

Numa partida, os robôs iniciam em qualquer posição arbitrária dentro das vias (área branca do mapa), necessitando se localizar para conseguir completar as tarefas. Na plataforma de embarque estarão dispostos tanto passageiros

que têm intenção de serem levados a algum destino quanto passageiros que não tem, e portanto não devem ser retirados de lá pelo robô. Além disso, o robô enfrentará complicações no trajeto, como lombadas que podem ser posicionadas em diversas posições de forma aleatória nas vias. Outro fator é que obstáculos podem ser colocados mesmo durante uma partida, simulando acidentes de trânsito ou obstrução de vias, e os carros devem ser inteligentes o suficiente para recalculando suas rotas e obter sucesso ainda assim.

### III. FERRAMENTAS

Esta seção descreve o conjunto de ferramentas utilizadas pela equipe RoboForge durante a execução deste projeto, principalmente tratando-se de *softwares* e serviços fundamentais para a execução técnica do desenvolvimento, mas também envolvendo aspectos relacionados à gestão, comunicação e planejamento. Tais aspectos envolvem, mas não se limitam a: montagem do robô, desenvolvimento do código, auxílio na tomada de decisões, comunicação e gestão de pessoas, etc. São elas:

#### A. Git/Github

Ferramenta de versionamento. Adaptaremos ideias do modelo Git Workflow para realizar os trabalhos de maneira sólida. Neste modelo, contamos com várias *branches* (ramificações, ou “versões alternativas correntes”) onde podemos bifurcar e gerar novas implementações diferentes a partir de uma mesma base de código, sem abrir mão de um controle de versões eficiente. Tais ferramentas e convenções já se mostraram muito úteis em competições anteriores, para facilitar lidar com diferentes alternativas de código, inclusive distribuídas no tempo, sem o risco de que novas alterações comprometessem o que já havia sido consolidado.

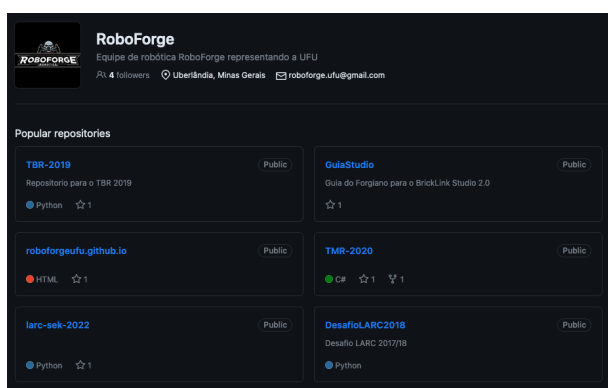


Fig. 2. Página da RoboForge no GitHub.

#### B. Bricklink Studio 2.0

*Software* utilizado no processo de design e construção da estrutura dos robôs. Nele é possível montar o robô em questão virtualmente com peças dos kits LEGO e analisar diversos fatores, como a força exercida em cada peça, distribuição de peso do robô, etc. Tais recursos são valiosos pois auxiliam em tomadas de decisão pela equipe responsável

pela engenharia, simulação e prototipação de montagens, e promovem a garantia de uma estrutura sólida e resistente.

Além disso, essa ferramenta é capaz de gerar manuais de montagem automaticamente, o que se mostra muito útil para catálogo de diferentes robôs, sem a necessidade de estarem todos montados ao mesmo tempo.

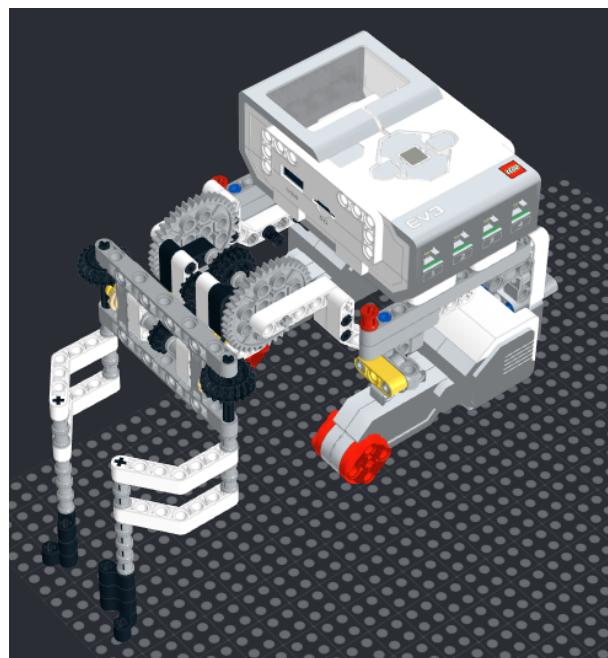


Fig. 3. Exemplo de robô construído no Bricklink Studio 2.0, com as peças do kit LEGO EV3.

#### C. Visual Studio Code e EV3 Micropython

A programação será desenvolvida utilizando a linguagem *Python*, que não é nativa ao controlador oficial dos kits LEGO utilizados pela equipe, o EV3. Para isso, utilizaremos o editor de código-fonte *Visual Studio Code* juntamente com a extensão do mesmo chamada de *EV3 MicroPython*, desenvolvida pela própria LEGO, que facilita o gerenciamento de conexões entre o computador e o controlador, assim como a transferência de arquivos entre os mesmos.

Porém, o VS Code se trata apenas de um editor de texto. Também é necessário instalar no controlador, através de um cartão SD, o sistema operacional alternativo baseado em linux, *ev3dev*, também disponibilizado pela LEGO. Através do *ev3dev* é possível escrever código para o controlador em diferentes linguagens, incluindo o *MicroPython*, uma implementação da linguagem *Python 3* otimizada para execução em microcontroladores e ambientes computacionais com disponibilidade restrita de recursos (energia, memória, CPU, etc). A biblioteca de funcionalidades para interação do código *MicroPython* com os recursos de hardware do robô é chamada *pybricks*, e traz um grande conjunto de funcionalidades para controle dos diversos sensores e motores presentes no kit.

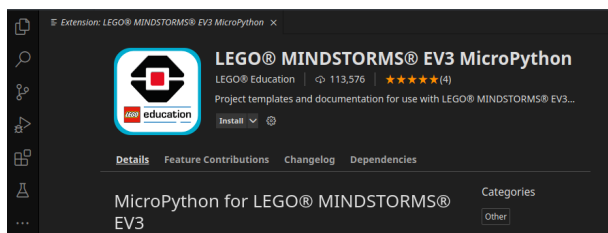


Fig. 4. Extensão EV3 MicroPython no Visual Studio Code.

#### D. Excalidraw

Ferramenta online para criação de fluxogramas, diagramas e outros recursos visuais de forma colaborativa. Será utilizada para criar fluxogramas dos processos de desenvolvimento, criar diagramas das estruturas dos robôs e outros recursos visuais que auxiliem na comunicação e no planejamento do projeto.

O *Excalidraw* oferece recursos de colaboração em tempo real que permitem que toda a equipe trabalhe nos mesmos diagramas simultaneamente. Isso facilita a comunicação e o compartilhamento de ideias, além de garantir que todos estejam na mesma página.

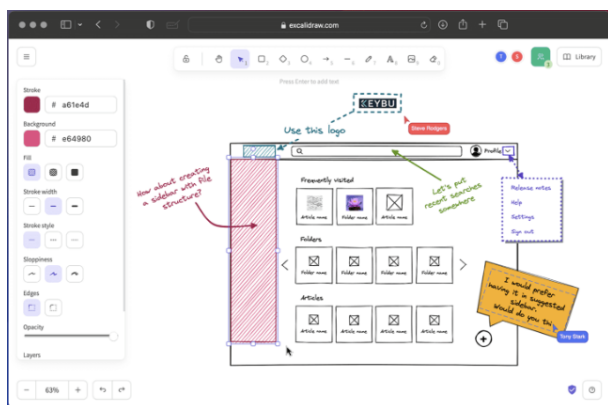


Fig. 5. Exemplo de esboço no Excalidraw.

#### E. Google Workspace (Docs, Sheets, Slides)

Suíte de ferramentas de produtividade do Google que inclui Docs, Sheets, Slides e Drive. Será utilizado para criar e editar documentos relacionados ao projeto e preparação da equipe, como relatórios, planejamentos e atas de reuniões, e controle financeiro. Além disso, o Google Drive oferece recursos de armazenamento em nuvem que facilitam a colaboração e o trabalho em equipe. A Roboforge utilizará para que diversos membros possam trabalhar nos mesmos documentos e arquivos simultaneamente, mesmo que de forma remota.

#### F. Canva

Ferramenta online para criação de designs gráficos e apresentações. Será utilizada para criar apresentações mais personalizadas para auxiliar na comunicação em reuniões, desenvolver materiais de marketing e criar designs para redes sociais e outros canais de comunicação.

#### G. Discord

Plataforma de comunicação online que oferece voz, vídeo e chat de texto. Será utilizada para realizar reuniões remotas da equipe, discutir o andamento do projeto em tempo real, através de canais de voz e chat de texto, compartilhar arquivos e links relevantes para o projeto.

Vale a pena ressaltar que a equipe considera importante sempre buscar utilizar o máximo de softwares com conexão com a nuvem, em todas as partes do projeto, justamente para simplificar o processo de compartilhamento de arquivos e possibilitar o trabalho concomitante de diversos membros numa mesma atividade, mesmo que remotamente.

### IV. ESTRATÉGIA

A estratégia consiste em panoramas para guiar os esforços da equipe em torno do campeonato, com o intuito de alinhar as expectativas, assim como definir metas objetivas a serem cumpridas dentro do intervalo de tempo até a competição.

Primeiramente foi discutido a disposição geral do fluxo de operações a serem executadas pelo robô, assim como quais seriam os requisitos para que este consiga concluir todas as tarefas de forma prática da forma mais direta possível. O desafio foi dividido em sub-tarefas menores como: auto-localização do robô no mapa, determinação de rotas eficientes entre quaisquer dois pontos do cenário, busca e embarque de passageiros, desembarque de passageiros e detecção de obstáculos.

Assim como em 2023, quando o desafio da categoria Standard Educational Kit era semelhante ao da Challenge 2024, a equipe seguiu a seguinte linha de raciocínio: a partir dos requisitos básicos de um robô, ou seja, um sistema único capaz de realizar de alguma forma todas essas subtarefas, foi pensado primeiramente no conjunto de sensores que fizesse isso ser possível. Com relação a motores, foi considerado o básico para locomoção (movimentação das rodas) e movimentação de garras. Como as tarefas a serem realizadas exigem maior diversidade de sensores que as portas disponíveis em um único controlador LEGO EV3, a equipe teve o entendimento que seria realmente necessário pelo menos dois controladores, assim como na estratégia levada à competição em 2023.

A nível de contexto, foi levantado, por exemplo, que seriam necessários, no mínimo: 2 sensores de cor destinados a identificação de marcações no chão para locomoção efetiva e auto-localização, 1 sensor de distância para identificação de obstáculos, 1 sensor de distância adicional para diferenciação dos passageiros entre crianças e adultos e 1 sensor de cor para identificação dos passageiros. Foi trazida à discussão este ano os pontos positivos e negativos da utilização de sensores que o robô construído no ano anterior trazia.

Em 2023, o robô apresentado pela Roboforge carregava 4 sensores de cor apontados para o chão, assim como sensores de distância na frente e atrás do robô. Todos esses eram responsáveis pela leitura do mapa e dos obstáculos e coordenação do robô para a movimentação com as rodas, que poderia ser tanto orientada para a frente do robô, quanto

para trás. O robô havia sido feito pensando em andar boa parte do percurso de “ré”, então havia essa necessidade dos sensores na parte de trás. Além desses, também foi colocado um sensor do lado direito do robô para a varredura dos passageiros e os sensores da frente (cor e distância) realizavam a diferenciação dos perfis dos passageiros (cor, criança/adulto), posicionados praticamente dentro da garra.

Em 2024, a equipe propõe uma abordagem diferente. Não foi vista grande vantagem em fazer o percurso “de ré” em 2023 comparado à possibilidade de fazer ele completamente orientado pela frente do robô, o que tira a necessidade dos sensores de cor e do sensor de distância na parte de trás do robô. Além disso, dada a novidade desse ano que traz a plataforma de embarque com uma elevação, e os passageiros não podendo ser arrastados pelo mapa, surge a necessidade de uma garra que levante o passageiro do chão. Todas essas observações e possibilidades estão descritas com mais detalhes na seção “Aspectos Técnicos de Engenharia” a seguir.

Dados os requisitos do robô, alguns pontos-chaves da estratégia foram classificados como dificuldades maiores que exigiriam maiores tratativas e testes relacionados. Com relação à determinação das rotas, não apenas seria necessário avaliar a aplicação de algoritmos de planejamento de caminhos (*pathfinding*), como também garantir a capacidade do robô de se locomover dentro das rotas planejadas de maneira adequada. Quanto à localização, não será uma tarefa trivial determinar a posição inicial do robô, e isso exigirá algumas tratativas mais complexas que trabalhos semelhantes realizados em competições anteriores pela equipe. O manuseio dos passageiros, e o dever de desembarcá-los devidamente de pé sem derrubar outros possivelmente na mesma localidade também foi considerado uma tarefa a ser trabalhada com maior atenção.

Posteriormente foram discutidas as metas e o planejamento estratégico a ser adotado pelos setores de programação e engenharia individualmente para a execução do desafio proposto da melhor forma possível, explicados detalhadamente nos tópicos abaixo.

## V. METODOLOGIA

Inspirando-se em técnicas de metodologias ágeis, a equipe traçou um plano de execução dos artefatos de desenvolvimento necessários até a data da competição. O desenvolvimento da solução será orientado por sprints iterativas de tamanho semanal. Um pequeno relatório simples das atividades realizadas, problemas encontrados e próximos objetivos específicos deverá ser dado por cada integrante em um dia combinado de cada semana. Os objetivos gerais são mapeados para cada mês, de forma que ao final de cada mês uma data será especificada para a ocorrência da reunião de revisão mensal.

A gestão do projeto será controlada através de um sistema de gerenciamento de tarefas, como o Trello, Jira ou até funcionalidades existentes dentro do GitHub para isso. A proposta é seguir o modelo Kanban, e deixar explícito a todos

os membros da equipe o tempo todo como está o avanço das tarefas.

Foram ilustrados na tabela abaixo os objetivos estipulados pela equipe, seus prazos e as datas das reuniões mensais destinadas a revisar as metas definidas, acompanhar o prosseguimento destas com base no feedback da equipe e adaptá-las caso seja necessário.

Objetivo	Prazo/Data
Construção de uma base funcional para locomoção básica no mapa seguindo as diretrizes da estratégia	08/2024
Desenvolvimento dos primeiros algoritmos de localização e navegação (inclusive considerando obstáculos)	08/2024
Primeiras iterações da construção da garra do robô	08/2024
Reunião mensal de Agosto	25/08/2024
Continuação do desenvolvimento dos algoritmos de navegação	09/2024
Implementação da varredura e detecção de passageiros	09/2024
Iterações para melhoria no mecanismo da garra	09/2024
Fechamento da rotina completa de localização e entrega de múltiplos passageiros	09/2024
Reunião mensal de Setembro	29/09/2024
Integrações finais do código, mais iterações de desenvolvimento para aumento de eficiência e otimização de todo o processo	10/2024
Reunião mensal de Outubro	30/10/2024

## VI. ASPECTOS TÉCNICOS DE ENGENHARIA

Com base principalmente no robô usado na competição do ano passado, foi possível perceber alguns lados positivos e negativos da engenharia usada nele, e assim possíveis melhorias a serem pensadas, testadas, e feitas. A locomoção dos robôs será realizada por meio de rodas comuns, impulsionadas por motores grandes ou médios. Testes serão realizados nas primeiras semanas de desenvolvimento da base estrutural para determinar quais motores oferecem melhor estabilidade, agilidade, velocidade e precisão. Em relação aos sensores, o robô contará com um conjunto de sensores infravermelho, ultrassônico e de cor para se localizar no mapa, encontrar passageiros e transportá-los ao destino correto.

A base do robô terá dois motores para a locomoção, um para cada roda (esquerda e direita). Um motor médio ou grande será usado para movimentar a garra que pegará os passageiros, sendo que testes definirão a melhor opção de motor, se possível usar um mesmo motor que consiga fazer a função “*grab and lift*” (segurar e levantar, com um único robô), caso contrário será optado por usar um motor para a garra abrir e fechar com os passageiros e outro motor para levantá-los do chão. Nesse robô dois controladores serão



empregados na montagem do robô, contribuindo para o peso e estabilidade e permitindo a integração e controle de todos os motores e sensores utilizados, já que com apenas um não seria possível empregar todas as portas necessárias para o controle dos sensores. A disposição dos dois controladores, da garra e dos outros sensores será um tópico importante de estudo durante o desenvolvimento do projeto.

Para os sensores de distância, a equipe usará sensores ultrassônicos e infravermelhos. O sensor infravermelho detectará a presença dos passageiros durante a partida, será disposto na lateral, de forma que ao andar ao lado da plataforma de embarque conseguirá detectar a distância próxima dos passageiros. Será testada a possibilidade de usar outro sensor infravermelho ao lado para que com diferentes alturas seja possível diferenciar o passageiro criança do passageiro adulto, enquanto o sensor ultrassônico será usado para localizar obstáculos devido à sua maior cobertura de área em comparação com o infravermelho, que tem maior precisão. O sensor ultrassônico também será testado para diferenciar o tamanho do passageiro, pois se localizado a 15 cm de altura em frente a garra, apenas captaria o passageiro adulto, conseguindo diferenciar então os passageiros de acordo com a faixa etária. Os sensores de cor serão utilizados para o alinhamento do robô e para detectar as cores dos estabelecimentos e dos passageiros. Diferente do ano passado que foi utilizado quatro sensores de cor para as extremidades do robô, será apenas usado duas nas extremidades da frente, considerando que o robô andará apenas para frente.

A primeira garra a ser testada será projetada para abraçar o passageiro horizontalmente, garantindo que ele não caia durante a locomoção, e irá levantar após estiver completamente agarrada ao passageiro, para que o mesmo não atrapalhe na hora de passar pelas lombadas, e também possibilite que o sensor ultrassônico seja desobstruído para localizar obstáculos a frente.

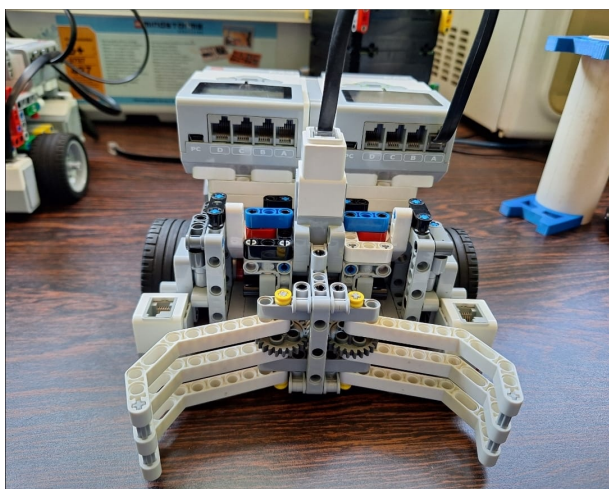


Fig. 6. Projeto inicial da base do robô com garra e dois sensores de cor.

## VII. ASPECTOS TÉCNICOS DE PROGRAMAÇÃO

Com o intuito de guiar os esforços da área de desenvolvimento da equipe, o primeiro passo do setor foi explorar

o desafio proposto pela categoria, e, levando em conta os esforços da equipe no mesmo desafio no ano anterior, discutir uma estratégia capaz de concluir o desafio com as novas atualizações nas regras, além de considerar os pontos fracos observados pela equipe anteriormente. Para isso os membros reuniram esforços para elaborar um fluxograma resumindo os algoritmos necessários e elencando as principais dificuldades do código a ser escrito.

Um fluxograma é uma ferramenta visual essencial na programação de sistemas, especialmente em projetos de robótica. Ele permite mapear de forma clara e sistemática as transições entre estados, facilitando a visualização do fluxo de execução dos algoritmos. Essa representação gráfica é fundamental para a documentação e a comunicação entre os membros da equipe, pois traduz a lógica do algoritmo em um formato acessível e compreensível.

No contexto do desafio, o diagrama desenvolvido pela equipe inclui todos os algoritmos que o robô executará, detalhando as ações que provocam mudanças de estado. Isso orienta eficientemente as atividades do setor de programação, promovendo uma abordagem sistemática que visa aprimorar a eficiência e evitar esforços redundantes.

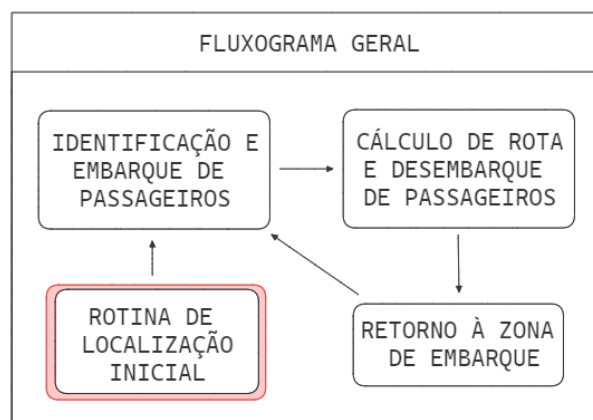


Fig. 7. Fluxograma geral

A partir dessas observações foram definidos os algoritmos necessários para a execução do desafio. Estes algoritmos são apresentados abaixo na ordem em que serão implementados.

### A. Movimentação do robô e controle da garra

Por se tratar de um aspecto que pode introduzir erros no trajeto do robô, a movimentação e seu refinamento foram elencados como o primeiro ponto a ser desenvolvido pelo setor de programação, para garantir que os demais algoritmos funcionem com uma precisão satisfatória. O controle do apêndice do robô será desenvolvido em conjunto com a movimentação, uma vez que os códigos não diferem significativamente e seu funcionamento satisfatório e robusto também será um pré-requisito para operações subsequentes.

### B. Localização inicial

Considerando que a posição inicial do robô é aleatória deverá ser implementada uma funcionalidade que permita

verificar a capacidade do robô de encontrar um ponto comum em todas as rodadas, estabelecendo uma posição e uma rotação padronizadas. Isso garantirá que, durante o desenrolar do desafio, o robô possa se localizar de maneira precisa e consistente. Este ponto comum será referido no código como a “origem” do que virá a ser um sistema de coordenadas baseado na geometria do tapete.

Para explorar a área do desafio, o robô seguirá um algoritmo específico para solução de labirintos, com o objetivo de evitar a entrada em ciclos de ações infinitas. Durante a exploração, o robô registrará em sua memória as cores das linhas encontradas, o que permitirá uma localização parcial. Com base em uma posição conhecida, o robô será capaz de alcançar a área de embarque.

Para ilustrar as ideias propostas e assegurar que o código desenvolvido esteja livre de falhas, os membros do setor de programação elaboraram um diagrama de fluxo para esta etapa do desafio. Vale ressaltar que os obstáculos não são considerados no fluxograma abaixo, mas a lógica necessária para evitar os mesmos estará presente no código final.

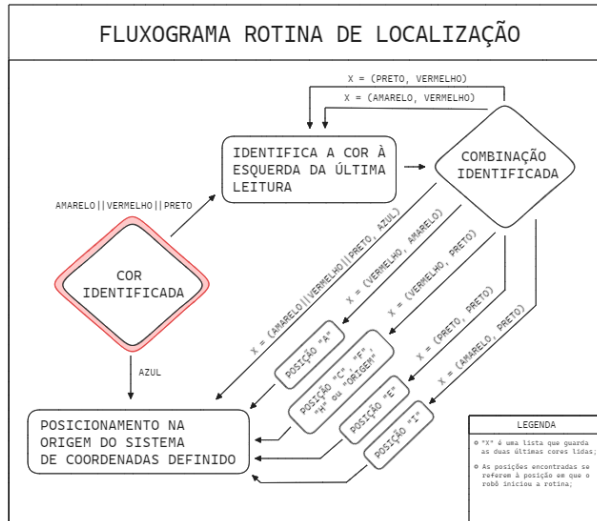


Fig. 8. Fluxograma da rotina de localização inicial.

### C. Identificação e embarque de passageiros

Para encontrar os passageiros na área de embarque, serão utilizados sensores de distância (ultrassônico ou infravermelho) posicionados na lateral do robô, enquanto que a identificação do tamanho e cor dos tubos, será feita após a coleta do mesmo pelos sensores posicionados na parte frontal.

Para otimizar o processo de embarque de passageiros a equipe notou que a coleta do tubo poderia ser feita antes da chegada à origem do sistema de coordenadas definido. Desta forma, quando o robô atingir a posição conhecida na área de desafio, este já poderia calcular a rota para o passageiro embarcado.

### D. Cálculo de rota

Após a coleta do passageiro, cabe ao robô determinar a rota mais eficiente para transportá-lo ao seu destino,

considerando os obstáculos presentes nas vias. Embora no passado a equipe tenha utilizado o algoritmo A\* (A-estrela) para solucionar este problema, é relevante destacar que a equipe considera a necessidade de explorar outras soluções para esta etapa do desafio.

O algoritmo A\* é amplamente utilizado em sistemas de *pathfinding* para encontrar o caminho mais curto entre dois pontos em um grafo ou uma matriz, levando em conta tanto o custo do caminho percorrido quanto uma estimativa do custo restante até o objetivo.

Como as posições dos estabelecimentos e a zona de embarque dos passageiros não variam entre as rodadas, o primeiro passo para o desenvolvimento da rotina de busca do robô será mapear cada posição possível da arena em uma matriz. Nessa matriz, cada elemento pode ser uma parede intransponível (representada por quadrados pretos), indicando zonas que o robô não pode ultrapassar; vias livres (quadrados brancos), onde a movimentação é permitida a qualquer momento; e possíveis obstáculos (representados pela cor cinza na imagem abaixo), que definem as posições onde um obstáculo pode estar presente. O robô deve verificar esses obstáculos antes de tentar passar por essas vias.

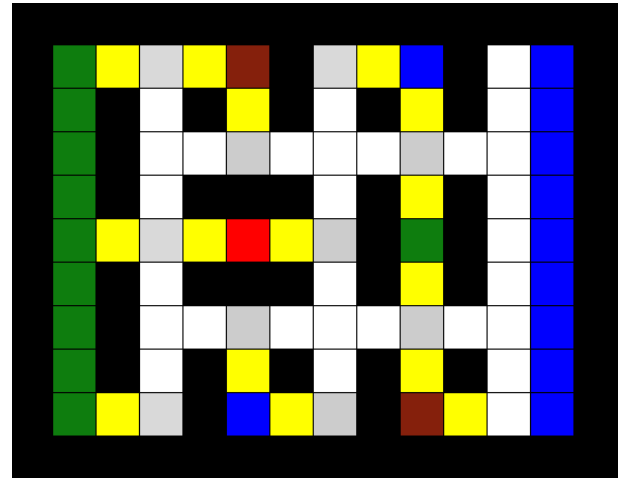


Fig. 9. Arena do desafio da RoboCup Challenge Kit 2024 traduzida em uma matriz 11x14.

No contexto do algoritmo de busca de caminho utilizado, às paredes intransponíveis foram atribuídas pesos muito altos (praticamente infinitos) para garantir que o algoritmo nunca as considerasse ao calcular a rota. As posições onde podem ser colocados obstáculos possuem pesos variáveis: se o robô não detectar um obstáculo em uma determinada posição, a essa posição é atribuída um peso baixo. No entanto, quando um obstáculo é identificado, a matriz é atualizada para refletir uma parede naquela posição, e a rota é recalculada da posição atual do robô, levando em conta a nova configuração do ambiente.

### E. Processamento de dados dos sensores

Um desafio recorrente para um bom desempenho na competição para a equipe tem sido lidar com falhas e

limitações das leituras dos sensores presentes nos kits utilizados. Entender os possíveis valores de leitura de cada sensor e projetar tratativas em *software* para melhor compreender as situações envolvidas em cada cenário tem sido parte fundamental de todo o trabalho realizado durante as preparações para competições passadas.

Os primeiros esforços nesse sentido envolveram entender os aspectos matemáticos dos valores retornados, e aplicar conceitos estatísticos para lidar programaticamente com erros e limitações. A realização de múltiplas leituras e a consideração de valores médios, mínimos ou máximos, por exemplo, já foi bastante utilizada para superar limitações de sensores de distância, por exemplo, sejam ultrassônicos ou infravermelhos. Além disso, limitações superiores e/ou inferiores, assim como a utilização de limites de taxas de alteração para definir o que seriam consideradas “leituras válidas” são métodos que têm sido amplamente utilizados e trazido bons resultados.

Porém, o processamento de dados dos sensores de cor costumam exigir tratativas mais complexas. A biblioteca *pybricks* traz a funcionalidade de retornar, a partir de uma leitura do sensor de cor, qual a cor final identificada. O grande motivador dos trabalhos descritos a seguir é a questão de o modelo de classificação de cores padrão ali presente não ser sempre preciso o suficiente para os cenários de uso da equipe, e não ser aberto a possíveis alterações. Portanto, é proposto a construção de um modelo personalizado, que atenda essas necessidades específicas e seja mais facilmente calibrável. As primeiras estratégias da equipe envolviam análise manual dos valores numéricos de múltiplas leituras em múltiplos cenários e a estipulação de intervalos significativos para classificação de cada cor, utilizando normalização dos dados no processo para considerar diferenças de leituras identificadas entre sensores diferentes (ainda que do mesmo modelo). Com o tempo, foi visto grande vantagem em construir modelos que exigissem menos intervenção manual.

Para isso, em 2023, a equipe agregou a utilização de aprendizado de máquina à solução deste problema, e obteve resultados consistentes. Neste ano é prevista a evolução deste modelo, considerando outras possibilidades e buscando melhorar a precisão e performance da identificação de cores através dos sensores. O modelo inicial considera o problema das cores como um problema de classificação, e utiliza aprendizado supervisionado sobre dados de leituras coletados diretamente do robô para treinamento. Dessa forma, os esforços de calibração passam a ser apenas uma execução guiada de *software* no robô, onde um operador humano coloca o sensor em questão em cada uma das cores por vez, numa ordem pré-determinada e conhecida. Múltiplas leituras são feitas e “etiquetadas” de acordo com a cor, o que compõe o conjunto de dados.

O conjunto de dados é então submetido a um algoritmo de treinamento (executado fora do robô, num computador, por exemplo), que gera um modelo de classificação a ser posteriormente implantado no *software*. A intenção é que a definição dos parâmetros que caracterizam cada cor seja feita de forma a generalizar o que é observável pelo algoritmo

no conjunto de dados. Os métodos que a equipe pretende explorar incluem, mas não se limitam à construção de árvores de decisão com os algoritmos CART e/ou ID3, através da biblioteca *Scikit-Learn*, disponível em *Python*.

## VIII. CONCLUSÃO

Não apenas uma elaboração de estratégia, se fará necessário implantar tais sistemas de gestão e desenvolvimento técnico descritos por meio deste documento, além de se prosseguir com a certeza de que será necessário lidar com adversidades imprevisíveis, e haver metodologias preparadas desde o início para tal. É de entendimento geral no time que as ideias descritas aqui são apenas uma perspectiva inicial, e em geral estão passíveis de alteração conforme o decorrer da execução do projeto. Traçados os objetivos e planos, a Equipe RoboForge de Robótica se propõe a prosseguir com as ações efetivas que garantam um desempenho satisfatório na divisão KIT da categoria Challenge da CBR 2024, promovendo o desenvolvimento pessoal e científico, humano e técnico, dentro e fora da competição.

## REFERENCES

- [1] “Regras Challenge 2024”, versão 1.1 - Agosto, 2024.
- [2] “Getting started with LEGO® MINDSTORMS Education EV3 MicroPython”. Disponível em <https://pybricks.com/ev3-micropython/>. Acesso em agosto de 2024.
- [3] “ev3dev is your EV3 re-imagined”. Disponível em <https://www.ev3dev.org/>. Acesso em agosto de 2024.
- [4] “MicroPython - Python for microcontrollers”. Disponível em <https://micropython.org/>. Acesso em agosto de 2024.
- [5] “Red Blob Games: Introduction to A\*”. Disponível em: <https://www.redblobgames.com/pathfinding/a-star/introduction.html>. Acesso em agosto de 2024.
- [6] “scikit-learn: Machine Learning in Python”. Disponível em: <https://scikit-learn.org>. Acesso em agosto de 2024.
- [7] “Decision Trees”. Disponível em: <https://scikit-learn.org/stable/modules/tree.html>. Acesso em agosto de 2024.