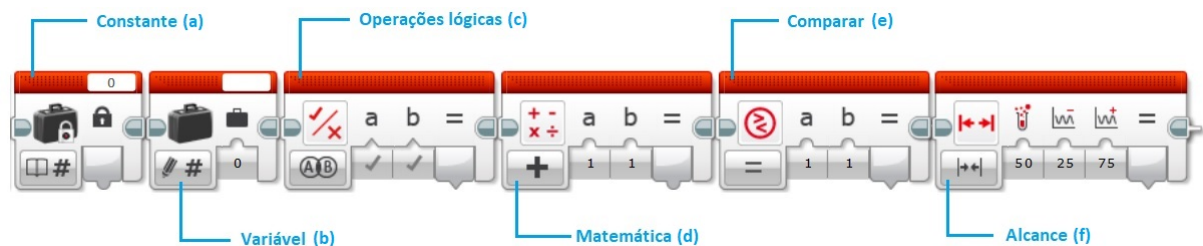


3.4.3 A paleta seletora de operações de dados

A paleta seletora de operações de dados é composta por nove blocos dos quais selecionou-se seis, veja figura 18, que serão usados no presente trabalho. Todas as operações lógicas e numéricas passam por aqui, seja na definição de uma constante, na declaração de uma variável, no retorno de uma função, numa comparação de valores, enfim trata-se de um componente essencial para elaborar programas mais avançados.

Figura 18 - Blocos de operações de dados



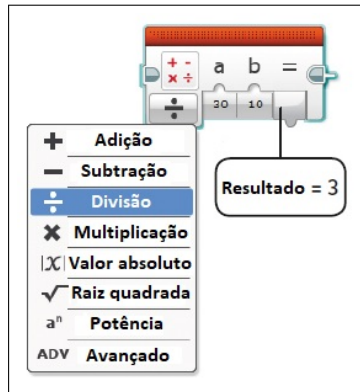
Fonte: O autor, 2018.

Sabendo utilizar os blocos de operações de dados em conjunto com o fio de dados, os alunos podem construir programas bem interessantes. Por exemplo, é possível fazer o EV3 combinar e processar valores de sensores para que possam ser usados como entrada para outras ações. O emprego do bloco *alcance* permite programar o robô para realizar uma ação somente se o retorno da luminosidade captada pelos sensores estiver dentro da faixa de leitura ou fazer algo somente quando dois sensores são acionados ao mesmo tempo através da aplicação do bloco de *operações lógicas*, em vez de executar ações estritamente pré-programadas. Com o bloco *Matemática* eles poderão verificar na prática como os cálculos numéricos, estudados durante o ensino fundamental, são aplicados nos programas gerando uma ação. Por exemplo, o robô pode calcular a distância que deve percorrer com base em uma leitura do sensor.

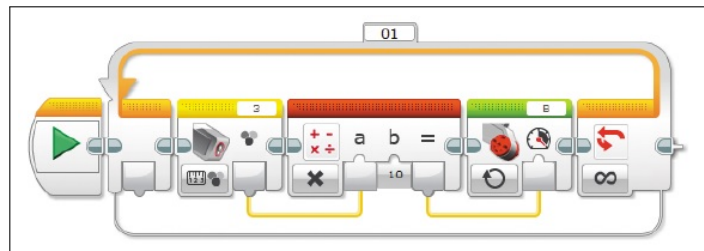
O bloco *Matemática*, conforme dito, veja a Figura 19(a), permite que o EV3 faça operações aritméticas, como adição, subtração, multiplicação, e divisão. O uso é simples, digita-se dois números nos campos *a* e *b* no bloco, em seguida usa-se o seletor de modo para escolher qual operação deve ser aplicada a eles (no caso de divisão, *a* será dividido por *b*). Um fio de dados apresenta o resultado. Em vez de digitar números para *a* e *b*, de acordo com a situação, pode-se fornecer esses valores através de um fio de dados.

O programa na figura 19(b) mostra o desempenho do bloco *Matemática* sendo usado multiplicando a medição do sensor de cor para controlar a velocidade do motor B. O valor do sensor de cor (um número entre 0 e 7) não é adequado para controlar a velocidade porque isso faria o motor girar muito lentamente (velocidade associada a potência 7 na melhor das hipóteses). O bloco *Matemática* multiplica o valor do sensor

Figura 19 - Um fio de dados passa o valor medido no sensor de cor para o bloco *Matemática* que por sua vez aumenta esse valor em 10 vezes. Por último, outro fio de dados transfere o resultado para a o motor B



(a) O bloco *Matemática*



(b) Potência do motor em função da cor detectada pelo sensor.

Fonte: THE LEGO[®] MINDSTORMS[®] EV3 DISCOVERY BOOK - Laurens Valk - 2014.

por 10 e envia o resultado (um número entre 0 e 70) para a configuração de potência do bloco motor. Como resultado, o motor se move com velocidade 10 para preto, velocidade 20 para azul e assim por diante, até a velocidade 70 para marrom (VALK, 2014).

Além dos operadores básicos descritos na Figura 19(a) o bloco *Matemática*, disponibiliza na aba ADV, os operadores *negar*, *limite mínimo*, *ceil* (teto), *arredondar*, *log*, *ln*, *sen*, *cos*, *tan*, *asen*, *acos* e *atan*. Atividades planejadas com esse bloco favorecem o ensino/aprendizagem na ótica da construção de conhecimento ou aplicação contextualizada de inúmeros conteúdos curriculares referendados pela BNCC. Pensando nisso, propõe-se no Apêndice A atividades adaptadas da obra “The LEGO[®] Mindstorms[®] EV3 Discovery Book” de Valk (2014).

4 AVANÇOS E CONQUISTAS DA ROBÓTICA EM MACAÉ

A Matemática é um método geral do pensamento aplicável a todas as disciplinas e desempenha um papel dominante na ciência moderna.

Antônio Monteiro

A primeira experiência com a Robótica em Macaé levando-se em conta a empolgação, comprometimento e produtividade dos alunos fomentou a possibilidade de expandir o campo de atuação dos estudantes para além do espaço escolar. Pensou-se então na possibilidade de mostrar a produção acadêmica vislumbrando a socialização de conhecimento, troca de experiências, apresentação dos métodos e avaliação do ensino, dando a oportunidade para que os discentes pudessem participar de eventos científicos como a Olimpíada Brasileira de Robótica (OBR) e o torneio Juvenil de Robótica (TJR), chancelados por diversas universidades do Brasil e que acontecem em diversas instituições dentro ou fora do município de Macaé. Essas competições desafiam os discentes a resolverem os mais diversos tipos de problemas envolvendo ciência e tecnologia. Almejando alcançar esse objetivo apresenta-se nesse capítulo: as regras das competições, os pressupostos para a construção e automação de um “robô competidor”, as estratégias de eficiência para superar as situações-problema pertinente ao desafio e a organização e desenvolvimento das aulas. Divulga-se ainda os resultados das investigações envolvidas nesse processo e um histórico dos resultados.

A Olimpíada Brasileira de Robótica é um evento realizado no Brasil para difundir a Robótica entre jovens estudantes do ensino fundamental e médio. Sua criação deve-se a iniciativa de pesquisadores dessa área de conhecimento de instituições como ITA, UFRN, UNESP, FEI e FURG.

Atualmente a OBR oferece aos discentes duas modalidades de participação: modalidade teórica e modalidade prática. Na modalidade teórica, os alunos fazem provas objetivas com o intuito de testarem seus conhecimentos e obterem novas informações acerca do mundo robótico. As questões são concebidas de forma interdisciplinar envolvendo as seguintes áreas de conhecimento: Linguagem, Matemática, Ciências da Natureza, Ciências Humanas e Robótica. A OBR elabora, a cada ano, uma lista estruturada de assuntos objetivando direcionar os participantes nos seus estudos para as provas. Os conteúdos são baseados na Base Nacional Comum Curricular (OBR, 2018a). Essa maratona de conhecimento é uma boa oportunidade para os jovens verificarem como a Matemática está inserida nessa área científica de ampla expansão, comprovada pelo histórico de participações de equipes inscritas para essa competição.

As provas são divididas em seis níveis de escolaridade distribuídos entre os ensinos

fundamental e médio, da maneira que segue na Tabela 4.

Tabela 4 - Distribuição da modalidade teórica da OBR

Níveis	Escolaridades
0	1 ^o ano
1	2 ^o e 3 ^o anos
2	4 ^o e 5 ^o anos
3	6 ^o e 7 ^o anos
4	8 ^o e 9 ^o anos
5	Ensino Médio

Fonte: <http://www.obr.org.br/modalidade-teorica/fase1/>.

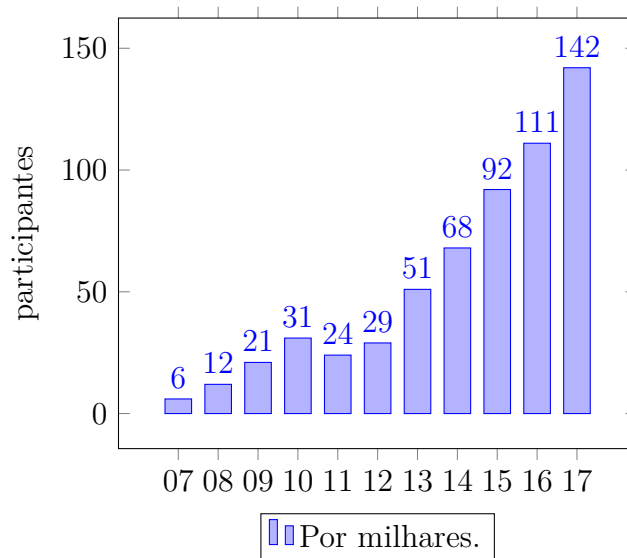
A OBR, em sua modalidade prática, tem como premissa proporcionar ao aluno competidor a oportunidade de aplicar os seus conhecimentos nas áreas de Matemática, Robótica e Computação num grande evento patrocinado por grandes universidades do Brasil. O torneio é dividido em etapas regional, estadual, nacional e internacional. O problema consiste em simular um espaço real de catástrofe onde o resgate dos sobreviventes é realizado por robôs. Segundo o manual de regras e instruções das etapas regionais e estaduais:

Em um ambiente hostil, muito perigoso para o ser humano, um robô completamente autônomo, desenvolvido por uma equipe de estudantes, recebe uma tarefa difícil: resgatar vítimas sem interferência humana. O robô deve ser ágil para superar terrenos irregulares, transpor caminhos desconhecidos, desviar de escombros e subir montanhas para conseguir salvar as vítimas desse desastre, transportando-as para uma região segura onde os humanos já poderão assumir os cuidados. (OBR, 2018a).

Para essa missão o robô deve ser projetado de tal forma que seja capaz de superar áreas irregulares passando por redutores de velocidades; ser capaz de ignorar as discontinuidades de linha (gaps) durante o percurso; contornar escombros (obstáculos), galgar montanhas simuladas por rampas e tentar salvar vítimas vivas e mortas – bolas de isopor prateada e preta – respectivamente. O sucesso dessa missão acontece quando as vítimas resgatadas são colocadas num local específico (nicho) representado por um triângulo isósceles preto. Esse local, caso fosse real, seria o espaço onde humanos poderiam socorrer as vítimas.

Desde a sua criação em 2007, o número de participantes na modalidade prática da OBR vem crescendo substancialmente, como pode ser observado no Gráfico 1, revelando o interesse e a motivação dos discentes em aprender e desenvolver boas práticas em ciências e tecnologia.

Gráfico 1 - Estatística do número de participantes da OBR



Fonte: <http://www.obr.org.br/historico/>. Acesso em: 09 Dez. 2018.

A partir do gráfico, nota-se um crescimento exorbitante entre os anos de 2007 e 2017. Esse resultado ratifica o fortalecimento da missão da Olimpíada Brasileira de Robótica, que juntamente com o apoio dos órgãos governamentais, Ministério da Ciência e Tecnologia (MCT), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e Ministério da Educação em parceria com a Fundação Nacional de Desenvolvimento da Educação (FNDE/MEC), vem mantendo o compromisso estabelecido em edital de: “atuar como instrumento para a melhoria dos ensino fundamental e médio, bem como identificar jovens talentosos que possam ser estimulados para carreiras técnico-científicas”. (OBR, 2018, 2018).

De todas as tarefas necessárias para cumprir a missão imposta pela OBR, a mais importante certamente é projetar, montar e programar um robô seguidor de linha, uma vez que a progressão no terreno particularmente preparado depende essencialmente disso. Muitos erros, sejam de projeto mecânico ou computacional, ocorrem durante os preparativos que antecedem a competição, e são exatamente essas falhas que levam os discentes a investigarem os motivos desses defeitos. A busca constante da melhor solução para os erros apresentados é que mantém os alunos ativos, concentrados e motivados numa prática científica interdisciplinar. Percebe-se que essas ações permitem aos alunos exercitar a metodologia empregada nessa pesquisa: *descrição, execução, reflexão e depuração*.

Divide-se a solução autônoma do robô em rotinas (procedimentos) que mais tarde serão encapsuladas no programa principal. Assim, os estudantes deverão ser capazes de fazer com que o robô:

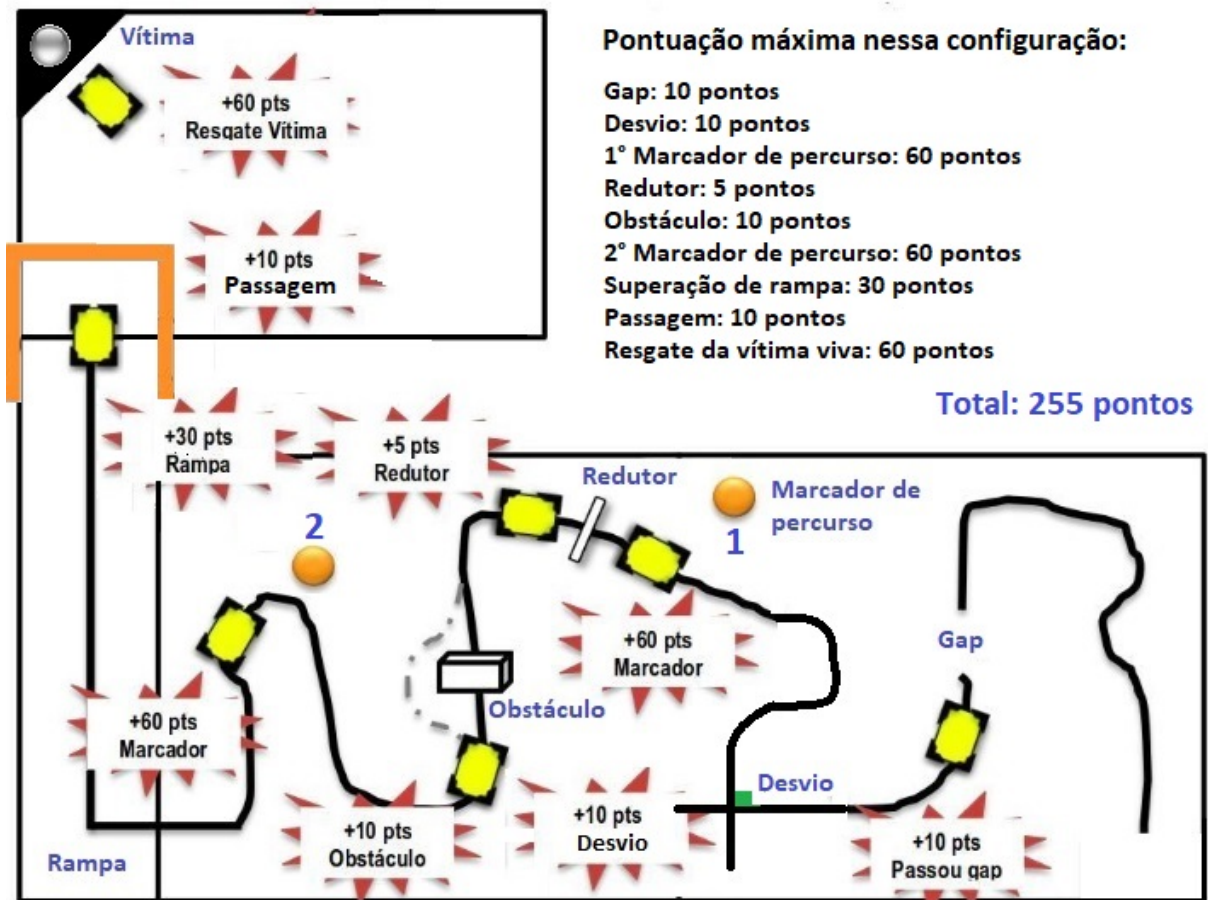
1. Siga a linha demarcada em preto;
2. Retome a linha caso haja uma falha de progresso no gap;
3. Desvie de obstáculos;
4. Transponha redutores de velocidade;
5. Desvie no verde em cruzamento, quando houver;
6. Supere passagem (portal quadrangular com 25 cm de comprimento por 25 cm de altura);
7. Identifique a rampa, se necessário;
8. Perceba a entrada da sala de resgate;
9. Identifique o nicho (triângulo preto);
10. Realize as manobras de resgate;
11. Descarte corretamente as vítimas no nicho.

A Figura 20 mostra uma representação de todos os componentes presentes no desafio *Rescue Line* da OBR e a pontuação correspondente à cada elemento superado ou tarefa executada com sucesso.

O Torneio Juvenil de Robótica (TJR), assim como a OBR, é um evento gratuito de abrangência nacional que ocorre desde 2009, para difusão da Robótica no ambiente escolar da Educação Básica até a Educação Superior. A competição conta com inúmeros desafios, planejados para o estudo e desenvolvimento da robótica para robôs autônomos. Os desafios promovidos são do tipo luta de robôs, resgate, cabo de guerra, dança e registro multimidiático. A categoria resgate conta com duas modalidades: *Resgate de Alto Risco* e *Resgate no Plano*.

O desafio *Resgate de Alto Risco* é bem semelhante ao proposto pela OBR. O robô deve seguir uma linha, superar obstáculos, descontinuidades e rampas. Não existe a superação da passagem, os desvios de encruzilhada e os marcadores. Em contrapartida, O robô deve ser capaz de resgatar o objeto alvo em qualquer posição sobre a linha preta (exceto na rampa) e descartá-lo num nicho representado por um quadrado preto $12\text{cm} \times 12\text{cm}$ localizado em qualquer posição na linha. Além disso, a condição de encerramento da missão não é o resgate da vítima, como na OBR, mas o retorno ao “espaço seguro”

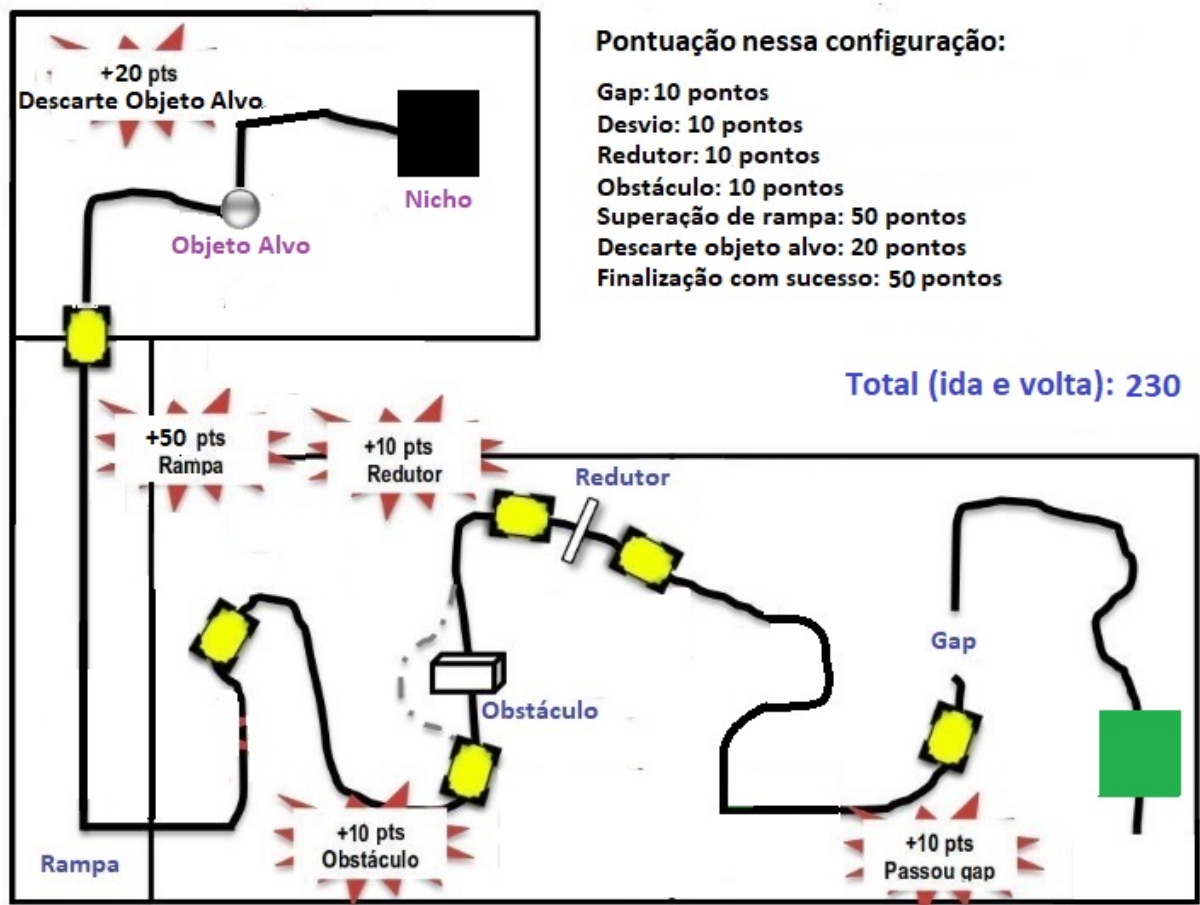
Figura 20 - Arena de competição da etapa regional/estadual da OBR



Fonte: Manual de regras e instruções etapa regional/estadual - versão 1 - OBR - 2018.

representado por um quadrado verde do mesmo tamanho que o nicho podendo ser colocado em qualquer posição sobre a linha (exceto na rampa). Uma possível configuração desses elementos pode fazer com que o robô tenha que percorrer toda a arena de competição num percurso de ida e volta, conforme mostrado na Figura 21. O desafio *Resgate no Plano* é uma versão simplificada do *Resgate de Alto Risco* cuja diferença principal é não possuir piso em elevação.

Figura 21 - Arena de competição TJR



Fonte: Adaptado de OBR (2018a) e (SILVA, 2018).

4.1 O robô seguidor de linha

Um robô seguidor de linha é um tipo de robô que segue uma faixa de linha, utilizando sensor de cor ou reflexão. Este é um dos problemas mais comuns em robôs industriais e é uma das aplicações mais úteis, porque permite que o robô se mova de um ponto para outro para cumprir suas tarefas. Para que o robô siga a linha, existem alguns tipos de controle que podem ser utilizados, como o controle On-Off, proporcional, propor-

cional integral derivativo, entre outros. Nesta pesquisa, devido ao nível de escolaridade dos estudantes, aplica-se o controle On-Off.

4.1.1 Controlador On-Off com um sensor

Controlar um robô pelo método On-Off é certamente uma das formas mais simples e de custo bem baixo, sendo empregada tanto em sistemas industriais quanto domésticos. Esse método só permite duas posições para o objeto final de controle (sensor de cor), isto é, sistema ligado ou desligado, ou em outras palavras, fluxo de programa aberto ou fechado (OLIVEIRA⁷, 1999) citado por (GUADAGNIN, 2014). Esse autor alega que o controle em duas posições apenas é elementar e econômico, sendo portanto aplicado no dia a dia. Como exemplo, podemos citar os controles de fornos elétricos pequenos, fornos de secagem, controladores de pressão etc.

Na sequência, aplica-se esse método na programação do robô seguidor de linha usando os blocos de controle de fluxo *ciclo* e *comutação* e o bloco de ação *mover direção*. Para essa tarefa surge de imediato a necessidade de aplicar alguns conceitos matemáticos num procedimento muito conhecido pelos alunos que fazem robótica de competição, a calibração dos sensores de cor. Esse procedimento é importante porque a chance da luz ambiente do local de competição ser diferente da luz ambiente do local de treino das equipes é grande. Essa variação de luminosidade pode alterar significativamente a maneira como o sensor de cor faz a leitura, passando dados incorretos para a programação. Nas competições, o juiz responsável pela marcação da pontuação das equipes, permite que a calibração seja feita em vários pontos da arena no tempo de 2 minutos (OBR, 2018b).

Mas, a propósito, o que é calibrar os sensores de cor no modo comparar a intensidade de luz refletida? O sensor de cor do LEGO[®] Mindstorms[®] EV3 pode medir o brilho de uma cor quando acessado na plataforma de programação ou no próprio controlador EV3, selecionando a *Port View* no modo de intensidade de luz refletida. Por exemplo, o sensor pode ver a diferença entre uma superfície de cor branca, cinza ou preta através do retorno do brilho da luz refletida. Essa intensidade de luz refletida é medida em porcentagem que varia de 0% (refletividade muito baixa: superfície escura) a 100% (refletividade muito alta: superfície clara). O sensor é capaz de retornar valores em porcentagem associados ao tipo de superfície que está incidindo: branco, cinza e preto. O brilho do sensor será alto tanto quanto for a superfície clara e baixo, caso contrário. Assim, espera-se que a superfície branca reflita muita luz, resultando num valor próximo ou igual a 100% e abaixo de 10% no caso absorção de luz (superfície escura, possivelmente preta). Resumindo, uma

⁷ OLIVEIRA, Adalberto L. D. L. Fundamentos de controle de processo. 1999. Trabalho realizado em parceria SENAI / CST (Companhia Siderúrgica de Tubarão), Vitória, 1999.

superfície escura não reflete muita luz, resultando em uma medição provavelmente abaixo de 10%. Uma superfície clara geralmente tem como resultado uma medição superior a 60%.

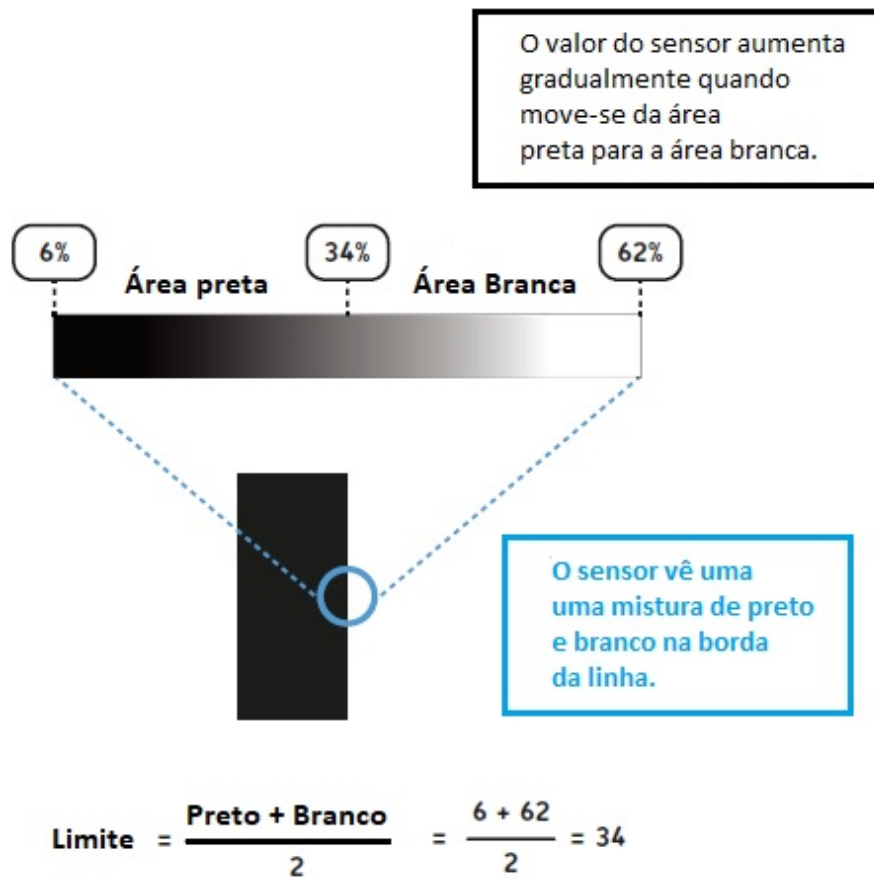
Para verificar esses valores espera-se que o aluno já esteja familiarizado com os dois modos de obter essa aferição, ou diretamente pelo bloco controlador EV3 ou através da plataforma. Nesse caso, o mesmo deverá estar conectado ao computador pelo cabo USB, bluetooth ou wi-fi. Suponha que a medição foi feita e que os valores encontrados foram de 6% para preto e 62 % para branco. O robô ao seguir a linha preta seguirá, na verdade, a borda entre o branco e o preto, ou seja, a região cinza, para isso ele precisa saber qual será o valor limite onde a leitura da programação considera o que é preto e o que é branco. Suponha ainda que pode-se arrastar o sensor do robô com as mãos da área preta para a branca sempre observando a medição da luz refletida. Nota-se que o valor aumenta gradualmente de 6% para 62%. Quando o sensor vê parcialmente a linha preta e parcialmente a linha branca, o valor será entre esses dois extremos, como se o sensor estivesse vendo uma superfície cinza. Pode-se usar essa medida mais detalhada para dizer ao robô qual medida será escolhida como limite onde a partir daí, teremos o que é considerado branco e o que é considerado preto. Em outras palavras, a programação atribuirá os tons de cinza claro como branco e os tons de cinza escuro como preto, veja Figura 22.

Tendo em vista o conteúdo curricular vê-se que na aplicação do método On-Off para controlar um robô seguidor de linha é de suma importância a calibração do sensor de cor no modo comparar a intensidade de luz refletida. Alguns conceitos matemáticos devem ser conhecidos e utilizados nessa tarefa. De fato, conforme visto na figura 22, a média aritmética é usada para obter a calibração no limite de valor 34, a porcentagem é usada na aferição da intensidade de luz refletida, seja diretamente pelo bloco controlador ou pelo programa e a comparação de números, considerando os valores menores ou maiores do que o limite. Essas comparações definem na programação os tons de cinza claro e cinza escuro que darão a direção do robô.

Percebe-se que ao propor aos alunos a execução da rotina de calibração do robô, valoriza-se a aplicação dos conteúdos numa aula pragmática. Além disso, pode-se observar a aplicação do método de ensino em espiral, pois os discentes tem a oportunidade de rever conteúdos de séries anteriores e até mesmo reforçar o aprendizado caso esse não tenha ocorrido.

De posse da média aritmética calculada no processo de calibração o aluno tem o que precisa para começar a desenvolver o algoritmo do robô seguidor de linha na plataforma de programação LEGO[®] Mindstorms[®] EV3. Convém, tendo como referência as boas práticas de programação, orientar os alunos a escreverem um pseudocódigo (algoritmo em linguagem comum) para documentar a ideia. Esse algoritmo fica da forma que segue:

Figura 22 - Na borda da linha, o sensor vê tons de cinza



Fonte: Learning LEGO[®] Mindstorms[®] EV3 Discovery Book –
Laurens Valk - 2014.

Início

Se sensor > média (34) **então faça**

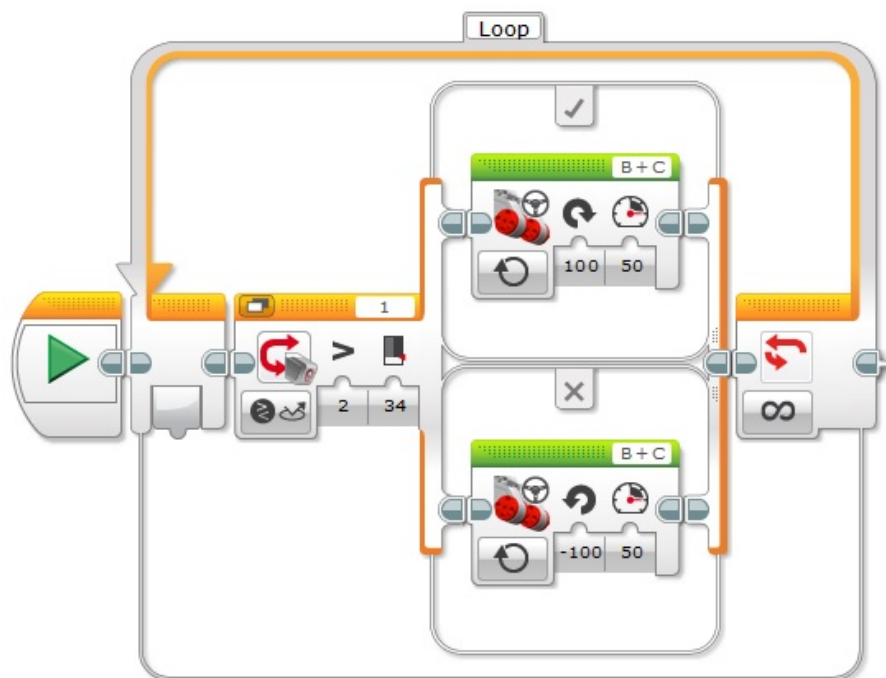
Gire o robô para a direita **senão**

Gire o robô para a esquerda

Fim

Na linguagem EV3 programming fica da forma que é mostrado na Figura 23:

Figura 23 - Algoritmo “escrito” na plataforma LEGO[®] Mindstorms[®] EV3



Fonte: O autor, 2018.

4.1.2 Controlador On-Off com dois sensores de cor

O uso de um sensor de cor apenas, seja no modo de medida da intensidade de luz refletida ou no modo de medida de cor, para progredir no terreno de competição seguindo uma linha preta no chão, poderá colocar os discentes diante de uma problemática muito comum: E se o robô encontrar uma falha na pista? Essa falha, conhecida por *gap*, fará com que o robô perca a orientação em linha pois, no momento em que o sensor de cor não conseguir contato com a fita preta, passará a enxergar apenas o branco no terreno sendo levado a rodopiar sem rumo, caracterizando a falha de progresso. Existem muitas soluções

mecânicas e computacionais para resolver essa falha de progresso acarretada pelo *gap*, os alunos podem agregar mais componentes ao projeto de robô adquirindo mais sensores, decisão custosa, pois sensores de cor tem um custo financeiro alto ou podem trabalhar com dois sensores, que é o mínimo possível, e mais econômico.

A opção pelo uso de dois sensores exige um desenvolvimento em lógica de programação mais elaborado do que quando se usa um sensor apenas, pois a orientação no espaço está atrelada a dois componentes. Para complicar ainda mais, elevando o grau de exigência do projeto do robô seguidor de linha com dois sensores, o que pode ser bastante instigador para os discentes do ponto de vista do desafio, é a presença dos sinalizadores de desvio, quadrado verde (tamanho $2,5\text{cm} \times 2,5\text{cm}$) que são colocados no trajeto sempre que houver um cruzamento de linha, indicando a direção que o robô deve seguir. Esse sinalizador por ser de cor verde, retorna uma numeração de leitura do sensor muito próxima da leitura de cor cinza. Essa proximidade de numeração pode ser um complicador dependendo da lógica de programação desenvolvida. Mais adiante apresenta-se a solução encontrada pelos discentes com base no desdobramento do método On-Off para um sensor.

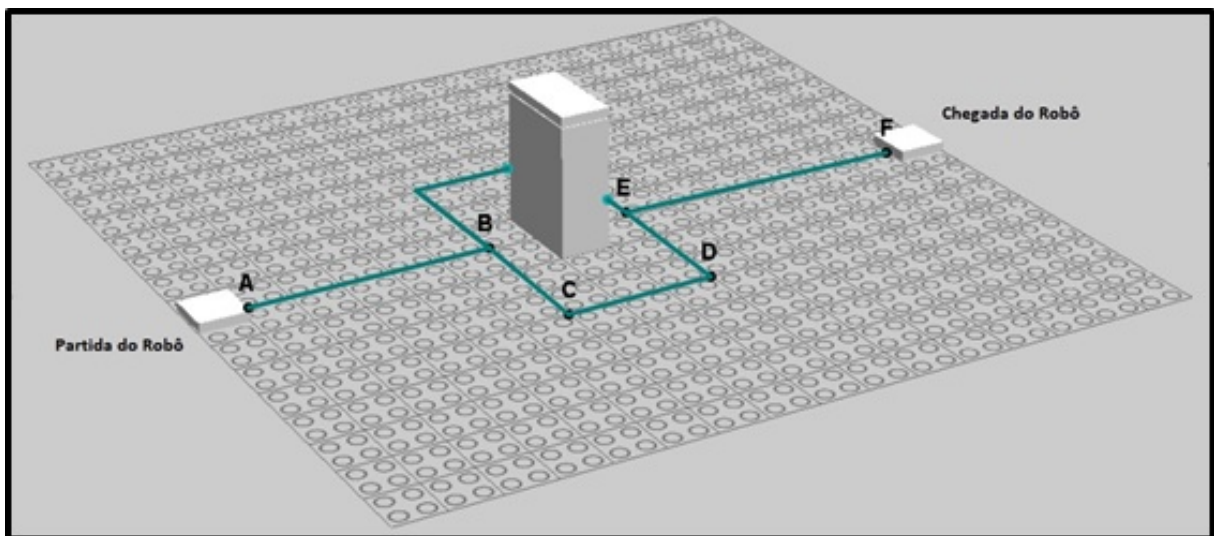
4.2 Descrição da situação-problema (Matemática no desvio do obstáculos)

Durante o percurso em linha no desafio do resgate, seja na Olimpíada Brasileira de Robótica ou no Torneio Juvenil de Robótica, o robô pode se deparar com um ou mais obstáculos. O problema a ser resolvido consiste em interromper momentaneamente o robô da tarefa de seguir linha e fazê-lo contornar o obstáculo, retornando imediatamente a linha, tão logo a manobra de desvio seja concluída. Perceba que basicamente deve-se programar o robô para contornar um obstáculo o que pode ser feito de três maneiras distintas. A eficiência dessa manobra e o seu grau de dificuldade aumenta à medida que são exigidos conhecimentos de diferentes anos de escolaridades. Numa turma de 6º ano ou 7º ano, recomenda-se o desvio pelo método que o autor chama de semiperímetro de quadrado ou retângulo, já nos anos finais do ensino fundamental, pode-se aplicar a manobra triangular ou por semicircunferência. O tempo de execução do desafio é um fator importante para o sucesso da missão haja vista que se trata de uma competição. Portanto, espera-se que os alunos analisem e escolham, dentre os três possíveis métodos propostos, aquele mais eficiente. Apresenta-se pormenor toda a sequência que os alunos podem desenvolver para tomar essa decisão.

4.2.1 A manobra semiperímetro

A seguir, a Figura 24 mostra a representação geométrica desse problema usando o método do semiperímetro. Os pontos A e F indicam, respectivamente, os pontos de partida e chegada do robô. O ponto B a detecção do alvo, condição de parada para o início da manobra de contorno do objeto. Os segmentos de reta AB e EF os percursos antes e depois da manobra. O objetivo é percorrer a linha poligonal $BCDE$. Para fins de exercício, consideramos que o robô está parado no ponto A e deve deslocar até o ponto B . Numa situação global, isto é, na programação completa e finalizada do desafio de resgate em linha, o robô chegaria até o ponto B através da rotina de seguir linha e detectaria o alvo automaticamente pelo sensor de ultrassom, infravermelho ou toque, dependendo da estratégia planejada pelo aluno. Como o nosso objetivo no presente trabalho é mostrar como a Matemática está subjacente na construção e automação de robôs Mindstorms[®] EV3, quebramos o problema em etapas menores facilitando a didática e a compreensão holística da solução do problema.

Figura 24 - O robô contorna o obstáculo por semiperímetro de retângulo



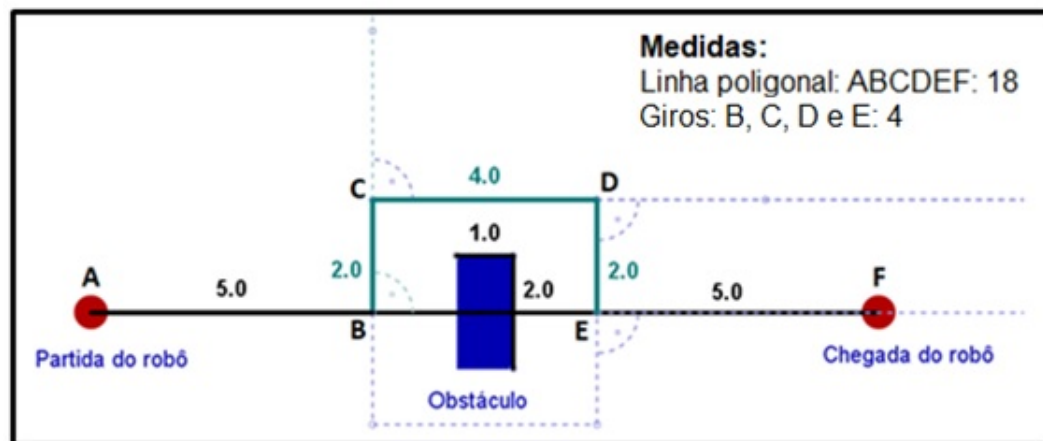
Fonte: O autor, 2018.

Dada a situação-problema, propõe-se aos alunos o estudo da melhor maneira de executar a tarefa, o que pode ser feita a partir do uso de um software de geometria dinâmica. Nada impede que a análise descritiva do problema, um dos componentes do ciclo de ações, possa ser feita numa folha de papel usando instrumentos de medição, lápis e borracha. Um recurso não se sobrepõe ao outro, e muito menos, a descrição do problema depende do objeto de aprendizagem escolhido ou material didático. Optou-se pelo primeiro porque considera-se que os alunos de hoje estão inclusos no que a literatura define de geração Z, ou seja, pessoas que nasceram entre 1990 e 2010. A geração Z é caracterizada

pelos indivíduos que já nasceram na era digital e tem forte influência profissional para as áreas de Ciências da Computação, Tecnologias da Informação, Engenharia e Matemática, por exemplo. Como o estudo de controle e automação de robôs está incluído nesse rol, é de se esperar que tenhamos facilidade na descrição do problema supracitado usando essa ferramenta.

O aplicativo de geometria dinâmica – R.e.C - que foi usado é gratuito e está disponível em <http://www.professores.im-uff.mat.br/hjbortol/car/index.html>, foi desenvolvido pelo professor René Grothmann da Universidade Católica de Berlim, na Alemanha, tem código aberto e funciona em qualquer plataforma (Microsoft Windows[®], Linux, Macintosh[®], etc). Feita a apresentação da situação-problema, o aluno pode representar no R.e.C a descrição do problema da maneira como é mostrado na Figura 25.

Figura 25 - Método do semiperímetro em geometria dinâmica



Fonte: O autor, 2018.

De acordo com a Base Nacional Comum Curricular (BNCC), descritor (EF02MA13) o aluno deverá ser capaz de desenvolver a habilidade de esboçar roteiros a serem seguidos ou plantas de ambientes familiares, assinalando entradas, saídas e alguns pontos de referências. Pode-se sugerir aos alunos que queiram utilizar com recursos digitais, que capturem a vista aérea dos componentes do desafio – robô – ponto de partida – obstáculo – ponto de chegada – para estudo e análise no software R.e.C.

Como hipótese, considera-se que as medidas apresentadas na Figura 25 estão em escala $\frac{1}{5}$, ou seja, cada centímetro no R.e.C corresponde a 5 centímetros reais de deslocamento do robô no exercício. Organizamos as etapas de manobras do robô usando a Tabela 5.

O bloco *mover direção* utilizado na programação permite o deslocamento do robô por *tempo*, por *graus* ou por *rotação*. No modo *tempo*, em segundos, o robô desloca

Tabela 5 - Registro das medidas de cada movimento do robô

Etapas	Manobras	Escala Gráfica	Escala Real
1	AB	$5cm$	$5 \cdot 5 = 25cm$
2	Giro B	-90°	constante
3	BC	$2cm$	$5 \cdot 2 = 10cm$
4	Giro C	$+90^\circ$	constante
5	CD	$4cm$	$5 \cdot 4 = 20cm$
6	Giro D	$+90^\circ$	constante
7	DE	$2cm$	$5 \cdot 2 = 10cm$
8	Giro E	-90°	constante
9	EF	$5cm$	$5 \cdot 5 = 25cm$

Fonte: O autor, 2018.

uma determinada distância no tempo programado. O giro de uma volta pode ser obtido nos modos *rotação* ou *graus*, onde uma rotação corresponde a uma volta no servomotor e, analogamente, a 360° no modo graus. Optou-se por utilizar o modo rotação pela praticidade obtida na aferição da razão *rotação/distância*. Os alunos devem pesquisar a relação entre uma rotação de motor, a distância percorrida e o tempo usando um robô do tipo tanque, mostrado na Figura 26.

4.2.2 A manobra triangular

A figura 27 mostra a representação geométrica para a manobra pelo método triangular. Os pontos A e E indicam respectivamente os pontos de partida e chegada do robô. O ponto B , a detecção do alvo, condição de parada para o início da manobra de contorno do objeto. Os segmentos de reta AB e DE , os percursos antes e depois da manobra. O objetivo assim como o método anterior, consiste em percorrer uma linha poligonal, porém o aluno deverá perceber que houve uma redução de um segmento de reta, pois a linha poligonal de contorno do obstáculo está representada pelos pontos B , C e D . Tal redução poderá influenciar na eficácia do robô em concluir o objetivo, o que poderá ser constatado quando o educando executar a etapa de **reflexão** do ciclo de ações.

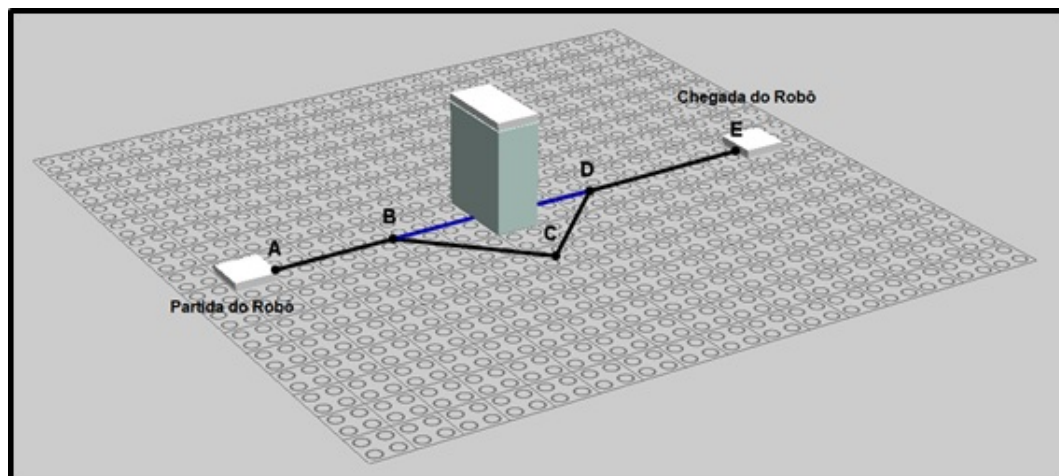
Nesta manobra o aluno deve estudar o ponto limite de parada do robô - distância entre o ponto B e o obstáculo - de maneira que no deslocamento representado pelo segmento de reta BC , o robô passe o mais próximo possível do obstáculo, sem tocá-lo. Conteúdos de geometria analítica como distância entre dois pontos ou distância entre

Figura 26 - Robô com esteiras comumente usado nas competições de robótica



Fonte: O autor, 2018.

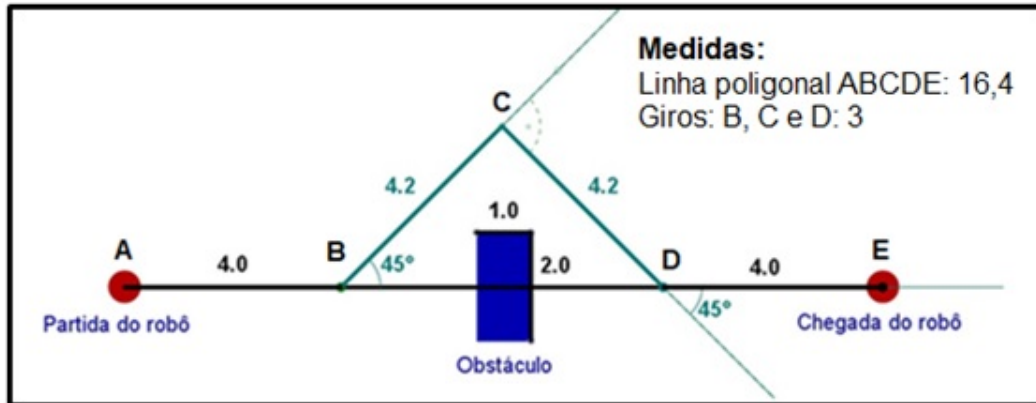
Figura 27 - Modelagem geométrica na plataforma LEGO[®] Digital Design 4.3



Fonte: O autor, 2018.

ponto e reta podem ser explorados pelos discentes. A Figura 28 mostra como fica a *descrição* do percurso do robô em ambiente R.e.C.

Figura 28 - Método triangular em geometria dinâmica



Fonte: O autor.

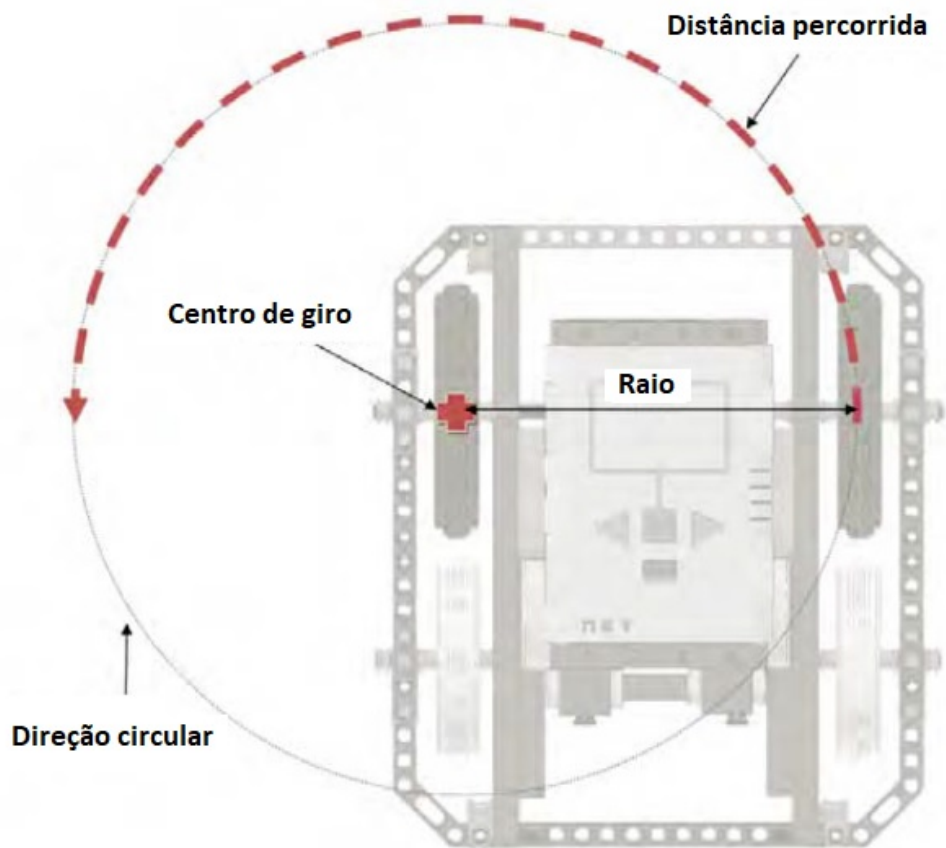
4.2.3 A manobra semicircunferência

Existem duas maneiras diferentes de direcionar um robô em curva: movimentar as duas rodas ou movimentar uma roda enquanto a outra fica parada. O segredo está em usar a Matemática corretamente para descobrir o quanto se deve girar a roda para obter a posição de rotação desejada. Mesmo que os cálculos para obter um giro estejam corretos, ainda há que se ter em mente que os robôs LEGO[®] não são máquinas de precisão (TROBAUGH, 2010). Dessa forma, é importante instruir os discentes a considerar que qualquer robô LEGO[®] construído ainda precisará de alguns ajustes finais. Esses ajustes certamente serão objeto de estudo nas fases de *reflexão* e *depuração* do ciclo de ações. São exatamente os erros que permitirão o refinamento para alcançar a precisão da manobra. Segundo Trobaugh (2010), no servomotor LEGO[®] NXT, equipamento similar aos motores do EV3, há cerca de seis a oito graus de folga na engrenagem. Portanto, inevitavelmente algumas correções serão necessárias.

No giro com uma única roda, a roda permanece estacionada enquanto a outra se move e controla a curva. Para girar o robô para a esquerda, mantém-se a roda da esquerda parada e a direita em movimento. Isso cria um movimento em circunferência sendo a posição da roda esquerda o centro dessa circunferência e a distância entre a roda direita e esquerda, o raio da circunferência, como mostrado na Figura 29.

A *distância circular* (D) relativa a uma volta completa em torno do centro de giro é calculada pela fórmula $C = 2 \cdot \pi \cdot R$, onde R é a distância entre as rodas do robô

Figura 29 - Exemplo de um giro de 180° com um único motor gerando um percurso circular com o diâmetro sendo o dobro da distância entre as rodas do robô



Fonte: Winning Design! LEGO MINDSTORMS NXT Design Patterns for Fun and Competition - James J. Trobaugh - 2010.

construído. Caso o aluno queira apenas parte da distância circular percorrida, $\frac{1}{4}$ por exemplo, aplica-se uma proporção simples. Como o aluno está habituado a expressar o percurso do robô, seja circular ou em reta, por meio de rotações, há que se conseguir uma relação entre o percurso circular desejado e o número de rotações correspondente. Como a distância circular equivalente a 360° é expressa por

$$D = 2 \cdot \pi \cdot R, \quad (1)$$

E a distância percorrida (d) por um giro de roda do robô construído é expressa por

$$d = 2 \cdot \pi \cdot r, \quad (2)$$

onde r é o raio da roda do robô construído. Tem-se de (1) e (2) que o número de rotações (N) equivalente a 360° é dado por (TROBAUGH, 2010):

$$N = \frac{2 \cdot \pi \cdot R}{2 \cdot \pi \cdot r} = \frac{R}{r}$$

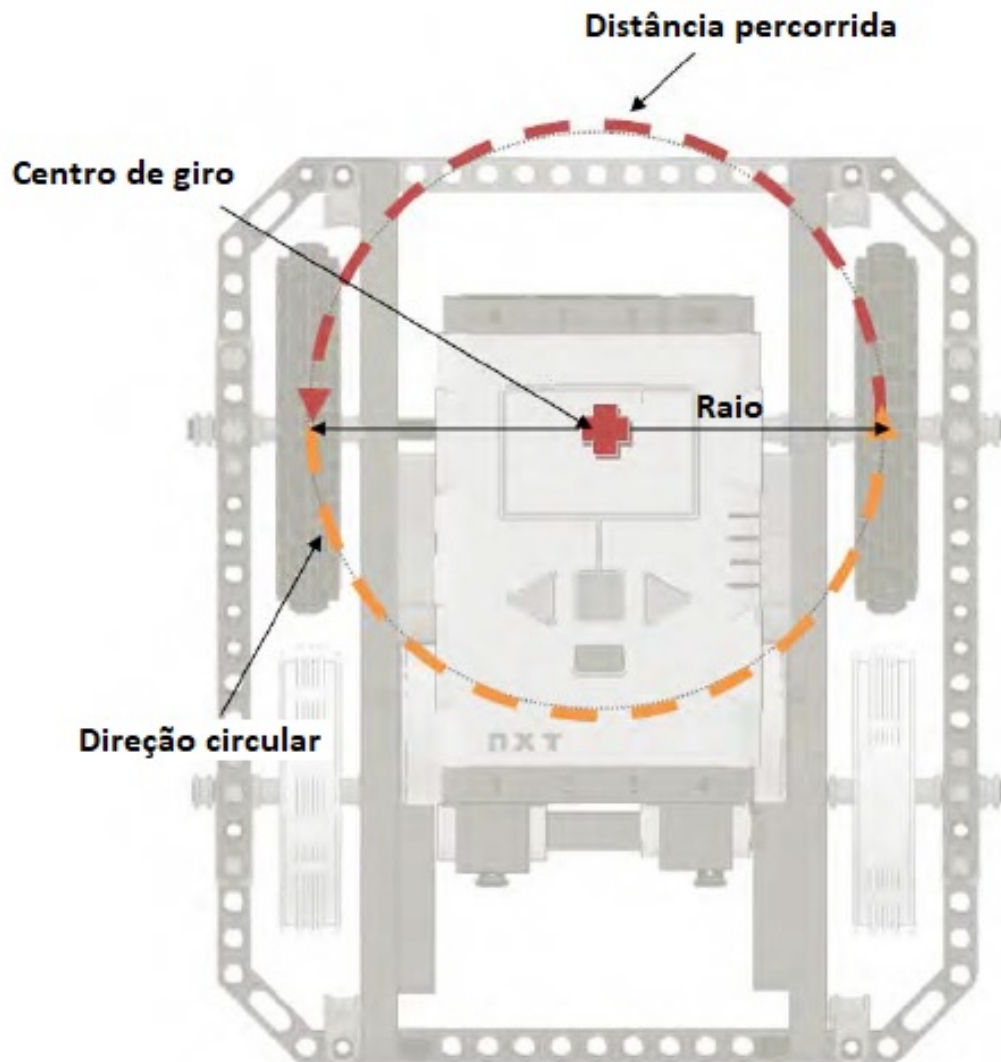
Nos testes para essa pesquisa usa-se um robô com esteira (tipo tanque) com raio de roda medindo $1,75\text{cm}$ e distância entre rodas medindo $14,5\text{cm}$. Usando a relação acima obtém-se $N \approx 8,28$. Isso significa que 8,28 rotações de roda, segundo Trobaugh (2010), aplicados no robô tanque fará com que ele gire 360° em torno de uma das rodas. Os testes revelaram que 8,28 rotações são insuficientes para alcançar o giro de uma volta. O atrito entre as rodas e a superfície de contato provavelmente motivou a diferença na manobra. Por tentativa e erro buscou-se uma solução para corrigir essa falha chegando a conclusão que o robo tanque precisa de 14 rotações (aproximadamente) de roda para girar 360° . Assim, é possível a partir da relação de Trobaugh (2010), *depurar* uma relação específica para o robô tanque. Como $14 \approx 1,69 \cdot 8,28$ e $8,28 \approx \frac{R}{r}$ tem-se que o número de rotações (N') para um giro de 360° do robô dessa pesquisa é:

$$N' \approx 1,69 \cdot \frac{R}{r}$$

Para um giro de duas rodas aplica-se a metade da relação acima, uma vez que nessa manobra ambas as rodas se movimentam, porém em sentidos opostos. Isso faz com que o centro de giro se localize na metade do eixo entre rodas portanto a metade do percurso, como mostra Figura 30.

A vantagem de se encontrar uma fórmula para o número de rotações de roda correspondente ao giro de 360° está na documentação e reutilização de códigos, prática muito utilizada na implementação de algoritmos computacionais. Tendo como referência a rela-

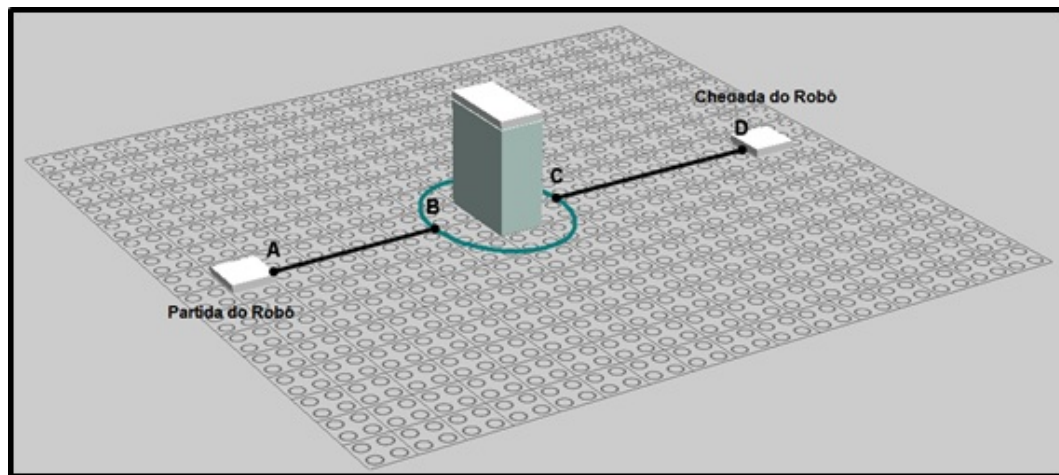
Figura 30 - Exemplo de um giro de 180° com dois motores em rotações opostas gerando percurso circular com o diâmetro sendo a distância entre as rodas do robô



Fonte: Winning Design! LEGO MINDSTORMS NXT Design Patterns for Fun and Competition - James J. Trobaugh - 2010.

ção obtida acima, os alunos podem criar um bloco de programação controlador de giros, capaz de fazer a alteração de ângulo se tornar algo simples e imediato sem a necessidade de escrever um novo código. A figura 31 mostra a representação geométrica para a manobra pelo método da semicircunferência. Os pontos A e D indicam respectivamente os pontos de partida e chegada do robô. O ponto B , a detecção do alvo, condição de parada para o início da manobra de contorno do objeto. Os segmentos de reta AB e DE , os percursos antes e depois da manobra. O objetivo agora consiste em contornar o objeto percorrendo uma semicircunferência cujos extremos são os pontos B e C . A pergunta natural é se haverá ganho de rendimento nesse caso.

Figura 31 - O percurso do robô pelo método da semicircunferência. Modelagem geométrica na plataforma LEGO ^c



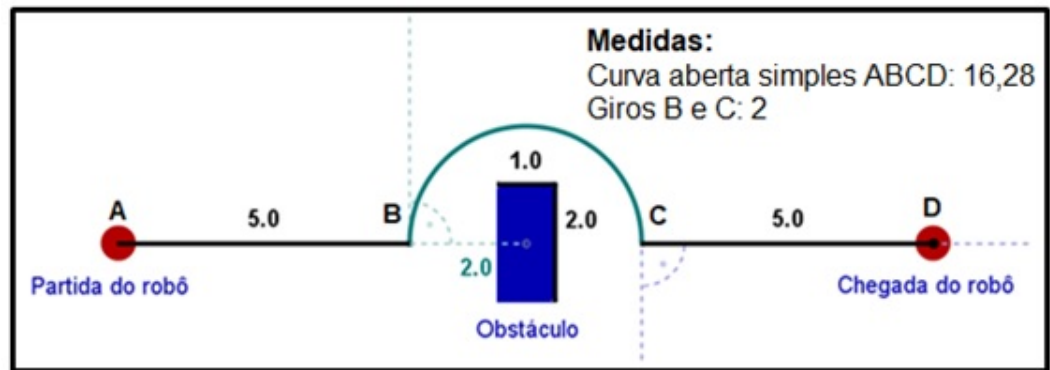
Fonte: O autor, 2018.

A manobra circular desafia o aluno a encontrar o raio de giro cujo centro coincide com o centro do retângulo. De maneira que o robô percorra o arco AB o mais próximo possível do obstáculo, conforme mostrado na Figura 32.

O ambiente de geometria dinâmica R.e.C permite ao aluno explorar com rigor e rapidez as medidas dos deslocamentos e giros do robô sem o uso de instrumentos de medição concretos. Não que o uso desses materiais - régua, compasso e transferidor - sejam menos importantes do que a ferramenta tecnológica, mas o que pretende-se é a manipulação de um instrumento digital que garante precisão de medidas, o que não se consegue com a construção geométrica tradicional por falta de prática. Percebe-se que muitos alunos tem dificuldades no manuseio desses instrumentos, pois as construções geométricas estão cada vez mais ausentes nos currículos, as escolas praticamente abandonaram o ensino dessa disciplina (PAVANELLO, 1993) e (SEMAT, 2014). A Tabela 6, a seguir, exhibe de maneira organizada os resultados dos métodos apresentados nessa seção.

Analisando a tabela constata-se que, pela quantidade de giros realizados durante as

Figura 32 - Representação geométrica da vista aérea do percurso do robô



Fonte: O autor, 2018.

Tabela 6 - Registro dos resultados das manobras do robô

Trajeto	Percurso	Giro
Retangular	18	4
Triangular	16,4	3
Circular	16,28	2

Fonte: O autor, 2018.

manobras combinados com as distâncias percorridas, prevê-se - em tese - que a manobra preferencial escolhida para a etapa de **execução** do ciclo de ações seja a triangular ou a circular pela diferença muito pequena (12 centésimos) de percurso e giro (1). A escolha do discente dependerá agora da **reflexão** acerca do impacto que a terceira variável ainda não considerada no rendimento do robô terá sobre o trajeto. A variável tempo obviamente só poderá ser aferida colocando o robô para por em prática tudo o que foi planejado até o momento.

Concluída a descrição da geometria de manobra o próximo passo, e o mais importante, é o estudo de como fazer o robô seguir a linha de salvamento até encontrar o objeto-alvo e resgatá-lo. Para a construção do algoritmo optou-se por uma exposição gráfica para elucidar todas as ocorrências de luminosidade aferidas pelos sensores na forma de fluxograma. Na sequência, e tendo como referência essa abordagem, constrói-se o pseudocódigo que já pode ser considerado a solução do desafio *Rescue Line*. Os resultados dos tempos de cada manobra para o desvio do obstáculo, um possível estudo de performance para o algoritmo do robô seguidor de linha, aqui mencionado, e os detalhes da sua construção constam no Apêndice A.

4.3 O desenvolvimento das aulas e a atuação nas competições em Macaé

A grade curricular atual do Colégio Municipal do Sana contempla as disciplinas obrigatórias previstas na BNCC e mais três disciplinas extracurriculares - Geometria, Produção Textual e Robótica - objetivando a completude da carga horária diária de tempo integral (8h às 14h40min). As aulas de Robótica são divididas em dois momentos: no turno da manhã são ministradas aulas de 50 minutos em cada turma (por semana) e, no turno da tarde, 3 horas de aula (por semana) destinadas a reforçar, aprimorar e aprofundar os conhecimentos em montagem e programação tendo em vista a participação em competições a nível regional, estadual e nacional.

No início do ano letivo, faz-se uma iniciação à linguagem de programação para preparar os alunos do 6º ano nas práticas computacionais. Inicialmente são propostas atividades bem simples voltadas ao emprego da linguagem de programação Superlogo 3.0 sobretudo na construção de figuras geométricas. São apresentados os comandos básicos da linguagem seguido de problemas de programação conforme mostrado na sequência:

Problema 1 - Faça um programa que construa um quadrado com lado de tamanho 100 pixels.

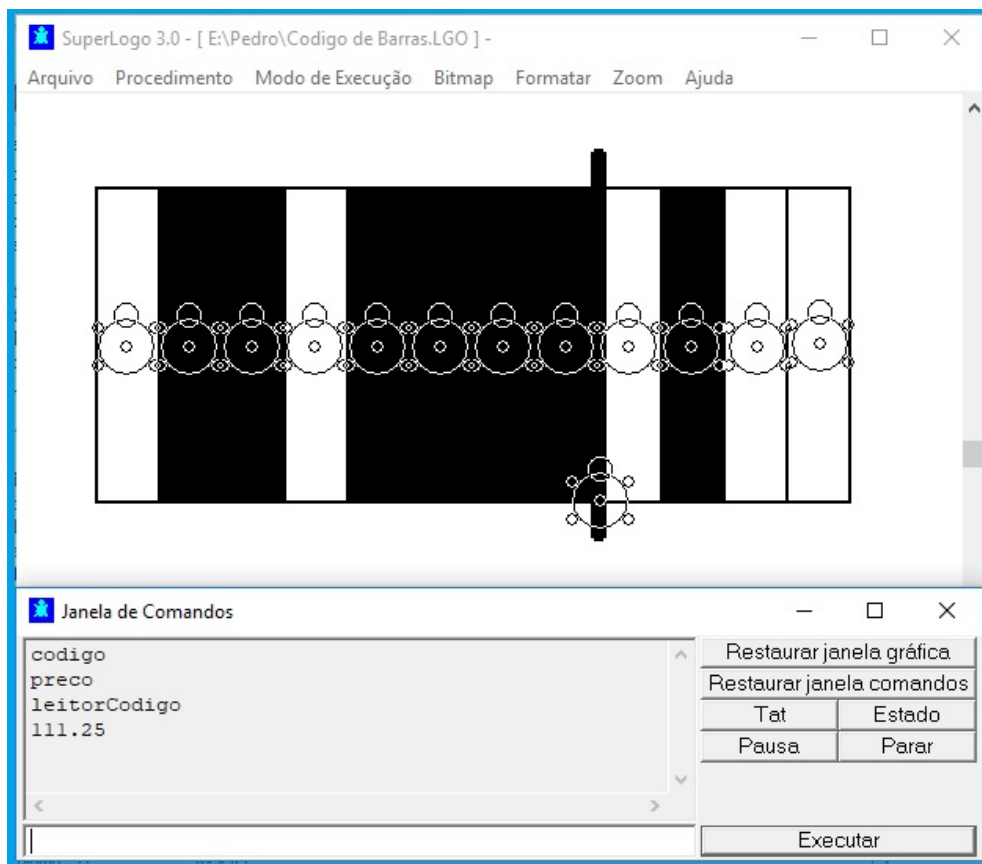
Problema 2 - Faça um programa que construa um retângulo com lados de tamanhos 50 e 80 pixels.

Problema 3 - Faça um programa que construa um triângulo equilátero com lado de tamanho 120 pixels.

Problema 4 - Faça um programa que construa uma malha quadriculada de tamanho 50 pixels preenchendo toda a tela do computador e quatro círculos de tamanho 50 pixels de diâmetro nas coordenadas $(-100, -100)$, $(-100, 100)$, $(100, 100)$ e $(100, -100)$.

No 2º bimestre são propostos problemas mais complexos que permitem a aplicação de comandos de repetição, procedimentos e sub-rotinas. Por exemplo, no projeto “Leitor de Código de Barras” os alunos tinham que construir um modelo geométrico usando essas técnicas de programação. A Figura 33 mostra a conclusão desse trabalho por um dos alunos. A janela de comandos descreve a sequência de execução dos procedimentos e o preço correspondente a leitura no sistema decimal. O código completo dessa programação se encontra no Apêndice B.

Figura 33 - Captura da tela do software Superlogo 3.0

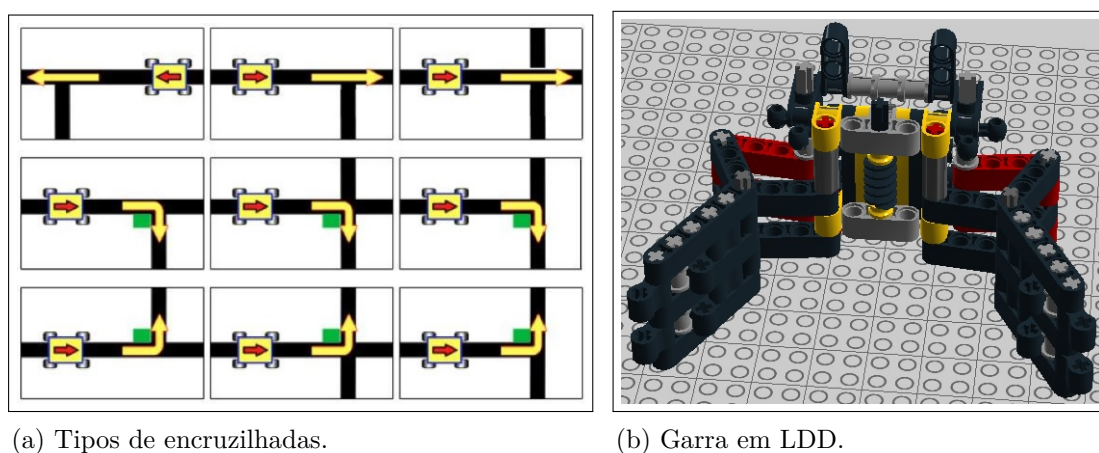


Fonte: O autor, 2018.

Ainda no primeiro semestre, intercaladas com as aulas de programação, inicia-se a

etapa de reconhecimento das peças robóticas e as primeiras montagens de protótipos. De acordo com as habilidades, competências e desempenho nessas atividades, alguns alunos são convidados para participar das aulas de aprofundamento dentro do limite de vagas, porque alguns alunos do ano anterior, que participaram de competições, já são ocupantes de vagas. Nessas aulas, os alunos são desafiados a solucionar os problemas relacionados com as competições, testando e analisando diferentes possibilidades para superar as adversidades relativas ao desempenho mecânico e computacional do robô. Um dos problemas relativo a automação consiste em elaborar um algoritmo para fazer com que o robô seguidor de linha desvie na direção da marcação verde presente numa encruzilhada, conforme mostrado na Figura 33(a). Um outro problema, correspondente a estrutura mecânica, consiste em construir um dispositivo que permite a realização do resgate da vítima (garra). As instruções de montagem seguem a metodologia do passo-a-passo através de arquivos com a extensão .lfx que são abertos na plataforma LEGO[®] Digital Designer (LDD), conforme mostrado na Figura 34(b).

Figura 34 - Atividades das aulas de aprofundamento



(a) Tipos de encruzilhadas. (b) Garra em LDD.
 Fonte: Manual de Regras e Instruções - Etapa Regional/Estadual - OBR 2018 (a) e autor, 2018 (b).

O projeto do robô deve considerar esses e outros problemas que constam no manual da Olimpíada Brasileira de Robótica (OBR, 2018a). As aulas de aprofundamento funcionam como um laboratório experimental voltado para os problemas relacionados com as competições. Um algoritmo para a solução do desvio, que foi discutido em aula, se encontra no Apêndice B.

No segundo semestre, ocorrem as competições internas visando selecionar até cinco equipes para participar do Torneio Juvenil de Robótica. Todas as turmas do 2º segmento do Ensino Fundamental participam dessa seletiva. Cada equipe compete com 4 integrantes sendo que cada aluno recebe função específica para desempenhar dentro do grupo: programador, montador, apresentador e organizador. Toda semana, dentro do horário

normal das aulas, uma situação problema é colocada para todas as equipes resolverem. Os desafios tem características semelhantes ao TJR. A descrição de duas dessas atividades se encontram no Apêndice B.

O histórico da atuação do Colégio Municipal do Sana nas competições de Robótica, no período de 2014 a 2018, é bastante expressivo. Em 2014, a primeira equipe - Sanandroid - com 10 integrantes, participou da Feira Educacional de Robótica realizada pela Secretaria Municipal de Educação de Macaé, com o tema “*Body Forward*” da FIRST[®] LEGO[®] League (FLL). Esse torneio é um programa internacional de exploração científica, projetado para fazer com que crianças e jovens de 9 a 16 anos se entusiasmem com ciência e tecnologia e adquiram habilidades valiosas de trabalho e de vida. (SESI, 2019). Segundo a organizadora do evento:

A iniciativa fortalece capacidade de inovação, criatividade e raciocínio lógico, inspirando jovens a seguir carreira no ramo da ciência, tecnologia, engenharia, artes e matemática. Por meio de uma experiência criativa, os competidores são desafiados a investigar problemas e buscar soluções inovadoras para situações da vida real, bem como programar robôs autônomos para cumprir as missões (SESI, 2019).

Como visto, o torneio exige muito mais do que montar e programar robôs. Cada temporada desafia os alunos a pensarem numa solução inovadora para um problema do mundo real. Na disputa regional em Macaé, a equipe conquistou vaga para participar da etapa estadual na cidade de Vitória no estado do Espírito Santo. O tema “*World Class*” versava sobre o seguinte assunto: “Como pode ser a aprendizagem no futuro? É possível tornar a busca pelo conhecimento mais instigante, criativa e desafiadora?” (SESI, 2019). Na oportunidade a equipe apresentou pesquisa sobre o tema “Escola de Vidro”, que propunha uma mudança na arquitetura de organização e ocupação dos espaços pelas escolas. O colégio conquistou o prêmio especial “Contra todas as Adversidades”, dada a equipe que supera alguma dificuldade significativa durante o torneio.

No ano seguinte, retornou ao Espírito Santo para participar da temporada estadual (2015/2016) da FLL com o tema “*Trash Trek*”. A equipe tinha que elaborar um projeto que considerasse a coleta, separação, reciclagem e manejo do lixo de forma inteligente. O projeto, intitulado “Lixarte” visava a reutilização do lixo descartado de forma inadequada pela comunidade transformando-o em arte. Nessa participação, a equipe conquistou o prêmio da categoria “*Core Values*” de “Trabalho em Equipe”. A Figura 35 exhibe o quadro “O coelho” exibido no evento.

A Tabela 7 a seguir resume a atuação do Colégio Municipal do Sana nos torneios da FLL de 2014 até 2018.

Em 2015, o colégio participou pela primeira vez da etapa regional da Olimpíada Brasileira de Robótica realizada na Fundação Municipal de Macaé (FUNEMAC). O evento contou com 47 equipes representantes do Estado do Rio de Janeiro oriundas dos municípios

Figura 35 - Quadro elaborado com sucata



Fonte: <https://sanandroidlixarte.blogspot.com/2015/11/o-coelho.html?fbclid=IwAR3v26-LT-c3N8pNKCJGbhYEd4mcSAOQT2HoagaQBnbnm2AYSdN2eq-3PmJU>. Acesso em 05 janeiro 2019.

Tabela 7 - Participação da equipe Sanandroid nos torneios da LEGO®

Torneio de Robótica da FIRST® LEGO® League				
Ano	Etapa	Tema e Projeto de Pesquisa	Local	Premiação
2014	Regional	Body Forward/Guia para Cegos	Macaé - RJ	Não
2014	Estadual	World Class/Escola de Vidro	Vitória - ES	Sim
2015	Estadual	Trash Trek/Lixarte	Vitória - ES	Sim
2016	Estadual	Animal Allies/Jogo da Extinção	Macaé - RJ	Não
2017	Estadual	Hydro Dynamics	Não participou	
2018	Estadual	Into Orbit/Massa Óssea	Brasília - DF	Sim

Fonte: O Autor, 2018.

de Campos dos Goytacazes, Rio das Ostras, Niterói, Rio Bonito, Rio de Janeiro e Macaé (ODEBATEON, 2018). O desempenho satisfatório garantiu vaga para a equipe na etapa estadual realizada na Pontifícia Universidade Católica (PUC) no Rio de Janeiro.

No ano seguinte, na FUNEMAC, o colégio conquistou o primeiro título em Robótica na etapa regional da OBR 2016. Duas das equipes que representaram o colégio conquistaram medalhas de ouro e bronze. Com esse resultado, passou para a etapa estadual realizada na PUC-RJ. Nessa etapa, o projeto do robô conquista o prêmio “Robustez” na forma de medalha concedida aos alunos da equipe que produziram o robô mais robusto da competição.

Figura 36 - 1º lugar na OBR 2016 - Regional - Macaé



Fonte: O autor, 2018.

A Tabela 8 a seguir, resume a atuação do Colégio Municipal do Sana na Olimpíada Brasileira de Robótica de 2015 até 2018.

Percebendo a empolgação dos alunos com a Robótica, o diretor Sol Grey Tavares Ribeiro, a pedido do autor, adquiriu novos equipamentos que permitiu a participação de um maior número de estudantes nas competições. Em 2015, com o bom retrospecto nas competições supracitadas, as equipes do Colégio Municipal do Sana, iniciaram a participação no Torneio Juvenil de Robótica na modalidade *Resgate no Plano*. Com um

Tabela 8 - Participação das equipes do Sana na OBR

Olimpíada Brasileira de Robótica			
Ano	Etapa	Local	Premiação
2015	Regional	Macaé - RJ	Não
2015	Estadual	PUC - RJ	Não
2016	Regional	Macaé - RJ	Ouro e Bronze
2016	Estadual	PUC - RJ	Robustez
2017	Regional	Macaé - RJ	Ouro
2017	Estadual	Benfica - RJ	Ouro e Bronze
2017	Nacional	Curitiba - PR	Não
2018	Regional	Macaé - RJ	Não

Fonte: O autor, 2018.

projeto bem definido, atuaram com três sensores de cor, o que facilitou sobremaneira as manobras nas curvas em ângulo reto, característica principal dessa arena. O torneio foi realizado pela Faculdade de Educação Tecnológica do Estado do Rio de Janeiro – (Faeterj - Petrópolis) no Estado do Rio de Janeiro e a equipe Sanandroid conquistou medalha de ouro no evento. Com essa vitória, se classificou para a etapa Nacional realizada no Colégio Santo Inácio, em Botafogo, no Rio de Janeiro. Nessa etapa, foi vice-campeão, na modalidade *Resgate de Alto Risco*. No ano seguinte, viajou para São Paulo para disputar a etapa final do torneio - International Tournament of Robots (ITR) - na Universidade Presbiteriana Mackenzie, onde conquistou a 1ª colocação na modalidade *Resgate no Plano*.

A Tabela 9 a seguir, resume a atuação do Colégio Municipal do Sana no Torneio Juvenil de Robótica.

Figura 37 - Atuação no Torneio Juvenil de robótica 2015/2016



(a) Equipe do Sana em *round* de competição no TJR nacional 2015.



(b) Medalha de Ouro no ITR 2016 (Mackenzie).

Fonte: Figura (a): o autor, 2018. Figura (b):

<http://www.divercidades.com/noticias/equipe-de-robotica-do-sana-e-medalha-de-ouro-em-torneio-internacional>. Acesso em 06 janeiro 2019.

Tabela 9 - Participação das equipes do Sana no TJR

Torneio Juvenil de Robótica			
Ano	Etapa	Local	Premiação
2015	Estadual	Petrópolis - RJ	Medalha de Ouro
2015	Nacional	Botafogo - RJ	Troféu de Vice-Campeão
2016	Internacional	São Paulo - SP	Medalha de Ouro
2016	Estadual	Macaé - RJ	Campeão e 3º Lugar
2016	Nacional	Botafogo - RJ	Não
2017	Estadual	Macaé - RJ	Troféu de Vice-Campeão
2017	Nacional	São Luis - MA	Prêmio Paulo Freire
2018	Internacional	Guarulhos - SP	Troféus de Vice-Campeão e 3º Lugar
2018	Estadual	Botafogo - RJ	Campeão
2018	Nacional	João Pessoa - PB	Não participou

Fonte: O autor, 2018.

CONCLUSÃO

É muito comum nas aulas de Matemática o aluno questionar o professor sobre a aplicabilidade dos conteúdos Matemáticos estudados em sala de aula. A indagação procede, haja vista que ainda perdura nas escolas o ensino tradicional calcado no trinômio - explicação, exercício e correção -. Procura-se nessa pesquisa apresentar a Robótica Educacional como uma alternativa para tornar as aulas mais atraentes e produtivas permitindo que o aluno aprenda Matemática “fazendo”. Através da construção e programação de robôs o aluno utiliza os conceitos em diversos momentos, elabora hipóteses, valida ou refuta conjecturas, aplica conteúdos apreendidos em anos anteriores de escolaridade e constrói novos conhecimentos interagindo a todo instante com diferentes meios tecnológicos.

Aplica-se o desafio *Rescue Line* das competições de Robótica como uma situação-problema a ser solucionada combinando Matemática e Computação. A complexidade para encontrar o algoritmo sugere a divisão do problema em partes menores para facilitar a compreensão (paradigma Divisão e Conquista). O rendimento dos alunos no planejamento e construção do robô, na descrição e execução da programação e na análise e depuração de resultados, revelou que essa decisão foi coerente. Os objetos de conhecimento previstos na BNCC foram desenvolvidos nas diversas fases do estudo, por exemplo, o desvio do obstáculo proporcionou a aplicação e construção de entes geométricos enquanto o algoritmo do robô seguidor de linha oportunizou o uso de contagem, números, desigualdades e proporcionalidade. As ferramentas tecnológicas de apoio, como o software de geometria dinâmica e a linguagem de programação Superlogo 3.0, ajudaram o aluno na descrição dos problemas e no desenvolvimento de suas ideias de forma autônoma e independente. A proposta também permitiu o uso de instrumentos de medição tradicional como régua, esquadro e transferidor. Observou-se que alguns alunos, mesmo nos anos finais do Ensino Fundamental, demonstraram ainda falta de habilidade com esses materiais.

Concluí-se que a metodologia escolhida e aplicada tanto nas aulas de aprofundamento quanto no horário normal foi satisfatória porque observamos nas atividades para a seleção das equipes de competição, que os desafios foram superados a partir da discussão de ideias entre alunos, auxílio do professor e do uso do computador para depurar as informações. Uma característica da aula de aprofundamento é a presença de discentes de diferentes idades e anos de escolaridade o que não foi impedimento para o desenvolvimento e compreensão dos conceitos envolvidos. Em diversos momentos, pode-se observar alunos mais novos discutindo e orientando alunos mais velhos, indicando que o nível de cognição independe do grau de escolaridade e da idade. Nem toda dúvida deve ser sanada pelo professor, ela pode ser resolvida por diferentes sujeitos, fontes ou atitudes como um colega de classe, um software de apoio como o LEGO[©] Mindstorms[©] EV3 Programming, um

ciclo de ações (*descrição-execução-reflexão-depuração*), no computador ou pela observação do comportamento do robô.

A experiência descrita vem contribuir com o pensamento de que a Robótica Educacional desenvolve conceitos, habilidades e competências de forma interdisciplinar colaborando para a difusão da ciência e tecnologia estimulando a prática de atividades investigativas sobretudo no ensino/aprendizagem de Matemática tornando o estudo dessa disciplina mais interessante e produtiva. O erro, como um dos elementos da investigação, foi tratado como componente positivo e essencial no processo de busca da solução. Em nenhuma das atividades o erro foi motivo de impedimento para seguir na construção do conhecimento, muito pelo contrário, era comum a conclusão de um pensamento precedido da depuração de um erro.

Uma inquietação sempre me acompanhou no exercício da função docente: busco a todo instante inovar a minha prática pedagógica procurando me colocar no lugar do aluno refletindo sobre o tipo de aula que estou conduzindo. Penso ser cansativo ficar horas sentado apenas assistindo uma aula sem ter a oportunidade de “fazer” a aula. Acredito que cresço profissionalmente com esse trabalho porque reforço minhas concepções pedagógicas sobre como atuar diferente para poder alcançar resultados diferentes. Considero que uma aula deva ser um processo ativo de construção de conhecimento onde haja interação entre os diferentes sujeitos e recursos disponíveis. A pesquisa apontou que é possível utilizar concomitantemente vários recursos tecnológicos tornando as aulas mais dinâmicas. O estudo de algoritmo e a sua implementação em linguagem de programação gráfica mostrou-se viável e possível a partir do sexto ano de escolaridade. Ressalto porém que a sua utilização requer do professor uma capacitação específica. A participação em torneios de Robótica em diferentes níveis corroborou para me manter motivado e apto a planejar desdobramentos no tema. O número de alunos participantes em competições cresce rapidamente, assim é de se esperar que a Robótica Educacional se consolide como um paradigma de ensino/aprendizagem nas escolas brasileiras.

REFERÊNCIAS

- ALMEIDA, Maria Aparecida. *Possibilidades da Robótica para a Educação Matemática*. Curitiba: PDE, 2007. 28 p.
- COLL, César. *Psicologia e currículo: Uma aproximação psicopedagógica à elaboração do currículo escolar*. São Paulo: Editora Ática, 1996. 200 p.
- CONGRESSO NACIONAL DE EDUCAÇÃO MATEMÁTICA, 2., 2011, Tocantins. *ANÁLISE DOS ERROS COMETIDOS NAS OPERAÇÕES FUNDAMENTAIS COM NÚMEROS REAIS: UM ESTUDO DE CASO*. Tocantins: IFT, 2014. 10.
- DANTE, Luiz Roberto. *Didática da Resolução de Problemas de Matemática*. 10. ed. São Paulo: Ática, 1998. 176 p.
- FEITOSA, Jefferson Gustavo. *Manual Didático Pedagógico*. 1. ed. Curitiba, PR: ZOOM Editora, 2013. 124 p.
- GARBER, Gary. *Learning LEGO MINDSTORMS EV3: Build and create interactive, sensor-based robots using your lego mindstorms ev3 kit*. 1. ed. BIRMINGHAM - MUMBAI: Packt Publishing Ltd, 2015. 284 p.
- GASPARELLO, Arlette Medeiros. *Metodologia do Trabalho Científico*. Rio de Janeiro: CEP - EB, 2006. 116 p.
- GUADAGNIN, Alan Junior. *Controle híbrido de um robô seguidor de linha*. 2014. 65 f. Monografia (Graduação em Engenharia Elétrica) — Universidade Tecnológica do Paraná, Pato Branco, PR, 2014.
- KARVINEN, Kimmo; TERO, Karvinen. *Primeiros passos com sensores: Perceba o mundo usando eletônica, arduino e raspberry pi*. São Paulo: Novatec Editora Ltda, 2014. 160 p.
- MOLL, Jaqueline. *Diretrizes Curriculares Nacionais para Educação Básica*. Brasília, DF: MEC, 2013. 542 p.
- MOREIRA, Antônio. *Currículos e programas no Brasil*. 2. ed. São Paulo: Papirus Editora, 1995. 232 p.
- MOREIRA, Antônio; SILVA, Tomaz. *Currículo, Cultura e Sociedade*. 2. ed. São Paulo: Cortez Editora, 1994. 154 p.
- OBR. *Manual de Estudos – Modalidade Teórica – Versão 1.0: Março/2018: 2009 a 2018*. São Paulo: Olimpíada Brasileira de Robótica, 2018. 23 p. Relatório Técnico.
- _____. *Manual de Regras e Instruções – Etapa Regional/Estadual – Versão 1.0: Março/2018 Modalidade Prática/2018: 2007 a 2018*. São Paulo: Olimpíada Brasileira de Robótica, 2018. 41 p. Relatório Técnico.
- OBR, 2018. Olimpíada Brasileira de Robótica. 2018. Apresenta informações sobre a competição. Disponível em: <<http://www.obr.org.br>>. Acesso em: 09 dezembro 2018.

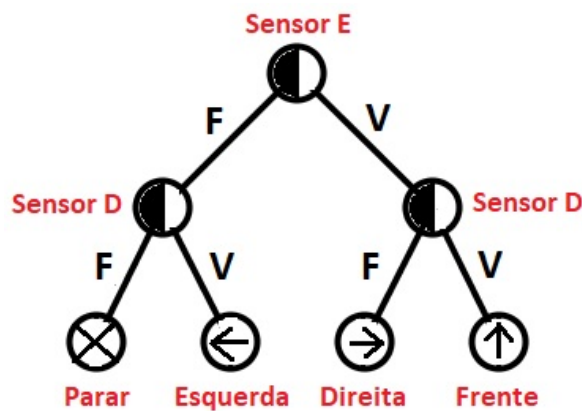
- ODEBATEON. Jornal Eletrônico. 2018. Apresenta notícias gerais. Disponível em: <[https://www.odebateon.com.br/site/noticia/detalhe/34090/etapa-regional-da-olimpiada-de-robotica-reune-estudantes->](https://www.odebateon.com.br/site/noticia/detalhe/34090/etapa-regional-da-olimpiada-de-robotica-reune-estudantes-). Acesso em: 05 janeiro 2018.
- PARK, Eun Jung. *Exploring LEGO® MINDSTORMS® EV3: Tools and techniques for building and programming robots*. 1. ed. Indianapolis: John Wiley & Sons, 2014. 405 p.
- PAVANELLO. Zetetiké. 1993. Periódicos: O abandono do Ensino de Geometria no Brasil. Disponível em: <<https://periodicos.sbu.unicamp.br/ojs/index.php/zetetike/article/view/8646822/13724>>. Acesso em: 03 janeiro 2019.
- POLYA, George. *A arte de resolver problemas: Um novo aspecto do método matemático*. 9. ed. Rio de Janeiro: Interciência, 1995. 180 p.
- RATHS, J. A. *Teaching without specific objectives*: In magoon, r. a. Columbus - Ohio - Meurill: Education and psychology, 1973.
- SAVIANI, Dermeval. *Escola e Democracia*. 42. ed. São Paulo: Autores Associados Ltda, 2018. 219 p.
- SECRETARIA DE EDUCAÇÃO FUNDAMENTAL. *Introdução aos Parâmetros Curriculares Nacionais: Terceiro e quarto ciclos do ensino fundamentais*. Brasília, 1998.
- SECRETARIA MUNICIPAL DE EDUCAÇÃO. *Referencial Curricular da Rede Municipal de Ensino de Niterói: Uma construção coletiva*. Niterói - RJ, 2010.
- SECRETARIA MUNICIPAL DE EDUCAÇÃO. *Caderno de Orientação Curricular*. Macaé - Rj, 2012.
- SEMINÁRIO DA LICENCIATURA EM MATEMÁTICA, 6., 2014, Espírito Santo. *DESENHO GEOMÉTRICO: CONSTRUÇÕES FUNDAMENTAIS*. Cachoeiro de Itapemirim, Coordenadoria de Licenciatura em Matemática: IFES, 2014. 3.
- SESI. Torneio de Robótica. 2019. Apresenta informações sobre a temporada World Class. Disponível em: <<http://www.portaldaindustria.com.br/sesi/canais/torneio-de-robotica/world-class-20142015/>>. Acesso em: 04 janeiro 2019.
- SILVA, Luis Rogério da. *Torneio Juvenil de Robótica: Difundir desafios sempre: Resgate de alto risco no plano: O desafio de fazer um robô para explorar áreas de risco*. 5. ed. São Paulo: <http://www.torneiojrobotica.org>, 2018. 39 p.
- TROBAUGH, James J. *Winning Design! LEGO MINDSTORMS NXT: Design patterns for fun and competition*. New York, NY: Apress., 2010. 313 p.
- VALENTE, José Armando. *A espiral da espiral de aprendizagem: O processo de compreensão do papel das tecnologias de informação e comunicação na educação*. 2005. 234 f. Tese (Livre Docência) — Instituto de Artes, Universidade Estadual de Campinas, Campinas, SP, 2005.
- VALK, Laurens. *The Lego® Mindstorms® EV3 Discovery Book: A beginner's guide to building and programming robots*. 2. ed. São Francisco, CA: No Starch Press, Inc., 2014. 371 p.

ZOOM - FASCÍCULO DE EDUCAÇÃO PARA A VIDA. Curitiba: ZOOM Editora Educacional LTDA, 2010.

APÊNDICE A – A solução do desafio *Rescue Line*

Para a construção do algoritmo foi discutida uma maneira de representar todas as ocorrências de leitura dos sensores para manter o robô na linha usando reflexão. A discussão levou a construção do esquema a seguir, Figura 38, onde *Sensor E* e *Sensor D* significam, respectivamente, o posicionamento dos sensores à esquerda e à direita no robô.

Figura 38 - Expansão do método On-Off para dois sensores



Fonte: O autor, 2018.

Duas ocorrências verdadeiras (VV) indicam que o robô deve seguir em frente. A ocorrência (VF) indica que o sensor da direita fez uma leitura de cor escura, logo o robô deve girar para a direita. A ocorrência (FV) indica que o robô deve girar para à esquerda. E finalmente duas ocorrências falsas (FF) indicam que o robô deve parar porque os dois sensores estão sobre cor escura. Essas leituras devem ocorrer ininterruptamente, em ciclo. Considerando a calibração padrão de limite 50, a análise das ocorrências acima, levou a construção do seguinte algoritmo, denominado *segueLinha*:

INÍCIO

Enquanto não (condição de parada) **faça**

Se (SensorE e SensorD) > 50 então **faça**

Robô para frente;

Se (SensorE > 50 e SensorD < 50) então **faça**

Gire o robô para a direita;

Se (SensorE < 50 e SensorD > 50) então **faça**

Gire o robô para a esquerda **senão**

PARE;

Fim do Enquanto

FIM

Um estudo em separado deve ser feito para o caso em que o robô está sobre cor escura, que pode ocorrer em qualquer lugar do percurso. Por exemplo, uma curva fechada faz o robô parar. No TJR ou na OBR o descarte do objeto-alvo é numa região escura, portanto outra condição de parada. Como fazer o robô distinguir uma curva de um nicho? Uma solução encontrada segue a partir da posição atual do robô (parado sobre cor escura). Que pode ser local para descarte de objeto-alvo ou local de estacionamento do robô encerrando a missão (TJR). O próximo algoritmo, denominado *conflito*, resolveu a questão:

INÍCIO

Robô para frente por 0,5 segundo;

Se (SensorE) > 50 (branco) então **faça**

Robô para trás por 0,5 segundo;

segueLinha;

Se (10 < SensorE < 15) (verde) então **faça**

segueLinha (1ª ocorrência de verde);

Se (SensorE < 6) (Preto) então **faça**

Descarte o objeto alvo;

Gire o robô 180 graus;

segueLinha;

Se (10 < SensorE < 15) (verde) então **faça**

PARE (2ª ocorrência de verde)

FIM

O algoritmo *conflito* é colocado dentro do algoritmo *segueLinha* no lugar da condição de parada (FF). A partir de agora, toda vez que os sensores detectarem duas cores escuras (preto ou verde) a sub-rotina *conflito* é executada para informar o robô se ele deve seguir, descartar ou parar.

O teste das manobras considerando o tempo de execução comprovou que a circular é mais rápida que a triangular com diferença extremamente pequena, conforme mostrado na Tabela 10 a seguir.

Pelos resultados apresentados quanto a velocidade de execução, as manobras triangular e circular, na prática são equivalentes, ficando a cargo de cada equipe competidora a preferência na sua implementação em código. Pelos testes, verificou-se também que a manobra triangular deve ser iniciada a uma distância mínima de 14cm enquanto a circular 6cm. Com isso, se o obstáculo estiver muito próximo de uma curva - 10cm pela regra

Tabela 10 - Método circular em média 31 milésimos mais rápido que o método triangular

Tempo das manobras (em segundos)			
Teste	Retangular	Triangular	Circular
1	16,715	13,029	13.055
2	16,639	13,145	13.155
3	16,689	13.216	13.087
Média	16,681	13,13	13,099

Fonte: O autor, 2018.

- o robô terá que recuar para realizar a manobra triangular, enquanto que a manobra circular pode ser iniciada sem recuo. Nesse caso, como o caminho da pista de competição é aleatório convém se preparar para o pior caso de posição do obstáculo. Escolhido o método circular falta agora construir um algoritmo para diferenciar o objeto-alvo do obstáculo. Pelas regras do TJR, o obstáculo mede 16cm de altura enquanto o objeto-alvo mede $12,5\text{cm}$, (SILVA, 2018). Logo, uma estratégia que surgiu foi a colocação de um sensor de toque no robô posicionado entre $12,5\text{cm}$ e 16cm e um sensor ultrassônico abaixo de $12,5\text{cm}$. Dessa forma, ao se deparar com um objeto a sua frente, se o sensor de toque for acionado executa-se a manobra de desvio, caso contrário, a distância mínima é detectada pelo sensor ultrassônico indicando objeto-alvo. O algoritmo denominado *desvioAlvo* ficou dessa forma:

INÍCIO

Se (SensorT for acionado) **faça**

manobraCircular;

segueLinha;

Se (SensorU < 6) **faça**

Gire o robô 180 graus;

Capture o objeto-alvo;

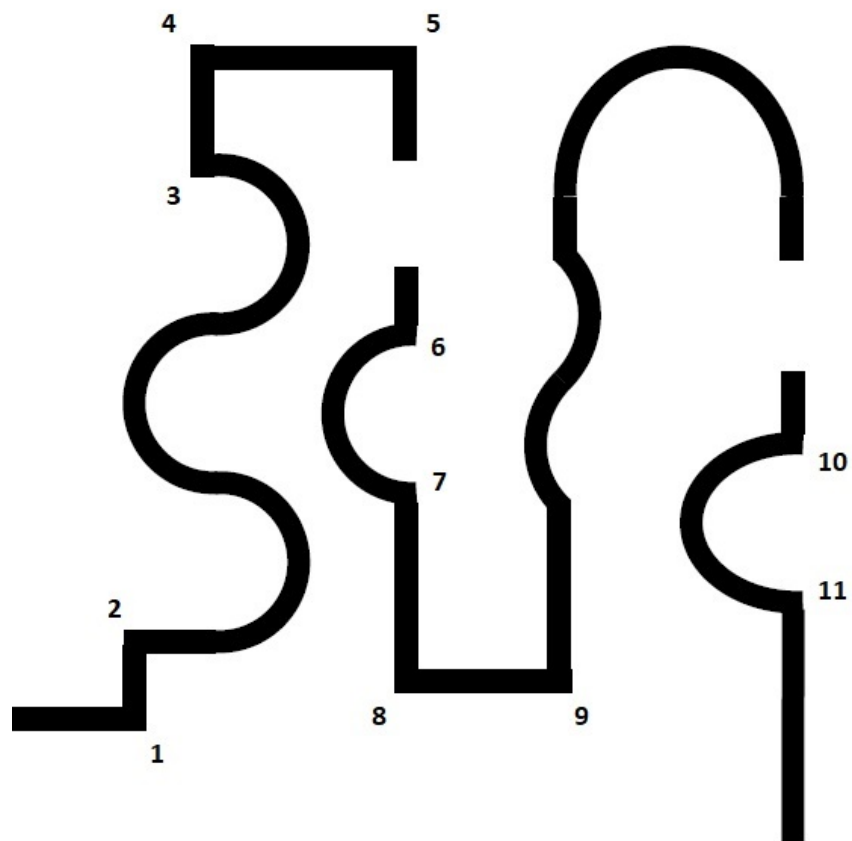
Gire o robô 180 graus;

segueLinha;

FIM

A performance do algoritmo *segueLinha* pode ser realizada tomando como referência a primeira metade da arena do desafio *resgate no plano*, idealizada e confeccionada por Raul Dario Cabrera Tapia, percurso oficial utilizado no Torneio Juvenil de Robótica, Figura 39. Essa arena é uma boa referência de teste devido a quantidade de curvas fechadas.

Figura 39 - Metade da arena do resgate no plano, percurso com 11 pontos críticos



Fonte: <http://torneiojrobotica.org>. Acesso em: 13 Dez. 2018.

O algoritmo *segueLinha* mostrou-se eficiente em 83,3% das tentativas. A falha de progresso ocorrido na curva 4 pode suscitar um futuro estudo de caso. O tempo médio da execução da manobra também foi satisfatório em comparação com o previsto para o percurso completo, que é de 3 minutos (SILVA, 2018). Considerando a média de 90 min para a metade do percurso, o robô completou em 61% do tempo em comparação com a pior performance mostrada na Tabela 11, que foi de 55,4 segundos. A Tabela 11 mostra o resultado dos testes realizados.

Tabela 11 - Avaliação do algoritmo *segueLinha*

Testes	Falha de progresso	Tempo
1	Não	51,7 segundos
2	Não	52,3 segundos
3	Sim - curva 4	não computado
4	Não	53,5 segundos
5	Não	55,4 segundos
6	Não	53,7 segundos

Fonte: O autor, 2018.

APÊNDICE B – Sugestões de atividades

B.1 Atividade 1: 100% de velocidade Matemática!

O bloco *Matemática* apresentado na subseção 4.4.3 (paleta de controle de dados) faz o controle da velocidade do motor usando a cor detectada pelo sensor, porém ela só permite alcançar 70%. Modifique o programa para usar toda a gama de valores para as velocidades do motor. Adicione um bloco de exibição ao seu programa para verificar se a saída do bloco *Matemática* atinge 100% quando a cor marrom é detectada. Sugestão: É possível inserir números decimais nos campos *a* e *b* do bloco *Matemática*.

Normalmente o aluno tende a seguir a sugestão dada por Valk (2014) na descrição do problema, ou seja, ele tentará encontrar um número decimal para substituir a constante 10 do resultado conhecido no programa *velocidade controlada*. Um aluno do 6º do ensino fundamental poderá investigar essa questão como divisão em partes iguais e chegará na fração $\frac{100}{7}$, cujo decimal correspondente é uma dízima periódica. Como esse tema, *a priori*, não é abordado nesse nível de escolaridade (EF08MA05 - BNCC), tem-se aqui uma boa oportunidade de aplicar as ideias da aprendizagem em espiral de maneira cautelosa haja vista que o objetivo mesmo, norteado pela BNCC pelo descritor (EF06MA11) é levar o discente a :

Resolver e elaborar problemas com números racionais positivos na representação decimal, envolvendo as quatro operações fundamentais e a potenciação, por meio de estratégias diversas, utilizando estimativas e arredondamentos para verificar a razoabilidade de respostas, com e sem uso de calculadora.

O arredondamento de $\frac{100}{7}$ com duas casas decimais é 14,28. Ao inserir esse valor como constante no bloco *Matemática* obtém-se como saída para o motor os seguintes valores constantes na tabela 12.

Note que a última linha da tabela retorna valor próximo de 100, mas não igual a 100, isto é, 99,96. Esses quatro centésimos de diferença fará com que o aluno caia num outro dilema: como exibir 100% na tela do tijolo EV3 usando o bloco monitor se 99,96 é diferente de 100? A solução usando apenas um bloco de operações de dados requer uma mudança no modo operativo, trocando o seletor multiplicação para o seletor avançado, onde é possível trabalhar com números racionais na forma fracionária e expressões numéricas envolvendo mais de um operador.

Para usar o bloco de dados *Matemática* necessariamente no modo multiplicação juntamente com o fator 14,28 deve-se colocar um arredondamento no decimal 99,96 usando um bloco específico *arredondar*. O resultado pode ser passado para o bloco de exibição *monitor* mostrando dessa forma que o motor atinge 100% quando encontra a cor marrom,

Tabela 12 - Leitura do sensor \times fator = Potência

Cor do sensor	Constante	Potência/velocidade
0	14,28	0
1	14,28	14,28
2	14,28	28,56
3	14,28	42,84
4	14,28	57,12
5	14,28	71,4
6	14,28	85,68
7	14,28	99,96

Fonte: O autor, 2018.

solucionando o problema.

B.2 Atividade 2: Velocidade Ultrassônica!

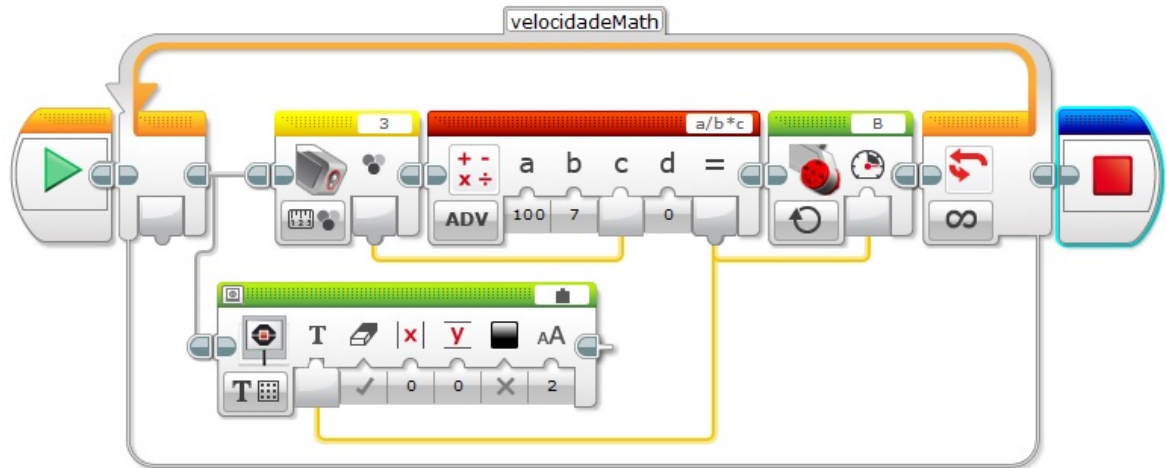
Crie um programa semelhante ao da atividade 1 para controlar a velocidade e direção do motor B usando medição de distância do sensor ultrassônico. O motor deve girar em potência 50 para uma distância de 100 cm, ficar parado para uma distância de 50 cm e girar em potência -50 (reverso) para distância nula. Sugestão: use a fórmula $\text{Velocidade} = \text{distância} - 50$.

A sugestão dada por Valk (2014) pode ser dispensada para que o aluno tenha a oportunidade de deduzir a fórmula praticando as ideias apresentadas nessa dissertação.

Uma solução para a atividade 1 é mostrada na (Figura 40).

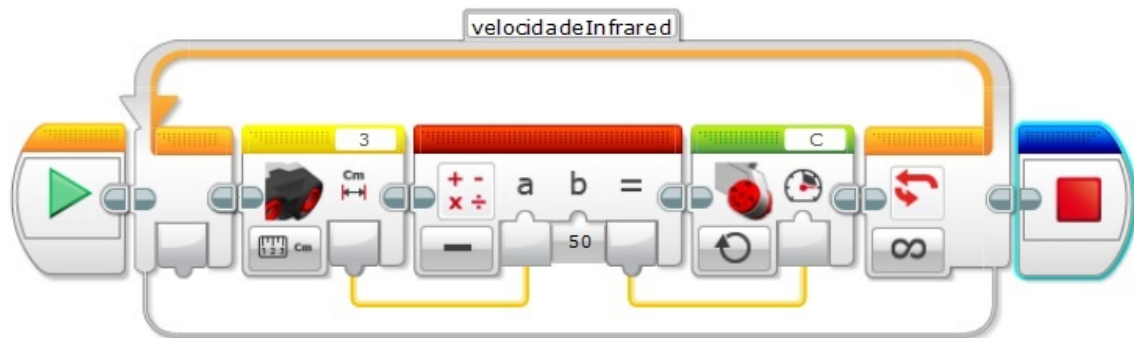
Uma solução para a atividade 2 é mostrada na (Figura 41).

Figura 40 - Bloco *Matemática* no modo avançado combinado com o sensor de cor



Fonte: O autor, 2018.

Figura 41 - Bloco *Matemática* no modo avançado combinado com o sensor ultrassônico



Fonte: O autor, 2018.

APÊNDICE C – Projetos

C.1 Realizações

C.1.1 Programação do código de barras em linguagem Superlogo 3.0

aprenda *retangulo*

pf 100 pe 90 pf 20 pe 90 pf 100 pe 90 pf 20 pe 90

fim

aprenda *digito*

repita 4 [*retangulo* pe 90 pf 20 pd 90]

fim

aprenda *separador*

un mudeel [5 5] pd 90 pf 160 pe 90 ul pf 110 pt 120 pf 10 mudeel [1 1]

fim

aprenda *leitor*

atat 1 un pf 50 pe 90 pf 10 pd 90 atat 2 un pf 50 pe 90 pf 30 pd 90 atat 3 un pf 50 pe 90 pf 50 pd 90 atat 4 un pf 50 pe 90 pf 70 pd 90 atat 5 un pf 50 pe 90 pf 90 pd 90 atat 6 un pf 50 pe 90 pf 110 pd 90 atat 7 un pf 50 pe 90 pf 130 pd 90 atat 8 un pf 50 pe 90 pf 150 pd 90 atat 9 un pf 50 pe 90 pf 170 pd 90 atat 10 un pf 50 pe 90 pf 190 pd 90 atat 11 un pf 50 pe 90 pf 210 pd 90 atat 12 un pf 50 pe 90 pf 230 pd 90

fim

aprenda *codigo*

repita 3 [*digito*]

separador

leitor

fim

aprenda *numero*

digito separador leitor

fim

aprenda *preco*

repita 12 [atat sortnum 12 mudecp “preto pinte]

fim

aprenda *zero*

numero

fim

aprenda *um*

numero

atat 1 mudecp “preto pinte

fim

aprenda *dois*

numero

atat 2 mudecp “preto pinte

fim

aprenda *tres*

numero

atat 2 mudecp “preto pinte atat 1 mudecp “preto pinte

fim

aprenda *quatro*

numero

atat 3 mudecp“preto pinte

fim

aprenda *cinco*

numero

atat 3 mudecp “preto pinte

atat 1 mudecp “preto pinte

fim

aprenda *seis*

numero

atat 3 mudecp “preto pinte atat 2 mudecp “preto pinte

fim

aprenda *sete*

```

    numero
    atat 1 mudec p “preto pinte atat 2 mudec p “preto pinte atat 3 mudec p “preto pinte
fim

```

```

aprenda oito
    numero
    atat 4 mudec p “preto pinte
fim

```

```

aprenda nove
    numero
    atat 4 mudec p “preto pinte atat 1 mudec p “preto pinte
fim

```

```

aprenda leitorCodigo
    atribua “valor 0
    para [x 1 12][atat :x se pixel = [0 0 0][atribua “valor :valor + pot 2 :x-5]]
    mostre :valor
fim

```

C.1.2 Discussão da estratégia para a encruzilhada sinalizada em verde.

Descrição da manobra tomando como exemplo a encruzilhada em cruz, conforme mostra a Figura 42:

O robô segue linha no sentido (1) \rightarrow (3) quando se depara com uma encruzilhada. A programação faz com que o robô incline para a esquerda, mudando a direção (1) \rightarrow (2) até que os dois sensores veem cor escura e para nos pontos A e C. A partir daí, os sensores do robô percorrem as distâncias AB (lado esquerdo) e CD (lado direito) fazendo contagem a cada incidência de cor verde ou preta por reflexão de luz. As contagens são armazenadas em variáveis que são acessadas no fim do processo. Com isso é possível saber a posição do sinalizador verde e executar a manobra para o lado do sensor que detectou a sua presença.

Em algoritmo fica:

```

Enquanto não cessa 1 segundo faça
    contadorE  $\leftarrow$  0 contadorD  $\leftarrow$  0
    Se sensorE < 5 (preto) faça
    contadorE  $\leftarrow$  contadorE + 1
    Se < 8 sensorE < 12 (verde) faça

```


contadorE \leftarrow contadorE + 1

Se sensorD < 5 **faça**

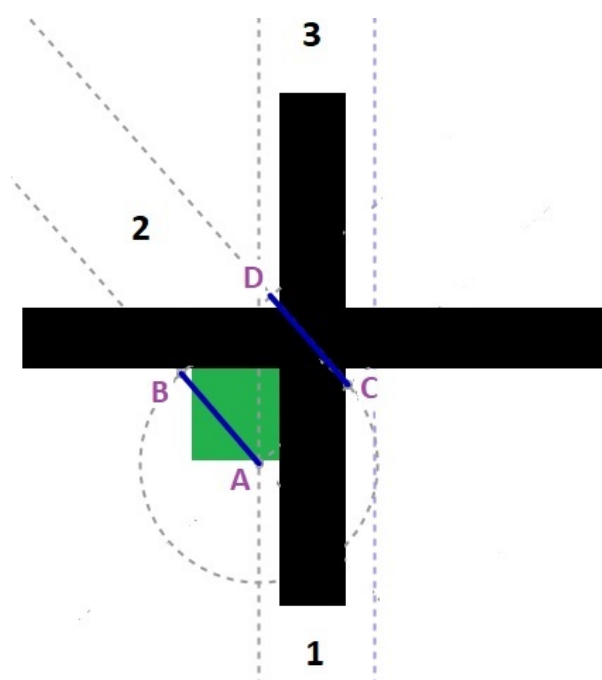
contadorD \leftarrow contadorD + 1

Se < 8 sensorE < 12 **faça**

contadorD \leftarrow contadorD + 1

Fim do **Enquanto**

Figura 42 - Contagem com sensores



Fonte: O autor, 2018.

C.1.3 Treinamento para as competições

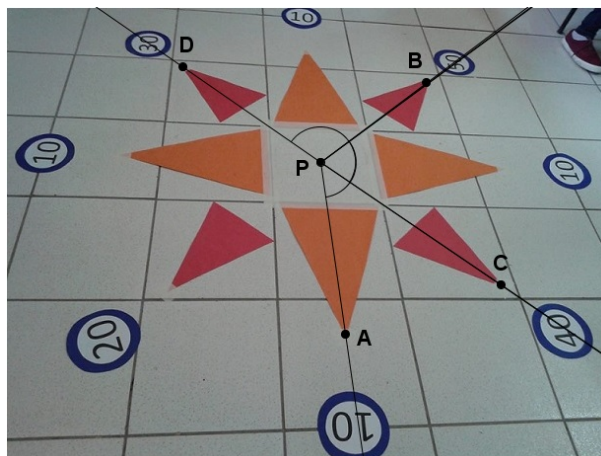
Descrição da atividade (rosa dos ventos):

O robô, inicialmente no centro da rosa dos ventos, deve se deslocar para três pontos cardeais representados por círculos numerados de 10 a 50. Definiu-se como norte a direção da semirreta PA , conforme mostrado na Figura 43. Os três pontos cardeais são sorteados e a pontuação só é considerada se o robô colocar pelo menos uma de suas rodas ou esteira sobre a região circular. É permitido percurso de ré e, obrigatoriamente, toda manobra de giro deve ocorrer no centro da rosa dos ventos.

Descrição da atividade (resgate das vítimas):

O robô posicionado na entrada da sala de resgate, conforme mostrado na Figura

Figura 43 - Programação de movimento

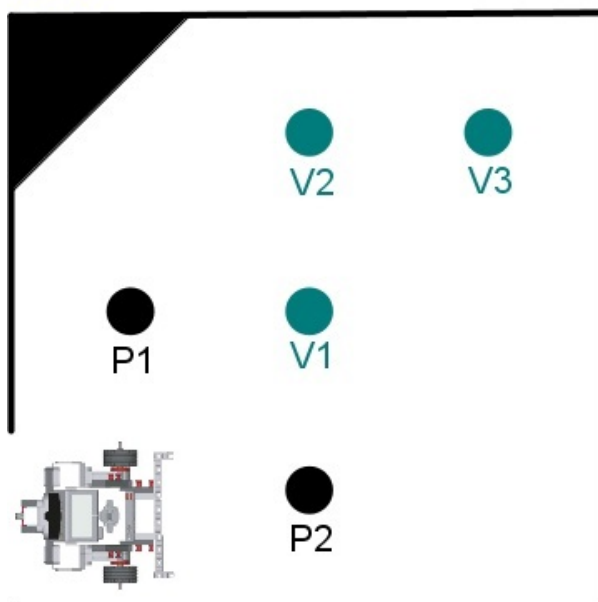


Fonte: O autor, 2018.

44, deve resgatar vítimas vivas (30 pontos) ou mortas (10 pontos) colocando-as no nicho (triângulo isósceles preto). Não há especificação de prioridade no resgate, podendo resgatar em qualquer ordem, vivas primeiro e mortas depois, ou vice-versa. A estratégia de percurso fica a critério das equipes.

Figura 44 - Programação de resgate

Nicho



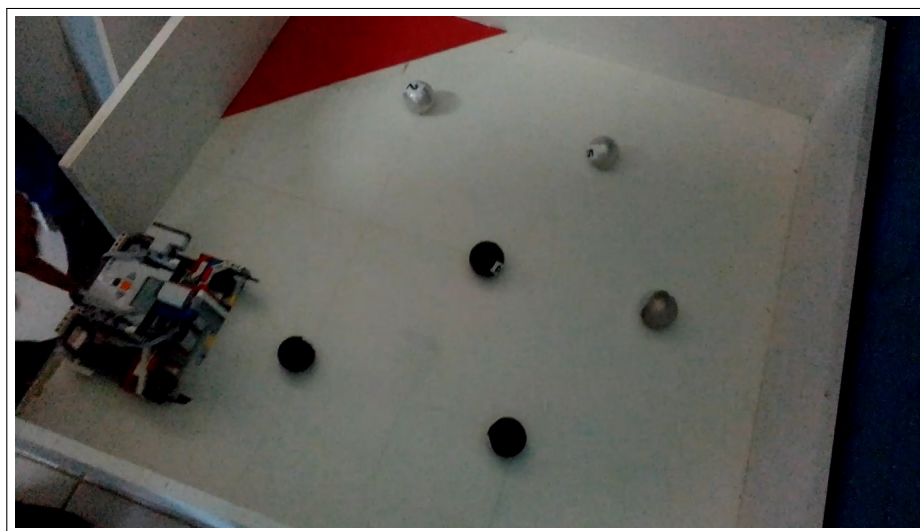
Fonte: O autor, 2018.

C.1.4 Atividades em sala ambiente

Figura 45 - Aulas de Robótica no laboratório de Informática



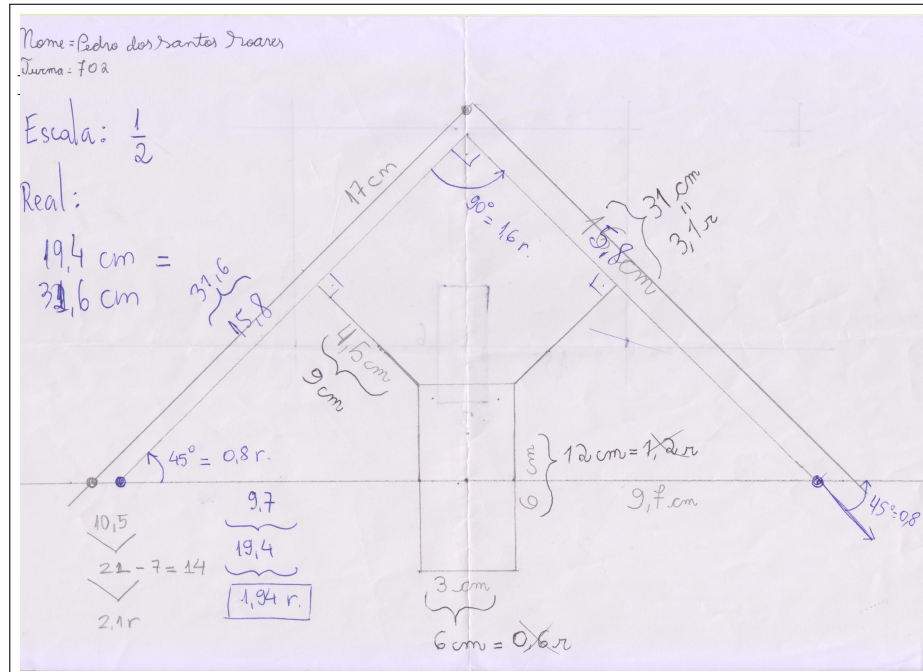
(a) Laboratório de Informática.



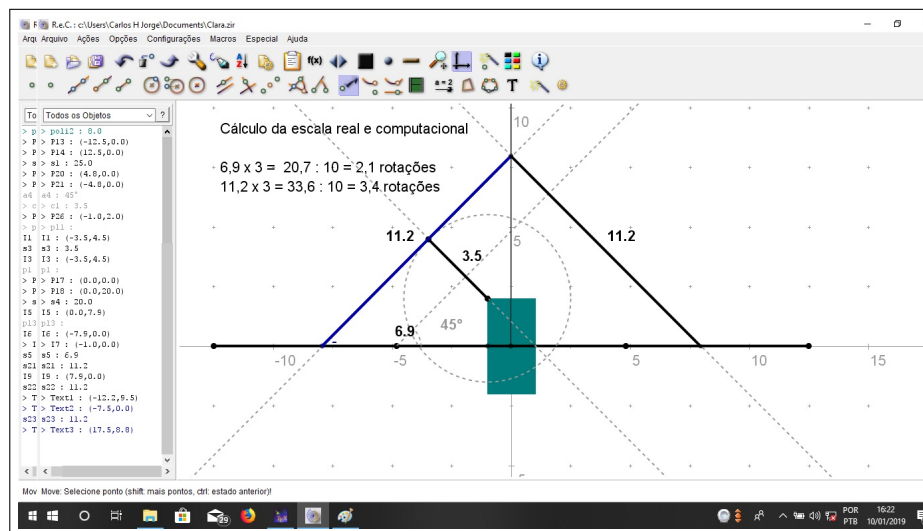
(b) Executando a programação de resgate.

Fonte: O autor, 2018.

Figura 46 - Aulas de Robótica no laboratório de informática



(a) Manuscrito da manobra triangular.



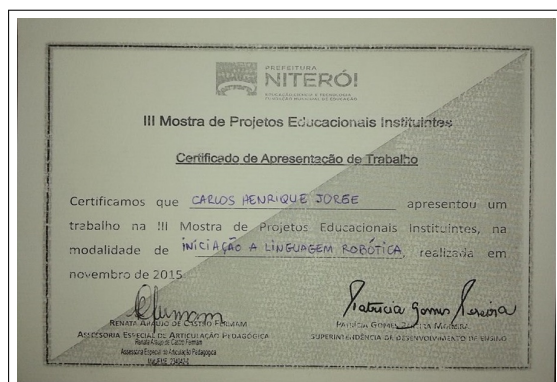
(b) Manobra triangular em R.e.C Fonte:

O autor, 2018.

ANEXO A – Certificações

A.1 Trabalhos apresentados

Figura 47 - Certificados de apresentação dos projetos de Robótica Educacional



(a) III Mostra de Projetos Educacionais Intituintes.



(b) I Mostra Acadêmico Científica de Niterói: Educação, Ciência, Tecnologia e Inovação.

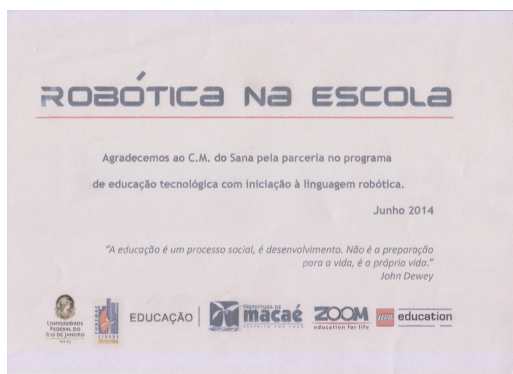


(c) I Seminário de Educação e Tecnologia de Niterói.

Fonte: Prefeitura Municipal de Niterói, Fundação Municipal de Niterói, Secretaria Municipal de Ciência e Tecnologia.

A.2 Participação em eventos e competições

Figura 48 - Primeiro ano da Robótica no Colégio Municipal do Sana.



(a) Iniciação à linguagem robótica do Sana.



(b) Projeto “*Body Forward*” da FLL.



c) Primeira participação em competições.

Fonte: Secretaria Municipal de Educação de Macaé 48(a) e 48(b) e SESI Vitória - Es.