durante curvas sobre seu eixo. Solucionando muitos problemas que ocasionavam em desalinhamentos.
- Biela-manivela no Disparador de Kits de Resgate:** Para

dispensar os kits de forma bidirecional usando apenas um Servo motor, um mecanismo de biela-manivela aciona um êmbolo que direciona os kits para a direita ou esquerda. Abre-longos na estrutura mantêm o êmbolo sempre alinhado, enquanto um na manivela garante um movimento mais suave e evita travamentos, solucionando problemas das versões anteriores.

- **Kit de Resgate Antiquique:** O kit de resgate preenchido com bolinhas de estanho para solda solucionou nosso problema em que os mesmos quicavam para fora da área pontuável de 15 centímetros.
- **Para-choque Tátil:** O para-choque sensível ao toque manteve sua função de proteger as esteiras durante os alinhamentos e, além disso, detectam obstáculos no percurso.
- **Sensores Modulares:** Os sensores de distância e câmeras não são soldados diretamente. Em vez disso, são conectados por barras pinadas, permitindo substituições rápidas e fáceis. A posição e o alinhamento são garantidos por parafusos e porcas previamente fixadas, o que facilita ainda mais a troca desses componentes.
- **Placa de Refletância com Filtros de Cor:** Para identificar os ladrilhos coloridos, utilizamos uma placa de refletância desenvolvida internamente, equipada com sensores LDR encapsulados e cobertos por filtros de gelatina luminosa nas três cores do espectro RGB. Esses filtros permitem a passagem apenas das cores correspondentes ao seu espectro. No código, quando um dos sensores registra um valor alto, os outros registram valores baixos, utilizando um circuito de pulldown.
- **Placa Mãe com Componentes Modulares:** Para identificar os ladrilhos coloridos, utilizamos uma placa de refletância desenvolvida internamente, equipada com sensores LDR encapsulados e cobertos por filtros de gelatina luminosa nas três cores do espectro RGB. Esses filtros permitem a passagem apenas das cores correspondentes ao seu espectro. No código, quando um dos sensores registra um valor alto, os outros registram valores baixos, utilizando um circuito de pulldown.

B. Inovações na Programação:

- **Processamento Eficiente de Rede Neural:** Redimensionamento, conversão para tons de cinza e execução das camadas da rede neural (Separable Convolution, Linear Bottleneck, Pooling, Flatten, Dense, Softmax) de forma otimizada.
- **BFS:** É o algoritmo de busca no qual aplicamos ao mapeamento que nos retorna os ladrilhos identificados, e qual direção foi menos explorada, juntamente com a ordem de descoberta dos grafos.
- **Biblioteca de Navegação no Labirinto:** A biblioteca "Maze_FranRobots.h" foi desenvolvida por nós para auxiliar no controle de variáveis, rotas e mapeamento durante a navegação, além de implementar uma gama de funcionalidades essenciais.
- **Gerenciamento de Conexões de Nós:** A classe *Node Connections* gerencia as conexões entre diferentes nós no labirinto, armazenando até quatro conexões por nó e fornecendo métodos para defini-las e imprimi-las.

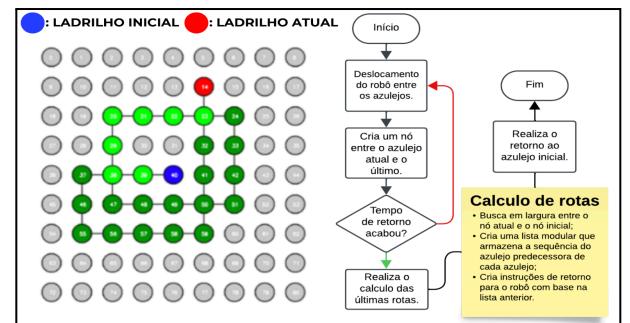
• **Gerenciamento de Listas:** A classe *List* é uma implementação personalizada de listas dinâmicas, semelhante ao que a biblioteca NumPy faz em Python. Ela permite alocação dinâmica de memória, redimensionamento, inserção, remoção e acesso a elementos, entre outras funcionalidades.

• **Estrutura de Ponto:** Define coordenadas bidimensionais dentro do labirinto, com métodos para inicialização e acesso às posições.

• **Algoritmos de Navegação e Busca:** A classe "MazeRunner" é crucial para a navegação do robô. Ela mantém o estado atual do robô, posição, direção e nós visitados. Além disso, implementa algoritmos de busca, como BF, para encontrar os caminhos mínimos entre nós e gerar instruções de movimento.

Esta biblioteca utiliza alocações dinâmicas de memória, que são essenciais para gerenciar grandes conjuntos de dados, como nós e conexões visitadas no labirinto (Figura 12). Ela oferece uma solução escalável para o controle do robô, com foco em navegação autônoma, utilizando listas inovadoras e algoritmos de busca robustos para maior eficiência.

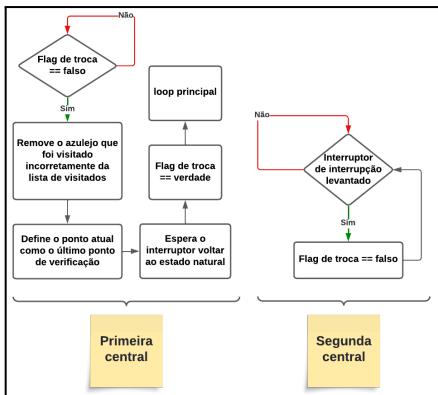
FIGURA 12
À ESQUERDA ESTÁ O GRÁFICO DO LABIRINTO: VERDE ESCURO PARA OS VISITADOS, VERDE CLARO PARA OS CAMINHOS MAIS CURTOS. À DIREITA, ESTÁ O FLUXOGRAMA DE NAVEGAÇÃO DO ROBÔ.



• **Task Direto no Núcleo (Core 0):** Uma otimização de desempenho foi alcançada ao mover a execução do código no ESP32 do "void loop" para uma tarefa de prioridade 1 no núcleo de processamento. Essa mudança interrompeu a de-priorização causada pelo sistema operacional, resultando em uma execução mais rápida.

• **Botão de Interrupção Multitarefa:** O segundo núcleo do nosso microcontrolador lê um interruptor físico que congela o robô durante falhas de progresso, eliminando a necessidade de desligamentos e perda do mapa. Ele atualiza uma variável para 'false' quando ativado, solicitando que o código principal pule tarefas até que o interruptor seja desativado (Figura 13). Durante essa pausa, variáveis são ajustadas para refletir a posição atual do robô e a direção cardinal com base no último ponto de verificação.

FIGURA 13
INFOGRÁFICO DO PROCESSO DE INTERRUPÇÃO.



VI. AVALIAÇÃO DE PERFORMANCE

Durante a etapa regional interna da escola, identificamos problemas relacionados ao tamanho do robô, que frequentemente ficava preso em locais estreitos da pista devido às dimensões quase equivalentes às da arena. Essa limitação motivou ajustes no chassi, placas e posicionamento dos componentes, resultando em maior manobrabilidade e eficiência.

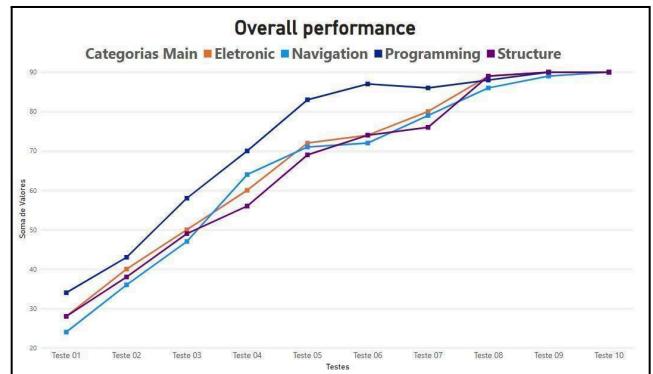
Em relação à comunicação I2C, todos os desafios prévios foram solucionados com o uso de multiplexadores. No entanto, durante o Mundial de Salvador, a ESP32 reiniciava ocasionalmente devido a divisões por zero em algumas operações matemáticas. Esse problema foi resolvido implementando uma guard que previne a execução dessas operações, garantindo estabilidade durante a competição.

Outros desafios incluíram a definição do tamanho correto do eixo da biela do disparador de kits. Após diversos testes e pesquisas, conseguimos determinar dimensões ideais, garantindo funcionamento confiável e alinhamento adequado.

No treinamento do reconhecimento de vítimas, foram tomados cuidados para evitar overfeeding do modelo de machine learning. Inicialmente, surgiram identificações falsas, resolvidas com a criação de um bucket “Unknown” para armazenar regiões suspeitas, evitando erros na tomada de decisão do robô.

Todo o processo de desenvolvimento foi documentado por vídeos e atualizações constantes no grupo de comunicação da equipe, permitindo acompanhamento detalhado das soluções implementadas. Além disso, construímos uma tabela no Power BI (Figura 14) para registrar e analisar a evolução do robô, incluindo modificações mecânicas, eletrônicas, de software e estratégias de navegação, facilitando a visualização do progresso e a tomada de decisões fundamentadas para futuras melhorias.

FIGURA 14
TABELA DE EVOLUÇÃO NO POWER BI



CONCLUSÃO

Com base nas experiências desta temporada, nosso robô evoluiu significativamente em design mecânico, eletrônico e software, apresentando maior confiabilidade, precisão e eficiência. Ajustes no tamanho e na modularidade dos componentes, bem como o uso de rodas de silicone, melhoraram a manobrabilidade e a estabilidade em diversos terrenos do labirinto. No software, a integração de C++ e MicroPython, aliada a técnicas de machine learning e processamento de imagens em tempo real, possibilitou o reconhecimento de vítimas com alta confiabilidade, enquanto estratégias de documentação e testes sistemáticos permitiram superar desafios como reinicializações da ESP32 e identificações falsas.

Em resumo, esta participação permitiu à equipe aprimorar habilidades críticas em modelagem 3D, eletrônica, navegação, reconhecimento de imagens e documentação, garantindo aprendizado contínuo e preparando o robô para futuras competições. Com esta iniciativa, nosso objetivo é ter um impacto positivo na comunidade de robótica e avançar em nossa jornada profissional ao abraçar experiências e aprender com os erros. A ciência nunca para, e nós também não.

REFERÊNCIAS

- [1] Ambiente de Desenvolvimento Integrado (IDE) do Arduino v1. Arduino Docs. <https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics>. Atualizado em 17 de janeiro de 2024. Acessado em 20 de junho de 2025.
- [2] BasuMallick, Chiradeep. O que é CAD (Desenho Assistido por Computador)? Definição, Tipos e Aplicações. Spiceworks. <https://www.spiceworks.com/tech/devops/articles/what-is-cad>. Atualizado em 2022. Acessado em 20 de junho de 2025.
- [3] Busca em Largura (Breadth First Search – BFS) para um Grafo. GeeksforGeeks. <https://www.geeksforgeeks.org/breadth-first-search-or-bfs-for-a-graph>. Atualizado em 21 de abril de 2025. Acessado em 21 de junho de 2025.
- [4] Campbell, Scott. Fundamentos do Protocolo de Comunicação I2C. Circuit Basics. <https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol>. Acessado em 21 de junho de 2025.
- [5] Definição do Espaço de Cores Lab. ScienceDirect. <https://www.sciencedirect.com/topics/agricultural-and-biologi>

- cal-sciences/lab-color-space. Acessado em 21 de junho de 2025.
- [6] Definição de MobileNetV2. ScienceDirect. <https://www.sciencedirect.com/topics/computer-science/mobilenetv2>. Acessado em 21 de junho de 2025.
- [7] Documentação do MicroPython. Docs MicroPython. <https://docs.micropython.org/en/latest>. Acessado em 20 de junho de 2025.
- [8] ESP32. Espressif. <https://www.espressif.com/en/products/socs/esp32>. Acessado em 20 de junho de 2025.
- [9] GPIO (Entrada/Saída de Uso Geral) – Definição. TEM Electronic Components. <https://www.tme.eu/en/news/library-articles/page/61888/gpio-general-purpose-input-output-definition>. Acessado em 21 de junho de 2025.
- [10] Introdução ao PID. FIRST Robotics Competition. <https://docs.wpilib.org/pt/stable/docs/software/advanced-controls/introduction/introduction-to-pid.html>. Acessado em 21 de junho de 2025.
- [11] Joshi, Vaidehi. Uma Introdução Suave à Teoria dos Grafos. Medium. <https://medium.com/basecs/a-gentleintroduction-to-graph-theory-77969829ead8>. Publicado em 2017. Acessado em 20 de junho de 2025.
- [12] O Que é C++? (E Como Aprender). <https://www.coursera.org/articles/what-is-c-plus-plus>. Atualizado em 17 de janeiro de 2025. Acessado em 21 de junho de 2025.
- [13] O Que é Power BI?. Microsoft. <https://learn.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview>. Atualizado em 22 de março de 2024. Acessado em 21 de junho de 2025.
- [14] Pontes H – Os Fundamentos. Modular Circuits. <https://www.modularcircuits.com/blog/articles/h-bridgessecrets/h-bridges-the-basics>. Acessado em 21 de junho de 2025.
- [15] Quem Somos. Bitrix24. <https://www.bitrix24.com.br/about>. Acessado em 20 de junho de 2025.
- [16] Raspberry Pi – Sobre Nós. Raspberry Pi. <https://www.raspberrypi.com/about>. Acessado em 21 de junho de 2025.
- [17] SOLIDWORKS – Software de Design 3D CAD. SOLIDWORKS. <https://www.3ds.com/products/solidworks>. Acessado em 20 de junho de 2025.
- [18] Sobre o Canva. Canva. https://www.canva.com/pt_br/about. Acessado em 20 de junho de 2025.
- [19] Sobre o EasyEDA. EasyEDA. <https://easyeda.com/page/about>. Acessado em 20 de junho de 2025.
- [20] Sobre a FIRST LEGO League. FIRST LEGO League. <https://www.firstlegoleague.org/about>. Acessado em 20 de junho de 2025.
- [21] Sobre o MicroPython. MicroPython. <https://micropython.org/>. Acessado em 21 de junho de 2025.
- [22] Sobre o NumPy. NumPy. <https://numpy.org/about>. Acessado em 21 de junho de 2025.
- [23] Sobre o WhatsApp. WhatsApp. <https://www.whatsapp.com/about?lang=EN>. Acessado em 20 de junho de 2025.
- [24] Sobre o Edge Impulse. Edge Impulse. <https://edgeimpulse.com/about>. Acessado em 21 de junho de 2025.
- [25] Sobre Nós. Lego. <https://www.lego.com/en-us/aboutus>. Acessado em 20 de junho de 2025.
- [26] Visão Geral | Sensor de Orientação Absoluta Adafruit BNO055. Adafruit. <https://learn.adafruit.com/adafruit-9-dof-orientation-imu-fusion-breakout-bno055>. Acessado em 21 de junho de 2025.
- [27] Visão Geral do OpenMV IDE. Documentação MicroPython. https://docs.openmv.io/openmvcam/tutorial/openmvide_overview.html. Acessado em 20 de junho de 2025.
- [28] V. Ryan. O Mecanismo de Manivela e Deslizante. Technology Student. <https://technologystudent.com/cams/crkslid1.htm>. Acessado em 20 de junho de 2025

