TECNOLOGIA E ENGENHARIA

EQUIPE ALPHA

Apresentado pela equipe

ALPHA TBR

TBR TORNEIO BRASIL DE ROBOTICA

Firjan SENAI SESI

SUMÁRIO

Sumário	1
informações do time	2
Organização da nossa equipe	3
Membros do nosso time	4
Moodboard	5
Design do robô	6 à 8
Linha do tempo	9
Evolução do robô	10 e 11
Estratégia de mesa	12 e 13
Organização de estratégia	14 e 15
Dificuldades	16
Programação	17 à 26
Softwares	27

INFORMAÇÕES DO TIME ALPHA TBR

NOME DA EQUIPE: ALPHA TBR

CATEGORIA: High

TÉCNICO/MENTOR: Marco Antônio, Fabiane Martins e Paulo Vitor.

INTEGRANTES: Alexandre, Flora Maria, Guilherme, Isabelly, João Phelipe, João pedro, Matheus Cardoso, Mayara Lúcia, Paulo Daniel e Pedro Douglas.

PATROCINADORES: Grupo JCA, Firjan SENAI SESI, Calara Software, SHARE, Meninas Digitais, Manitshow, Socorrista Polo Curicica e Bombeiro Civil Polo Curicica.

REDES SOCIAIS: Instagram: @alphatbr

ORGANIZAÇÃO DA NOSSA EQUIPE



MARCO ANTONIO
Técnico

Professor de Física no SESI e técnico da equipe. Responsável pela **coordenação geral** da equipe, fornecendo suporte na **construção do robô** e ajudando em questões **técnicas do projeto.**



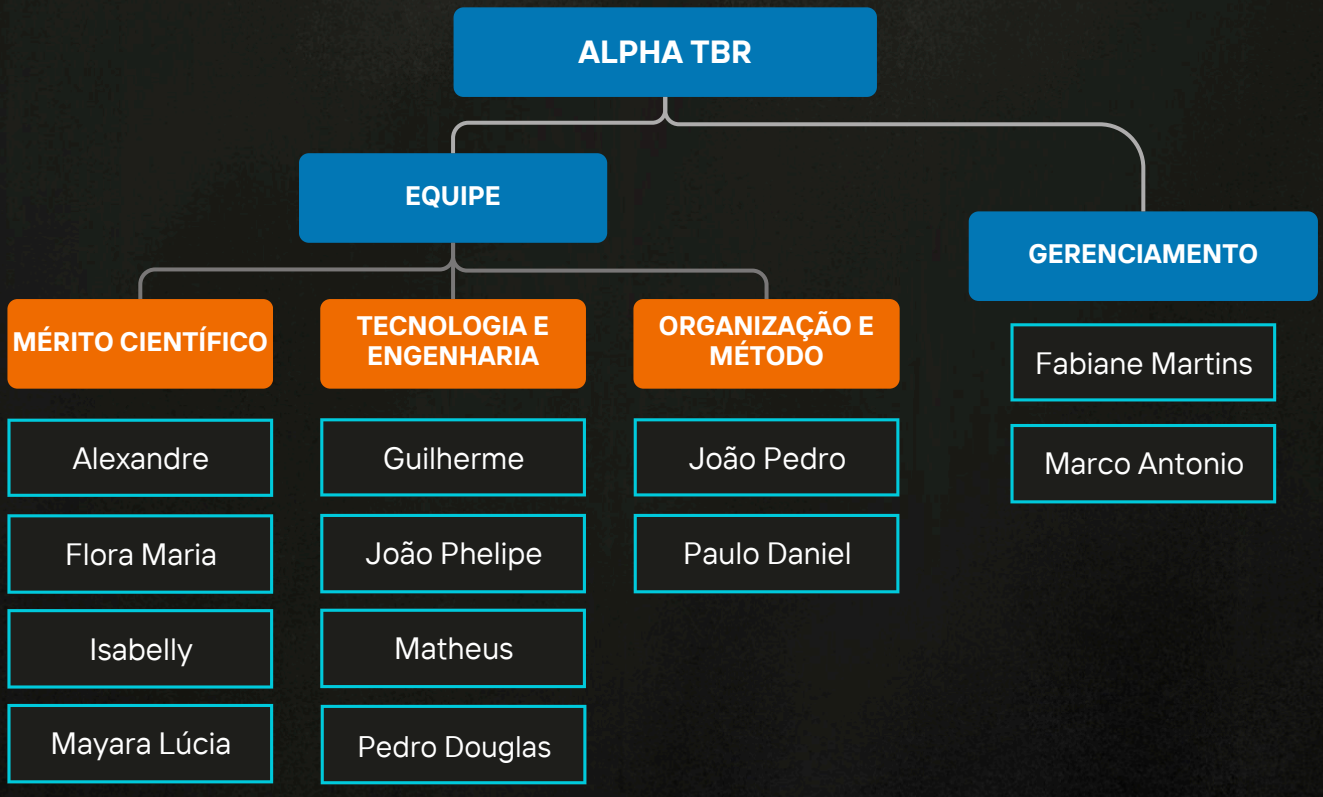
FABIANE MARTINS
Técnica

Professora de Química no SESI e técnica da equipe. Encarregada da **gestão e liderança**, estabelecendo metas, mantendo a **motivação da equipe** e oferecendo suporte em aspectos **técnicos do projeto.**

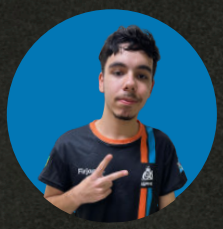


PAULO VITOR
Mentor

Auxiliar técnico do SESI, ex-membro da equipe, contribui para os **processos internos**, auxiliando no desenvolvimento de documentações, **definição de metas** e otimização das **tarefas de organização.**



OS MEMBROS DO NOSSO TIME



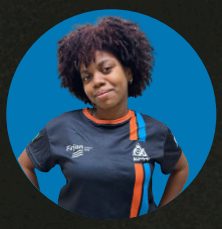
ALEXANDRE
Mérito científico
17 anos;
Técnico em eletroeletrônica;
1ª temporada no TBR.



FLORA MARIA
Mérito científico
16 anos;
Técnico em logística;
2ª temporada no TBR.



GUILHERME
Tecnologia e engenharia
16 anos;
Técnico em eletroeletrônica;
2ª temporada no TBR.



ISABELLY
Mérito científico
18 anos;
Técnico em informática;
1ª temporada no TBR.



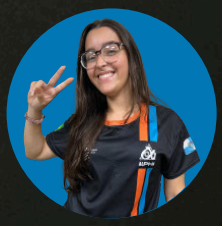
JOÃO PHELIPE
Tecnologia e engenharia
16 anos;
Técnico em logística;
1ª temporada no TBR.



JOÃO PEDRO
Organização e Método
16 anos;
Técnico em logística;
1ª temporada no TBR.



MATHEUS CARDOSO
Tecnologia e engenharia
16 anos;
Técnico em logística;
1ª temporada no TBR.



MAYARA LÚCIA
Mérito científico
19 anos;
Técnico em informática;
1ª temporada no TBR.



PAULO DANIEL
Organização e Método
17 anos;
Técnico em informática;
2ª temporada no TBR.

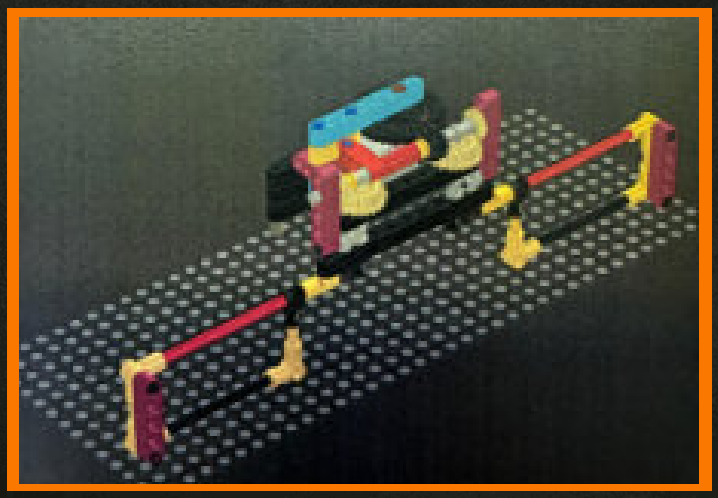


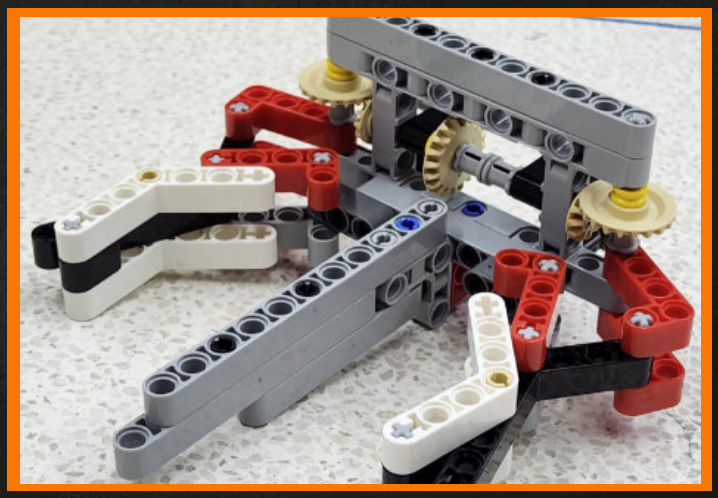
PEDRO DOUGLAS
Tecnologia e engenharia
15 anos;
Técnico em logística;
1ª temporada no TBR.

MOODBOARD

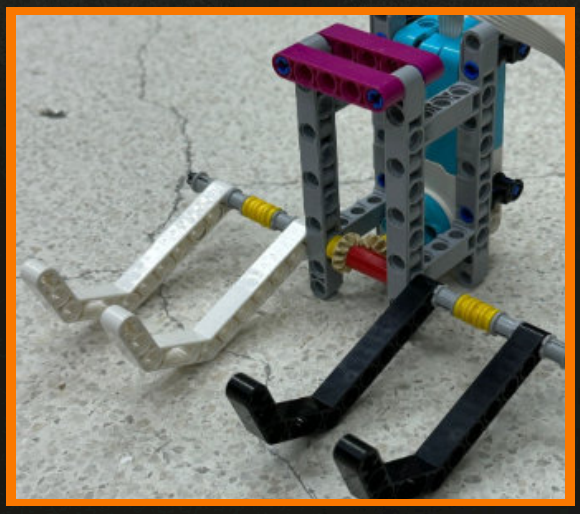
VEJA NOSSO PROCESSO

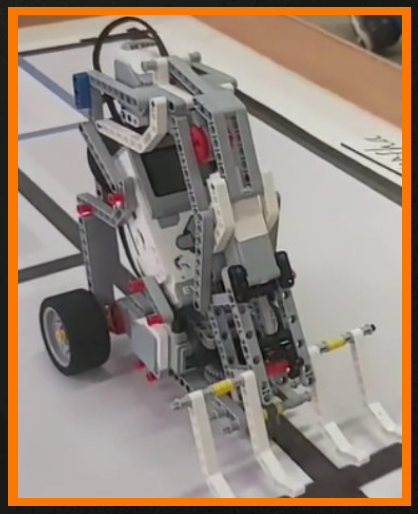
O moodboard é uma coletânea de referências visuais, como ferramenta para dar vida e visibilidade a uma ideia ou conceito. Essa abordagem nos permite reunir elementos inspiradores, como imagens, cores e texturas, que ajudam a transmitir a essência do projeto e a orientar o desenvolvimento criativo. Ao criar um moodboard, conseguimos visualizar melhor a direção que desejamos seguir e comunicar de forma mais eficaz nossos objetivos associados à proposta.







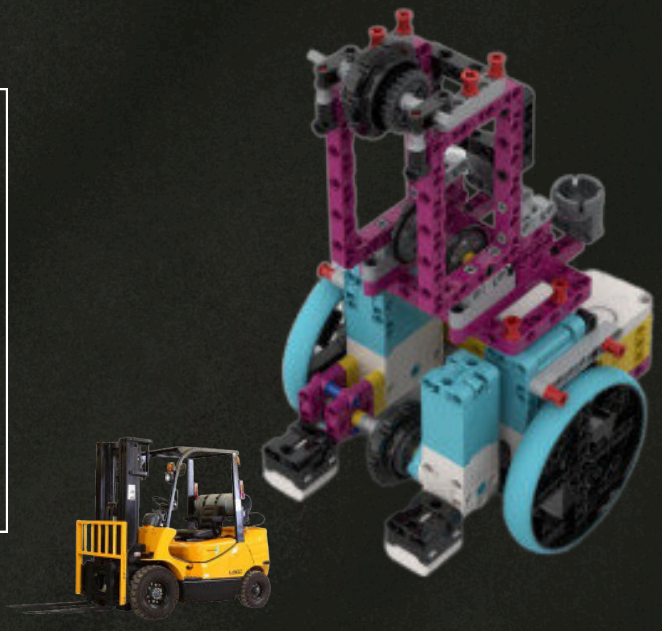




DESIGN DO ROBÔ
TIMELINE DO PROCESSO

1ª VERSÃO

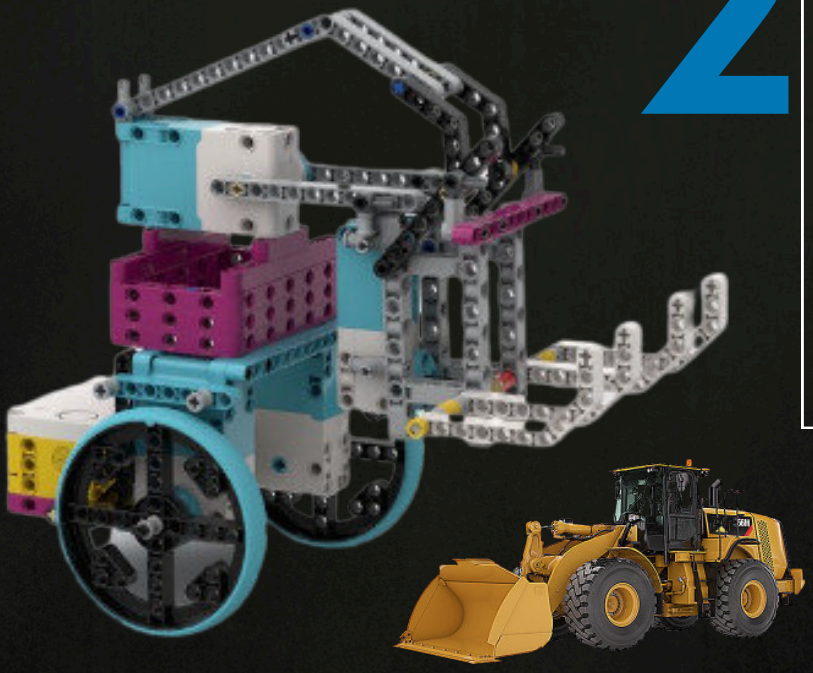
Esse foi o primeiro protótipo que montamos a partir de análises de mecanismos de elevação com esteiras. Percebemos que seu design se assemelha muito com uma empilhadeira. Testamos ele junto da estratégia no tapete e concluímos que esse não seria o design ideal, pois não tinha bom balanceamento de centro de massa, possuía um sistema de elevação impreciso, lento e não tinha tanta robustez em seu chassi.



EMPILHADEIRA

2ª VERSÃO

Decidimos alterar nossa estratégia, fizemos outro design completamente diferente, abandonando a esteira. Adotamos um braço para elevar as árvores, com **pontos de liberdade** para levantar a garra, que utiliza um sistema similar a um **CAT Loader**, onde as árvores se encaixam entre as barras. Percebemos que esse sistema estava muito pesado, o braço não era capaz de erguer a garra junto das árvores, então tivemos que procurar por mais referências.



CAT LOADER

DESIGN DO ROBÔ
TIMELINE DO PROCESSO

3ª VERSÃO

Seguimos a mesma estratégia, mas fizemos algumas alterações como a adição de uma relação entre engrenagens onde a menor (z1) passava rotação para a maior (z2) gerando torque para facilitar com que a garra eleve as árvores.

i=z2/z1

i=40/24=1,6

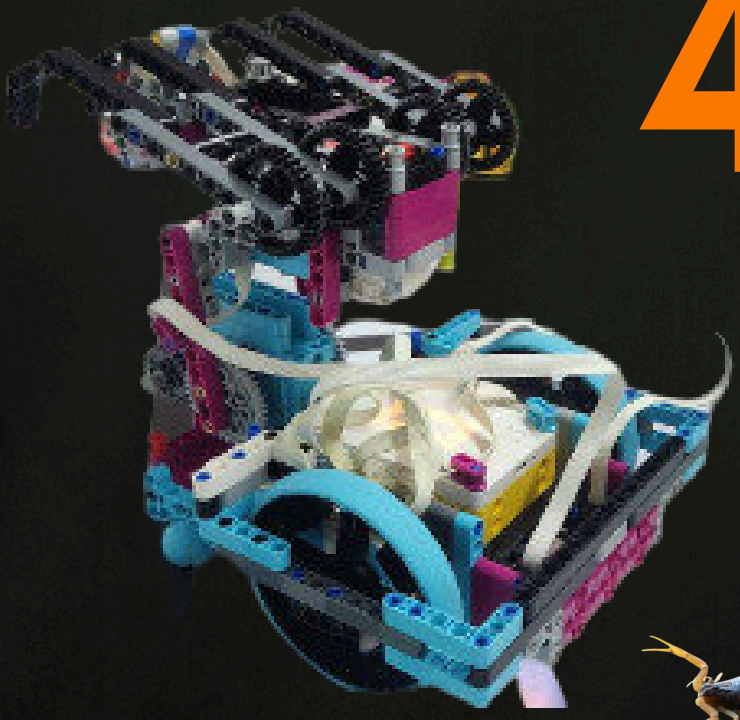
A principal característica desse design é que reutilizamos muitos recursos bem-sucedidos do anterior, como as rodas sendo o ponto mais à frente do chassi e uma bilha na parte traseira, que, juntos, trazem mais estabilidade para o robô na hora de subir na rampa.





TRATOR DA JOHN DEERE

4ª VERSÃO



Fizemos mudanças bruscas no chassi. Além de construirmos um para-choque total em volta das rodas para mitigar as chances das rodas se desconectarem do robô durante o round, centralizamos a posição do hub em relação as rodas, pois em sua programação estávamos utilizando movimentação com PID, baseada na angulação dada pelo giroscópio. Além disso, mudamos a posição da garra para centralizar melhor a massa do robô e facilitar com que a garra pegue as árvores.



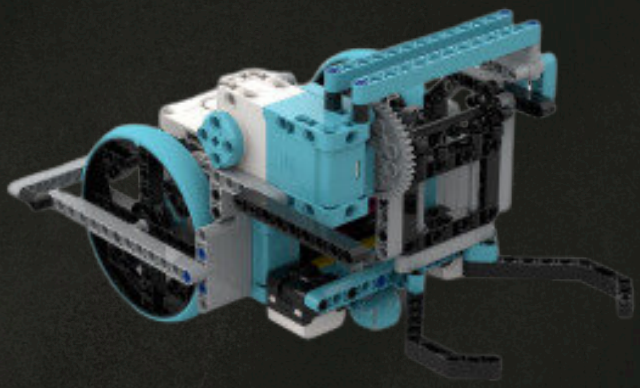
ESCORPIÃO

DESIGN DO ROBÔ
TIMELINE DO PROCESSO

5ª VERSÃO

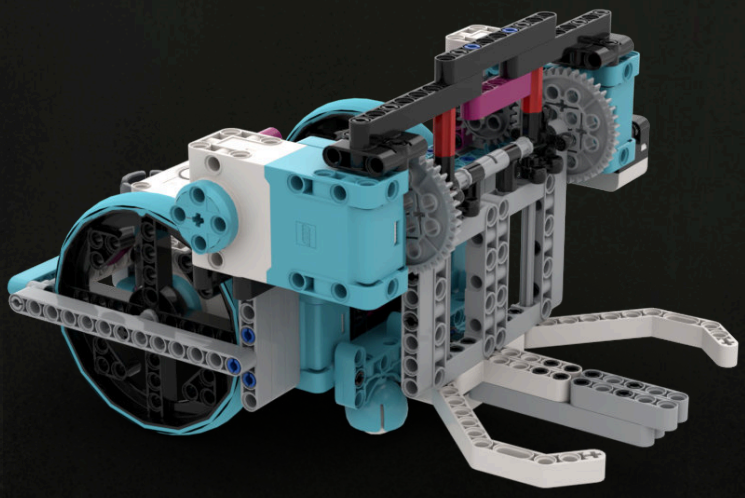
Essa foi a versão final do robô na última temporada e a que utilizamos como base para o robô atual. Ela continha 2 sensores de cor para uma leitura precisa das cores na rampa e para um seguidor de linha mais preciso. Além disso, possuía uma relação de engrenagens onde a rotação é transmitida da menor para a maior, gerando 5 vezes mais torque:

I = Z2/Z1 I = Z3/Z2 ITotal = 3 x 1,666
I = 24/8 I = 40/24 ITotal = 5
I = 3 I = 1,666



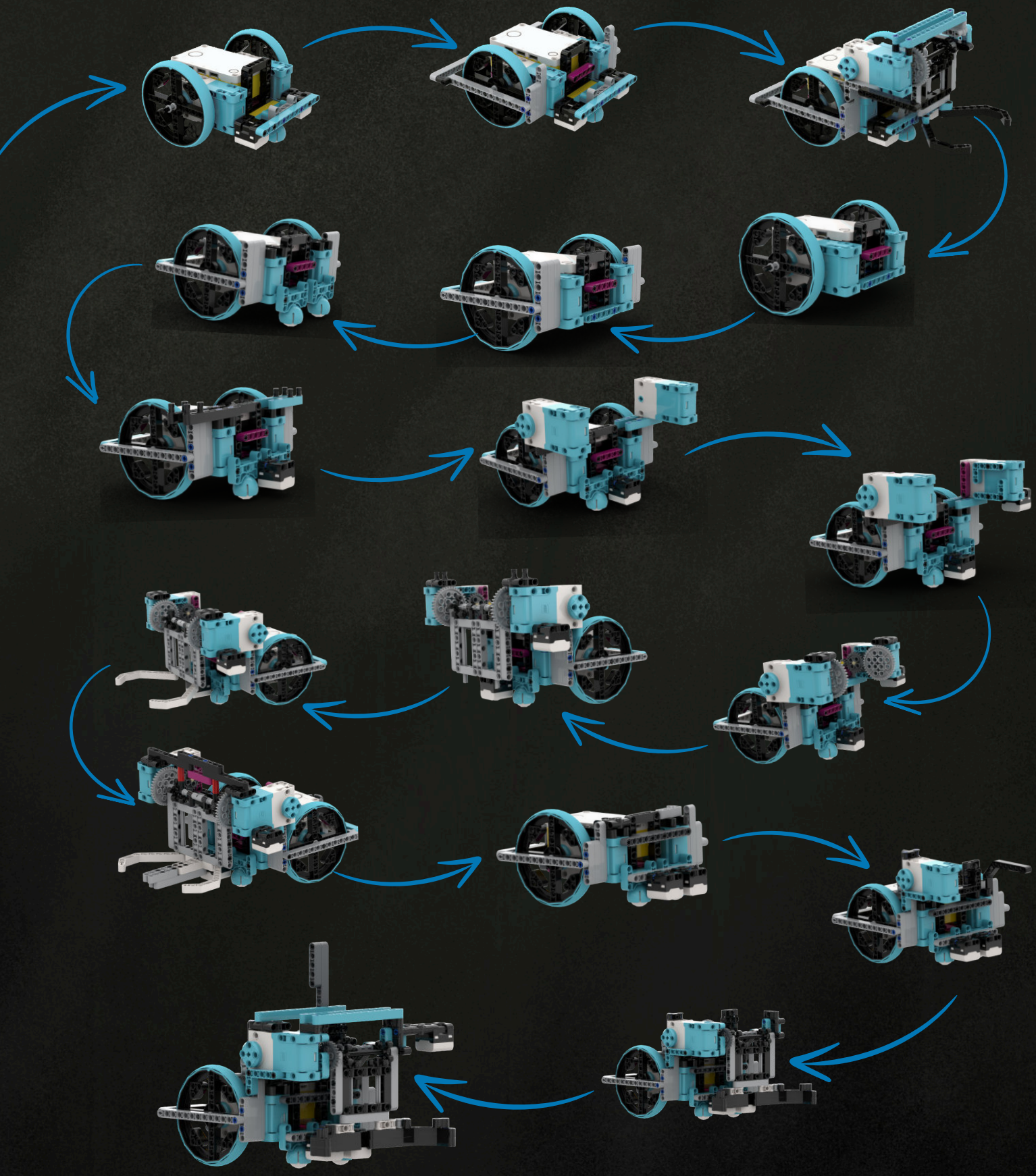
6ª VERSÃO

Fizemos mudanças bruscas no chassi. Removemos um sensor de cor e adicionamos mais um motor para que, a partir daí, acoplar mais um grab and lift, possibilitando coletar as duas árvores de uma vez e colocar uma por vez nas suas devidas áreas de reflorestamento, sendo melhor que o design anterior no quesito de coleta, porém muito maior e pesado, consequentemente mais impreciso no seguidor de linha.



LINHA DO TEMPO

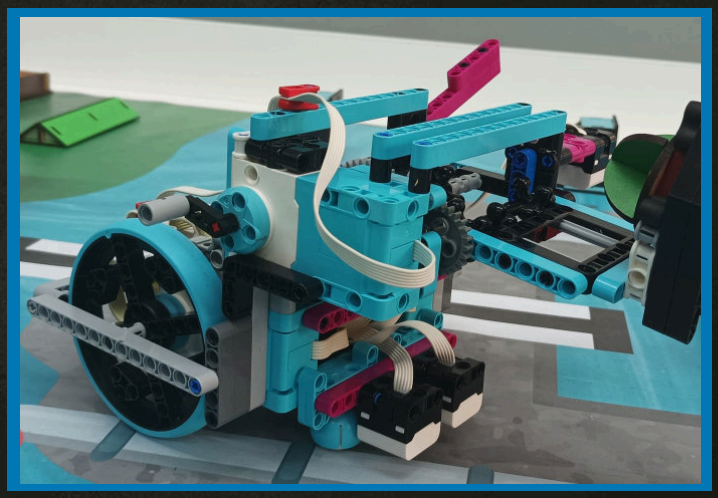
DO NOSSO ROBÔ



EVOLUÇÃO DO ROBÔ
ALPHA TBR

DIFICULDADES NA CONSTRUÇÃO

Um dos principais desafios na construção do robô foi montar um mecanismo com um sistema de elevação eficaz e simples, que tivesse um centro de massa ideal e que conseguisse coletar e pontuar as árvores em suas devidas áreas de reflorestamento com facilidade e precisão.



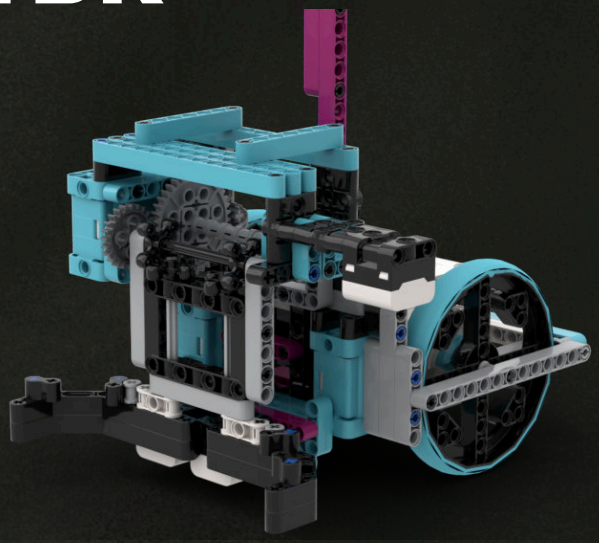
Dentre os designs citados anteriormente, percebemos que os que possuíam bilhas nas extremidades frontais ou traseiras possuíam facilidade em realizar o desafio, e melhor desempenho para subir a rampa de acesso.



EVOLUÇÃO DO ROBÔ
ALPHA TBR

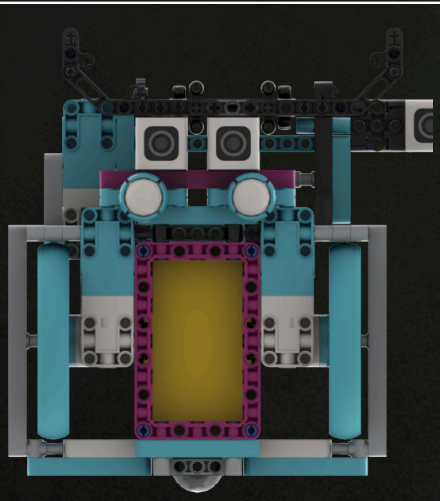
FICHA TÉCNICA

Motores.....3
Sensores de cor.....3
Giroscópio.....1
Peças.....200pcs
Tamanho.....20x20xcm
Relação de engrenagem.....3:1



O modelo final é, sem dúvida, o mais adequado para nossa estratégia. Trata-se de um robô extremamente compacto e simples, mas altamente eficiente e otimizado para o desafio. Seus dois sensores de cor na frente permitem que ele siga linhas de forma autônoma, e o sensor no canto superior direito detecta tanto árvores quanto poluentes, possibilitando a neutralização destes com sua cancela

Por meio de análises no caderno de engenharia do robô, desenvolvemos uma garra com um sistema de "agarrar e levantar". A transferência de rotação entre as engrenagens faz com que a garra se eleve ao fechar. Diminuímos a abertura máxima para incrementar a precisão na captura de uma árvore por vez e incorporamos borrachas para otimizar a coleta de árvores menores.





Inspiração

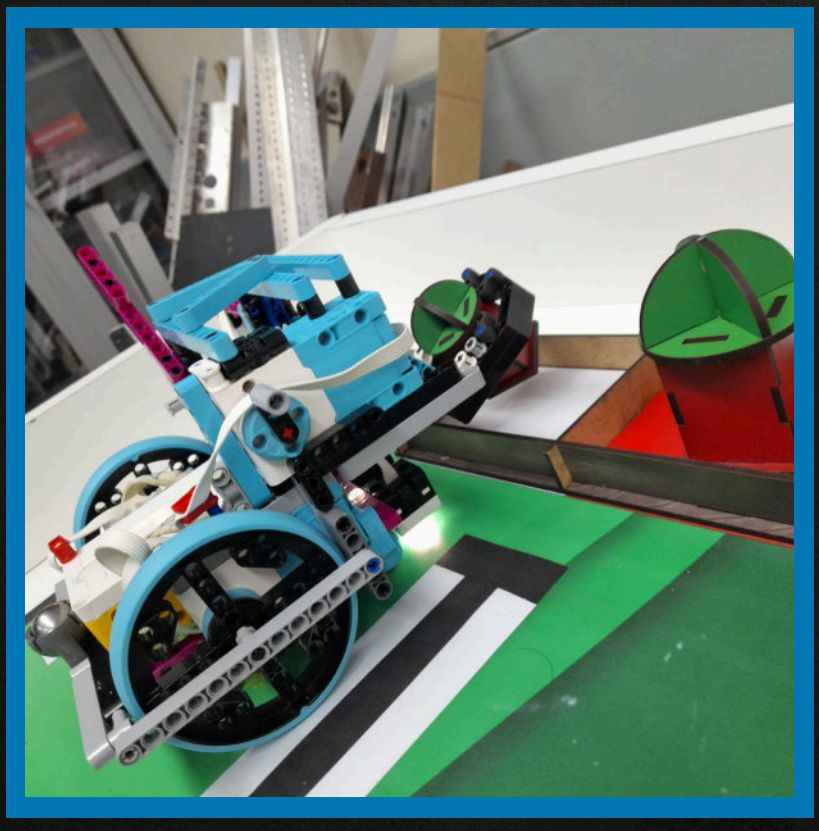
ESTRATÉGIA DE MESA

REFLORESTAMENTO

MISSÕES

1º REFLORESTAMENTO

A estratégia atual consiste em realizar o **Reflorestamento primeiro**, além de ser a rota mais eficiente, desenvolvemos a estratégia de forma que o robô pare em uma posição favorável para seguir para a segunda missão, a **Coleta e tratamento de água**, o verdadeiro problema desta missão foi a ordem das árvores por não serem fixas, por isso foi necessário diversos testes, e programações para caso a árvore maior estivesse na direita e a árvore pequena na esquerda, ou vice e versa. Decidimos utilizar o sensor superior para identificar a árvore, e o sistema de grab and lift para levanta-las e levá-las as suas respectivas áreas de reflorestamento. E somente foi necessário repetir esse processo no final da mesa.



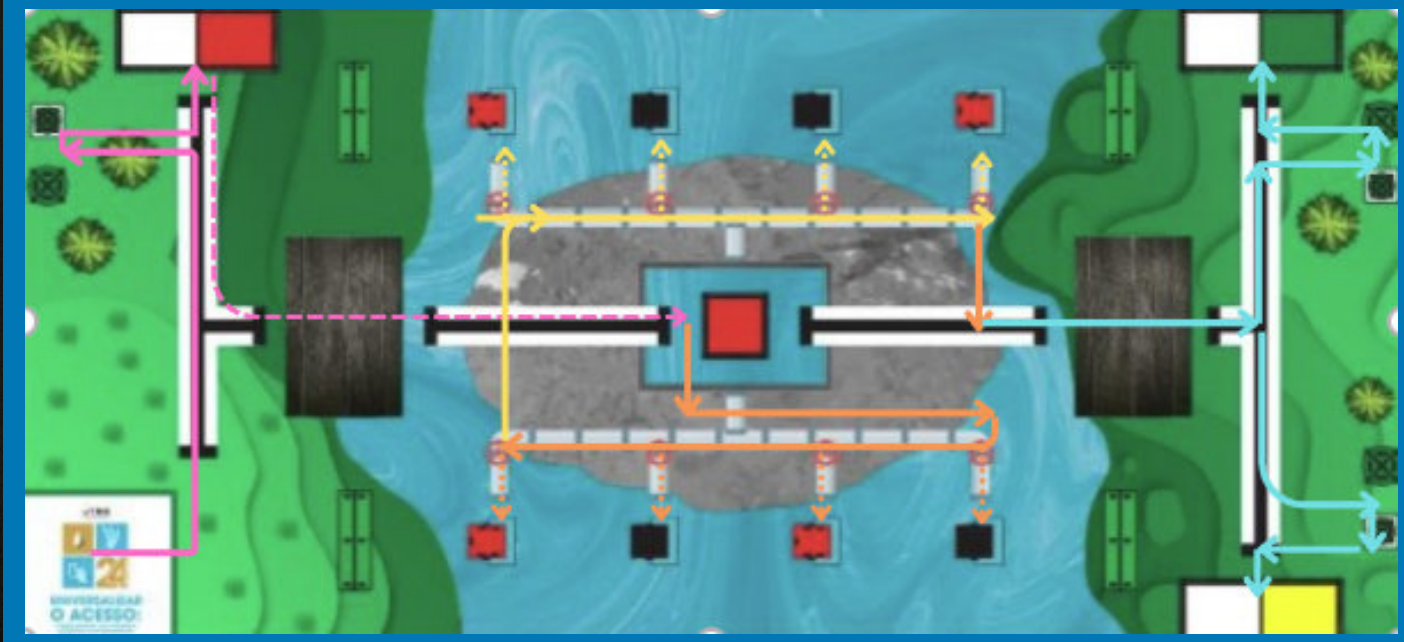
ESTRATÉGIA DE MESA

REFLORESTAMENTO

MISSÕES

1º REFLORESTAMENTO

Após terminar a **primeira missão e identificar a cor do sistema de coleta e tratamento de agua** no meio, decidimos que seria melhor para a estratégia, se o robô seguir primeiro para os **poluentes da direita**, primeiro, ele ignoraria os poluentes, até chegar ao último, onde ele iria virar 180° e voltar até o primeiro poluente, porém lendo e derrubando conforme necessário, após ter feito a parte da direita, o robô viraria 90° para a direita, em direção aos **poluentes da esquerda**, após chegar no primeiro poluente, ele viraria 90° para a direita e seguiria reto, já lendo e derrubando os poluentes de acordo com as cores, após chegar ao último poluente, o robô viraria 90° para a direita, até encontrar a linha preta, onde viraria, 90° para a esquerda, e seguiria para a terceira missão, que também é o Reflorestamento, **logo ele repetiria o mesmo processo da primeira missão.**



ORGANIZAÇÃO

Alternativas para a estratégia

PLANO A

MISSÃO	QUANTIDADE	PONTOS
Reflorestamento part. 1	2 árvores	70 pts.
Coleta e tratamento de água	4 poluentes	290 pts.
Reflorestamento part. 2	4 árvores	140 pts.
Total		500 pts.

PLANO B

MISSÃO	QUANTIDADE	PONTOS
Reflorestamento part. 1	2 árvores no low	70 pts.
Coleta e tratamento de água	4 poluentes	290 pts.
Reflorestamento part. 2	2 árvores	70 pts.
Total		430 pts.

Fizemos um plano B, caso haja algum imprevisto na estratégia A, realizando 24% menos pontos na arena.

ORGANIZAÇÃO

Alternativas para a estratégia

PLANO C

MISSÃO	QUANTIDADE	PONTOS
Reflorestamento part. 1	2 árvores	70 pts.
Coleta e tratamento de água	2 poluentes	140 pts.
Reflorestamento part. 2	4 árvores	140 pts.
Total		220 pts.

E realizamos o plano C, caso haja algum imprevisto nas estratégias A e B, realizando 30% menos pontos na arena.

ANÁLISE SWOT

Modelo atual

FORÇAS	FRAQUEZAS
• Precisão; • Autonomia; • Design compacto; • Centralização de Massa; • Hub com fácil acesso.	• Baixa velocidade de locomoção; • Cabos desorganizados.
OPORTUNIDADES	**AMEAÇAS**
• Adição de mais um sensor; • Mais complexidade ou rapidez no sistema de coleta.	• Saida da base; • Confusão de cores; • Se perder ao seguir linha.

DIFICULDADES NA PROGRAMAÇÃO

ALTA PRECISÃO

Uma das principais dificuldades que tivemos quanto a programação foi ter uma alta precisão na locomoção, uma vez que, para colocar as árvores em seus respectivos lugares, o robô teria que evitar empurrar as das árvores para fora da mesa(que acarretaria na perca de pontos), para quando voltar conseguir recolher a outra árvore sem que haja riscos de não conseguir pegar a árvore. De maneira ágil conseguimos superar, a partir da ajuda de todos os programadores da equipe.

SUBIR E DESCER AS RAMPAS

Um momento da qual nós encontramos uma dificuldade foi na hora em que o robô teria que passar pelas rampas.

A dificuldade em questão se devia ao fato do robô descer a rampa de uma maneira muito brusca, fazendo ele se desviar da rota. O desafio aconteceu por conta de não haver um meio do robô se equilibrar para fazer a rampa descer de maneira “mais leve”, o único modo que pensamos para acabar com essa dificuldade foi usar o sistema de angulação que o próprio spike possuí, porém o mesmo não reconhecia a inclinação e não executava a programação.

A maneira que econtramos para o robô passar deste “novo desafio” foi fazer-lo ir devagar até um certo ponto da rampa e depois dar ré até que abaixe, podendo dar sequência as outras missões.

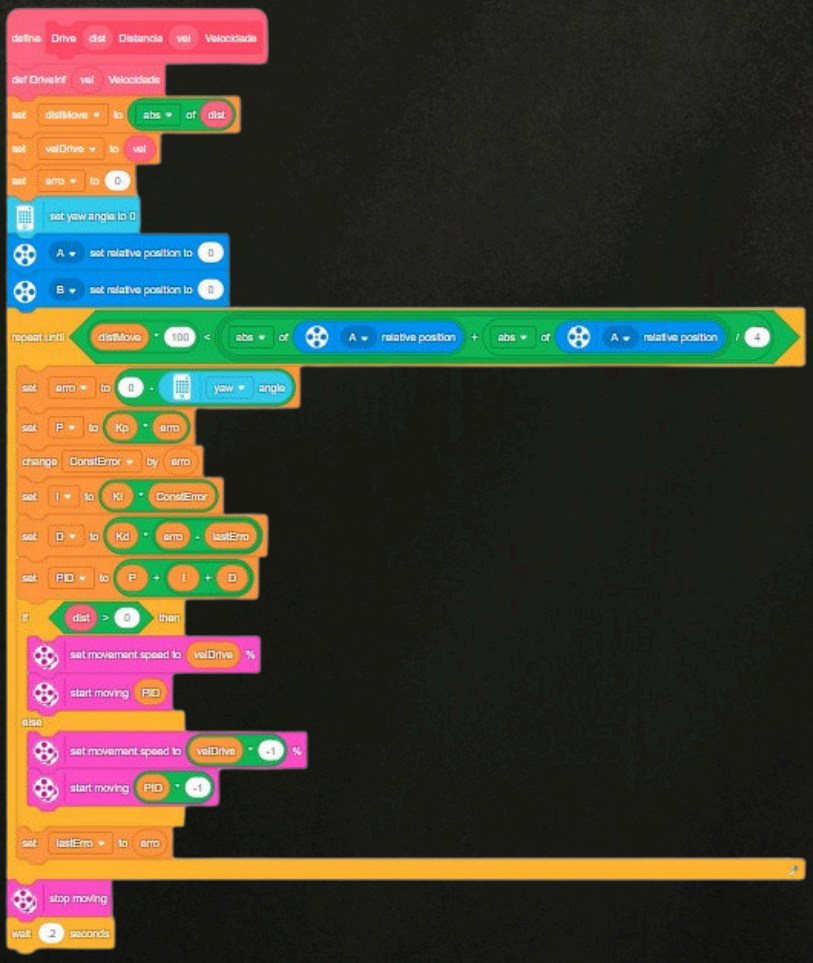
PROGRAMAÇÃO

Recursividade de código

DRIVE PID

Distância que o robô deve andar bloco valor absoluto para não ficar em valor absoluto;

- Valor da PROPORCIONAL do PID;
- Valor da INTEGRAL do PID;
- Valor da DERIVADA do PID;
- Redefine o valor do giroscópio para zero, pois assim, caso o robô tente se mantar no ângulo 0 ele irá para frente exatamente reto;
- Condição de parada baseada na posição dos motores dada pelo hub;
- Condição para o robô ir para frente ou para trás (caso o valor da distância seja negativo).



PROGRAMAÇÃO

Recursividade de código

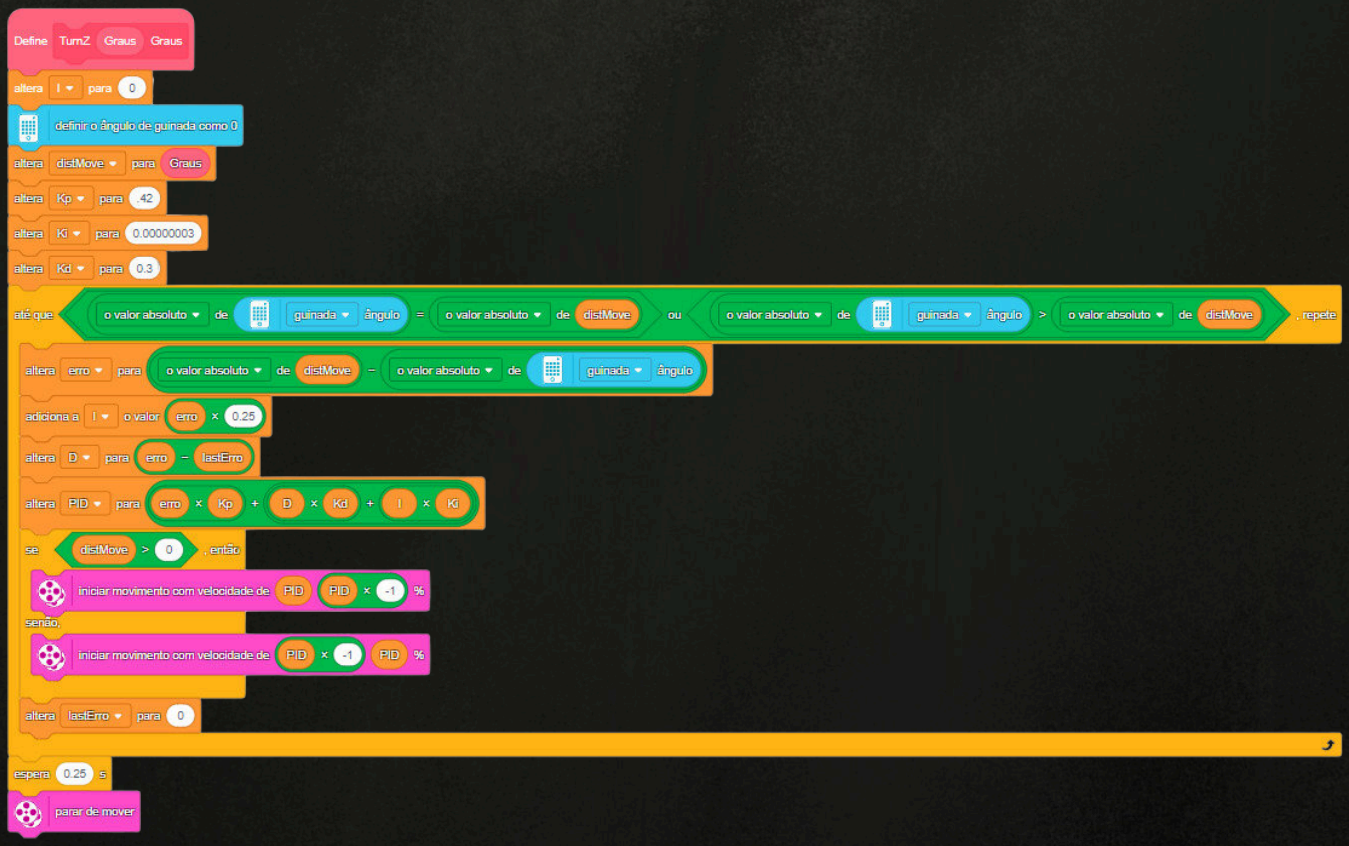
TURN Z

- Define a quantidade de graus que o robô deve girar;

- Condição para parar de girar;

- Define se o robô irá girar para a esquerda ou para a direita;

- Colocar o valor do PID na definição dos motores para que o robô gire no próprio eixo.



PROGRAMAÇÃO

Recursividade de código

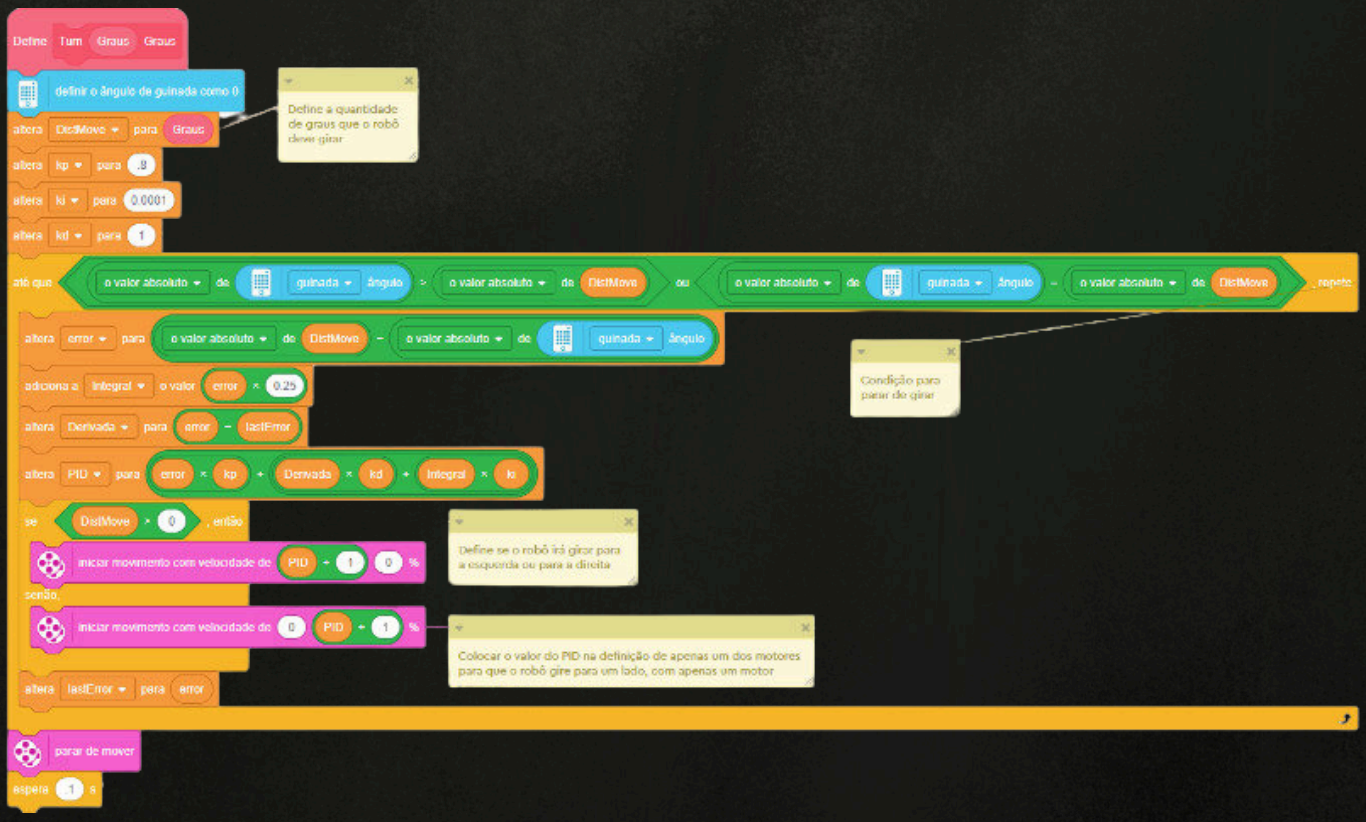
TURN

- Define a quantidade de graus que o robô deve girar;

- Condição para parar de girar;

- Define se o robô irá girar para a esquerda ou para a direita;

- Colocar o valor do PID na definição de apenas um dos motores para que o robô gire para um lado com apenas um motor.



PROGRAMAÇÃO

Recursividade de código

SEGUIDOR DE LINHA

- Condição da "ordem da linha", fazendo com que o robô leia a linha para um lado ou para o outro;

- Condição para ler a linha para frente ou para trás.



PROGRAMAÇÃO

Recursividade de código

DEFINIR DADOS DO SEGUIDOR DE LINHA

- Condição da "ordem da linha", fazendo com que o robô leia a linha para um lado ou para o outro;

- Condição para ler a linha para frente ou para trás.



PROGRAMAÇÃO

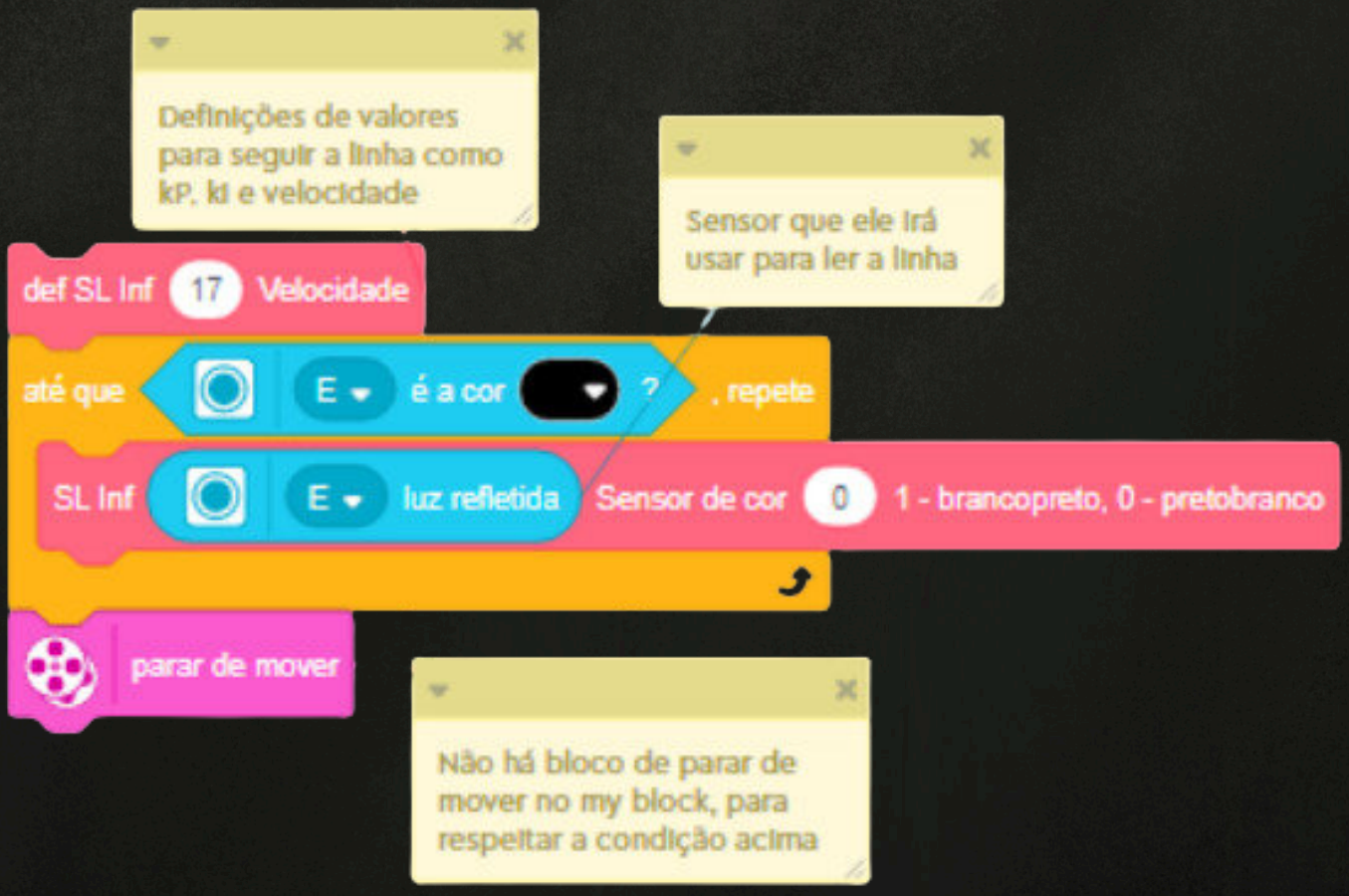
Recursividade de código

UTILIZAÇÃO DO SEGUIDOR DE LINHA

- Definições de valores para seguir a linha como kP, kD e velocidade;

- Sensor que ele irá usar para ler a linha;

- Não há bloco de parar de mover no my Block, para respeitar a condição acima.



PROGRAMAÇÃO

Recursividade de código

IDENTIFICAR CORES DOS POLUENTES (DIREITA)

- Utilizar segue linha até que o sensor "c" e o sensor "d" leiam ciano ou azul;

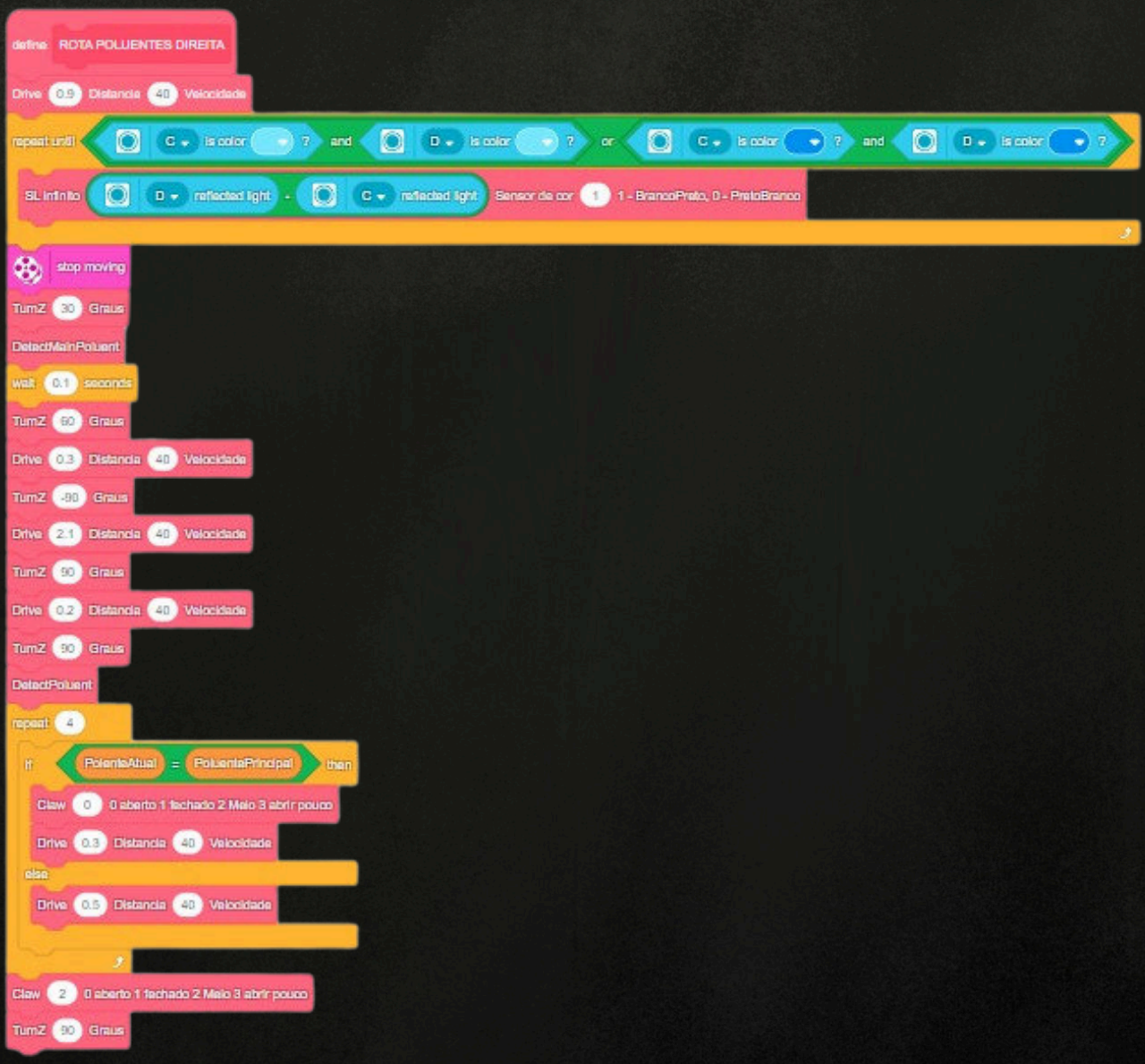
- Parar de mover e girar 30° no sentido horário, utilizar função "DetectMainPoluent";

- girar 60° no sentido horário e seguir em direção aos poluentes da direita;

- Girar 90° no sentido anti-horário e seguir até o último poluente;

- girar 180° no sentido horário, e utilizar a função "DetectPoluent";

- Comparar a cor da variável "Poluente atual" com a variável "PoluentePrincipal" repetir este processo 4 vezes;



PROGRAMAÇÃO

Recursividade de código

IDENTIFICAR CORES DOS POLUENTES (ESQUERDA)

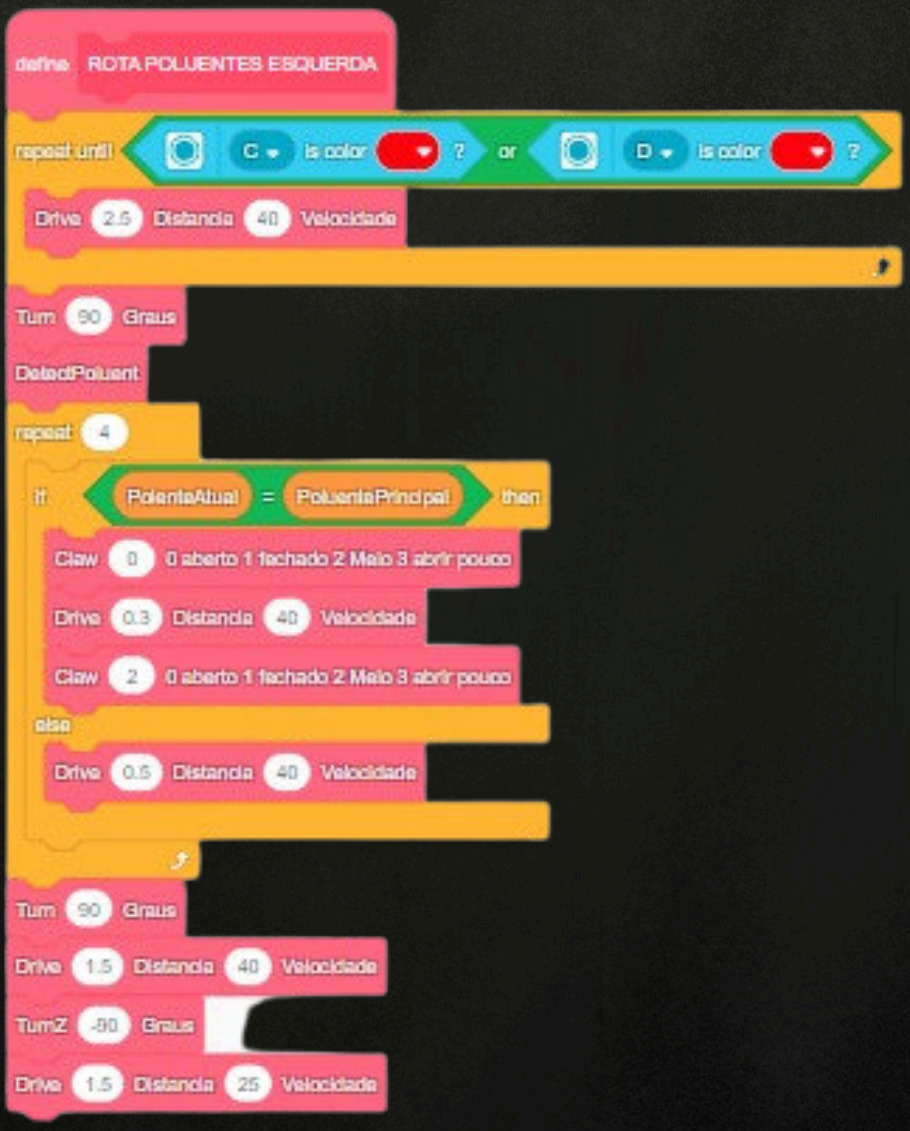
- Repetir a função "Drive" até que os sensores "c" ou "d" identifiquem a cor vermelha;

- Girar 90° no sentido horário e utilizar a função "DetectPoluent"

- Repetir 4 vezes a função abaixo:

- Se a variável "PoluenteAtual" for igual a variável"PoluentePrincipal" derrubar a plaqueta;

- Se a variável"PoluenteAtual" não for igual a variável "PoluentePrincipal" seguir reto.



PROGRAMAÇÃO

Recursividade de código

RECONHECER ÁRVORES

- Reseta o temporizador;

- Altera a variável “IsRightAlto” para 0;

- Altera a variável “BrakeDT” para 0;

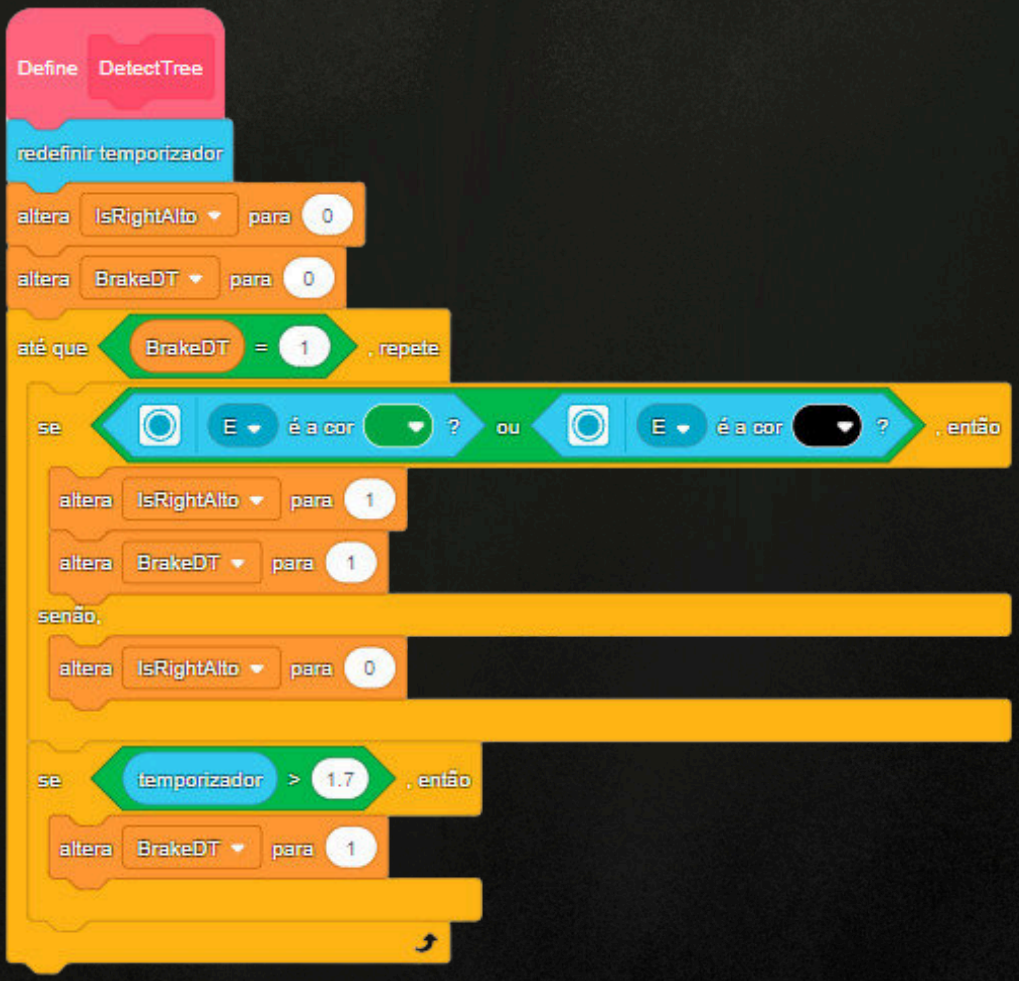
- Repete a função abaixo até que a variável “BrakeDT” seja igual a 1;

- Adiciona a condição SE ENTÃO, SENÃO;

- Verifica se o sensor superior reconhece a cor verde, ou preta, e então altera a variável “IsRightAlto” e a variável “BrakeDT” para 1;

- Se for falso, altera a váriavel “IsRightAlto” para 0;

- Se o temporizador for maior que 1,7, então alterar a váriavel “BrakeDT” para 1.



PROGRAMAÇÃO

Recursividade de código

RECONHECER POLUENTE PRINCIPAL

-Adiciona condição SE, ENTÃO;

- Se o sensor superior identificar a cor vermelha, alterar a variável “PoluentePrincipal” para “V”;

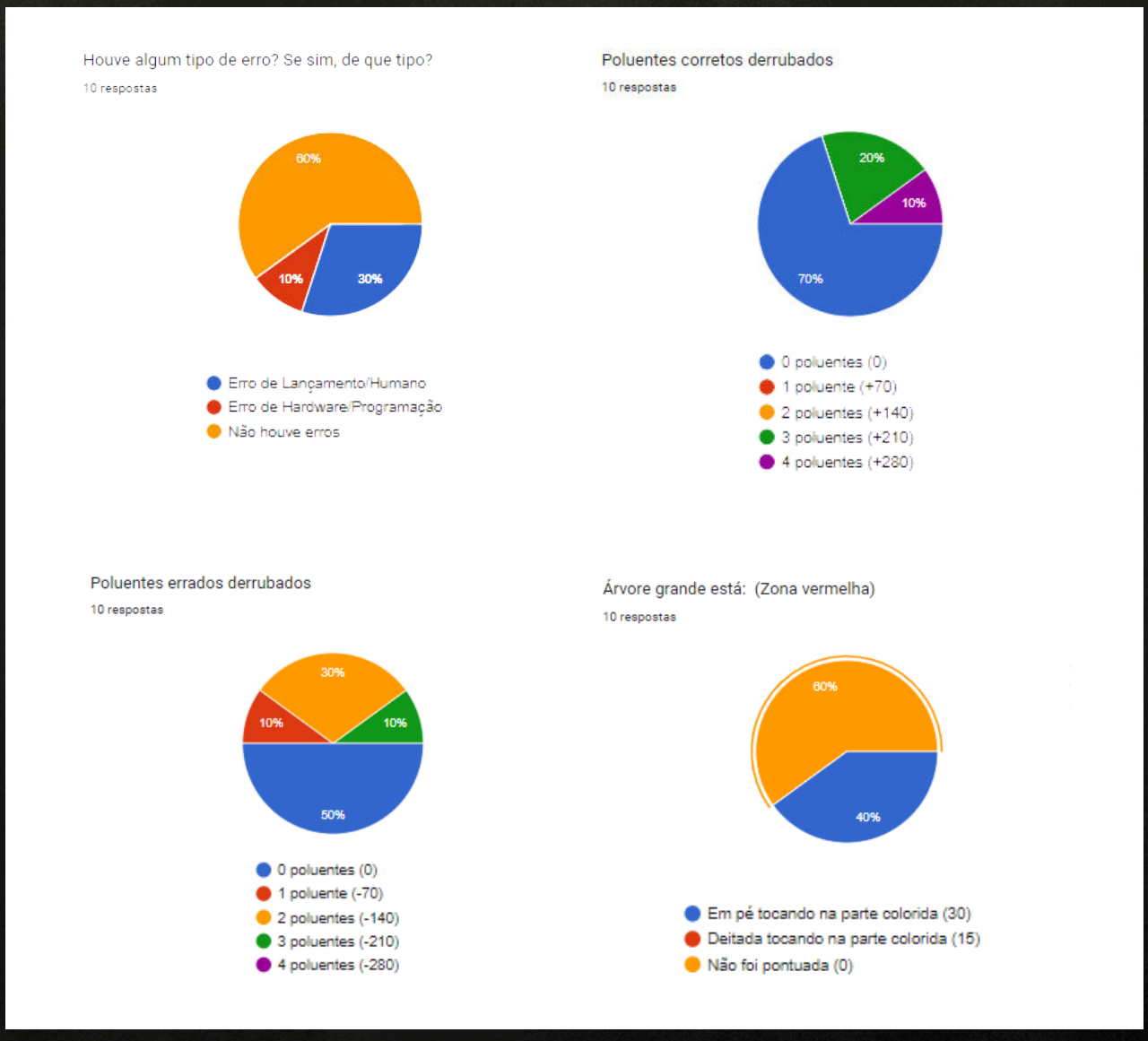
- Se o sensor não identificar a cor vermelha, altera PoluentePrincipal para P;



TESTES

Documentação de Desempenho

- Uso de formulários gráficos vinculados a planilha Excel.
- Acompanhamento visual e organizado do desempenho ao longo do tempo.
- Identificação de padrões, pontos fracos e estratégias eficazes.
- Atualização de dados a cada round para visão clara do progresso e áreas de melhoria.



SOFTWARES

UTILIZADOS NO PROCESSO

RECURSOS HUMANOS

Para que a equipe de engenharia do robô pudesse fazer um trabalho colaborativo mesmo distante do local de treino e desenvolvendo habilidades para o mercado de trabalho e nosso desenvolvimento humano, utilizamos a ferramenta em nuvem OneDrive. Dessa forma, temos um registro contínuo de tudo que fizemos no torneio além do caderno de engenharia que também fica disponível na nossa pasta compartilhada



DIVULGAÇÃO

Como estamos programando em bloco o robô, decidimos armazenar nossa programação no GitHub, que é um serviço baseado em nuvem que hospeda códigos e permite os desenvolvedores colaborem e façam mudanças em projetos compartilhados enquanto mantêm um registro detalhado do seu progresso. Disponibilizamos o link do repositório no nosso Instagram.

Como ferramenta para divulgar o nosso design de robô, nós utilizamos a plataforma do BrickLink, criadora do Studio 2.0 que utilizamos para modelar, dessa forma podemos deixar o modelo do nosso robô disponível para download e visualização online. Disponibilizamos o link do modelo no nosso Instagram.





CONTATOS

(21) 98518-6299

uniaoalpha41@gmail.com

@alphatbr

Firjan SESI/SENAI - Jacarepaguá

FEITO PELA

EQUIPE ALPHA TBR