

**Celso de Oliveira Lisboa**

**LOGIKID: SOFTWARE PARA O ENSINO DE  
LÓGICA E PROGRAMAÇÃO COM O AEDROMO**

**Bauru - SP**

**2015**

**Celso de Oliveira Lisboa**

**LOGIKID: SOFTWARE PARA O ENSINO DE  
LÓGICA E PROGRAMAÇÃO COM O AEDROMO**

Orientador: Prof. Dr. Renê Pegoraro

Coorientador: Prof. Me. Silas F. R. Alves

**Bauru - SP**

**2015**

**Celso de Oliveira Lisboa**

**LOGIKID: SOFTWARE PARA O ENSINO DE  
LÓGICA E PROGRAMAÇÃO COM O AEDROMO**

**BANCA EXAMINADORA**

**Prof. Me. Prof. Me. Silas F. R. Alves**

**DCo - FC - UNESP - Bauru**

**Coorientador**

**Prof. Dr. Aparecido Nilceu Marana**

**DCo - FC - UNESP - Bauru**

**Prof. Dr. Wilson Massashiro Yonezawa**

**DCo - FC - UNESP - Bauru**

**Bauru, 12 de janeiro de 2016**

*\*Que os vossos esforços desafiem as impossibilidades, lembrai-vos  
de que as grandes coisas do homem foram conquistadas do que  
parecia impossível. \**

*Charles Chaplin*

*\*A melhor maneira de prever o futuro é inventá-lo \**

*Alan Kay, cientista da computação, em 1971.*

## **AGRADECIMENTOS**

À minha família, que me apoiou e incentivou durante todos os anos de minha vida.

Agradeço ao professor Renê Pegoraro pelo apoio, amizade, orientação e ensinamentos.

A todos os professores que me auxiliaram e instruíram nas mais diversas áreas do conhecimento durante minha formação acadêmica.

Aos meus grandes amigos, pelo companheirismo.

## RESUMO

Diante da visível necessidade de novas tecnologias na Educação, dentre várias, a robótica é uma que se destaca no contexto atual pelo seu caráter inovador e cativante. O objetivo deste trabalho foi o desenvolvimento de um software denominado LOGIKID, que com o auxílio do AEDROMO, pode ser utilizado como técnica complementar no ensino da lógica para alunos do ensino fundamental e médio. O AEDROMO é um ambiente experimental e didático com robôs móveis. Formado por uma área de trabalho, objetos passivos e robôs; capaz de trabalhar com um software desenvolvido em qualquer linguagem que suporte sockets. Dentre as diversas linguagens existentes, optou-se pelo Processing. O LOGIKID é baseado em características de softwares como LOGO, SCRATCH, LIGHTBOT, LEGO MINDSTORMS e LEGAL que foram consideradas positivas no contexto educacional. A sua interface interage com o usuário de acordo com suas escolhas para a posterior geração e interpretação de um código. Essa interpretação utiliza um interpretador configurável desenvolvido no GISDI (Grupo de Integração de Sistemas e Dispositivos Inteligentes) que media o código gerado para ser utilizado com o AEDROMO.

**Palavras-chaves:** Ciência da computação; Robótica no ensino; Software educativo.

## **ABSTRACT**

Faced with the perceived need for new technologies in education, among many, robotics is one that stands out in the current context for its innovative and captivating character. The objective of this study was to develop a software called LOGIKID, which with the help of AEDROMO, can be used as a complementary technique in logic teaching for elementary and middle school students. The AEDROMO is an experimental and didactic environment with mobile robots. Formed by a workspace, passive objects and robots; able to work with software developed in any language that supports sockets. Among the various existing languages, we opted for the Processing. The LOGIKID is based on software features like LOGO, SCRATCH, Lightbot, LEGO MINDSTORMS and LEGAL that were considered positive in the educational context. Its interface interacts with the user according to their choices for the later generation and interpretation of a code. This interpretation uses a configurable interpreter developed in GISDI (Group of System Integration and Intelligent Devices) that measured the generated code to be used with the AEDROMO.

**Key words:** Computer Science; Robotics in education; Educational software.

## LISTA DE FIGURAS

Figura 1: Arquitetura do AEDROMO .....	12
Figura 2: Blocos do SCRATCH e fragmentos de algoritmo. ....	23
Figura 3: Jogo LIGHTBOT. ....	25
Figura 4: Possibilidades de robôs projetáveis do sistema LEGO.....	26
Figura 5: Tela do software EV3 do Kit LEGO MINDSTORMS.....	27
Figura 6: Regiões do ambiente de programação LEGAL. ....	29
Figura 7: Ambiente do AEDROMO.....	31
Figura 8: Simulador.....	32
Figura 9: Diagrama de estados.....	35
Figura 10: Fluxograma dos comandos dos botões.....	38
Figura 11: Interface desenvolvida no Processing. ....	39
Figura 12: Interface com opção de comando "andar para frente" .....	40
Figura 13: Interface com comando "virar". ....	41
Figura 14: Interface após alguns comandos.....	42
Figura 15: Sinalização da arena e do robô (fora de escala).....	43
Figura 16: Atividade 1 - Exercitando o básico da lógica. ....	44
Figura 17: Atividade 2 – Deslocamento de bolas.....	45
Figura 18: Atividade 3 – Desviar dos obstáculos. ....	46
Figura 19: Atividade 4 – Deslocamento de bolas com manobras. ....	47
Figura 20: Atividade problema 5 – Sequência de passos.....	48
Figura 21: Apresentação do LOGIKID no evento Campus Party Brasil 2016. ....	50

## LISTA DE QUADROS

Quadro 1 – Características comparativas entre os trabalhos pesquisados e o LOGIKID. ....	33
Quadro 2: Palavras reservadas para o LOGIKID.....	36
Quadro 3: Operadores lógicos. ....	36
Quadro 4: Rotinas. ....	37



<b>Sumário</b>	
<b>RESUMO .....</b>	<b>6</b>
<b>ABSTRACT .....</b>	<b>7</b>
<b>CAPÍTULO 1 – INTRODUÇÃO .....</b>	<b>10</b>
<b>1.1 Objetivo .....</b>	<b>11</b>
<b>1.2 Justificativa .....</b>	<b>11</b>
<b>1.3 Estrutura da monografia .....</b>	<b>13</b>
<b>CAPÍTULO 2 - FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>15</b>
<b>2.1 Software LOGO .....</b>	<b>18</b>
<b>2.1.1 Pressupostos .....</b>	<b>19</b>
<b>2.1.2 Pesquisas com software LOGO .....</b>	<b>20</b>
<b>2.2 Software SCRATCH .....</b>	<b>21</b>
<b>2.2.1 Pressupostos .....</b>	<b>21</b>
<b>2.2.2 Pesquisas com software SCRATCH .....</b>	<b>23</b>
<b>2.3 Software LIGHTBOT .....</b>	<b>24</b>
<b>2.3.1 Pressupostos .....</b>	<b>24</b>
<b>2.4 LEGO MINDSTORMS .....</b>	<b>26</b>
<b>2.4.1 Pressupostos .....</b>	<b>26</b>
<b>2.4.2 Pesquisas com software MINDSTORMS .....</b>	<b>28</b>
<b>2.5 LEGAL .....</b>	<b>29</b>
<b>2.5.1 Pressupostos .....</b>	<b>29</b>
<b>2.6 AEDROMO .....</b>	<b>30</b>
<b>CAPÍTULO 3 – LOGIKID .....</b>	<b>34</b>
<b>3.1 Estrutura do LOGIKD .....</b>	<b>34</b>
<b>3.2 Palavras reservadas e rotinas do LOGIKID .....</b>	<b>35</b>
<b>3.3 Interface do LOGIKID .....</b>	<b>37</b>
<b>CAPÍTULO 4 – EXPERIMENTOS REALIZADOS .....</b>	<b>43</b>
<b>CAPÍTULO 5 – CONCLUSÕES E CONSIDERAÇÕES FINAIS .....</b>	<b>49</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>51</b>

## CAPÍTULO 1 – INTRODUÇÃO

Diante da visível necessidade de novas tecnologias na Educação, dentre várias, a robótica é uma que se destaca no contexto atual pelo seu caráter inovador e cativante. A presente pesquisa pretende ajudar nesse sentido por meio do desenvolvimento de uma tecnologia que possa ser utilizada no meio educacional.

Mesmo perante o avanço tecnológico atual, muitas pesquisas mostram que o uso dessa tecnologia no processo educacional ainda é precário. Junta-se a isso o fato da educação se tornar cada vez menos motivadora para os alunos que assumem que o ensino se baseia somente em métodos decorativos, cansativos e distantes de sua realidade (NETTO, 2005).

Alves et al. (2011) afirmam que a presença de um robô na sociedade em um futuro próximo é incontestável. Diante disso, faz-se imprescindível a familiarização e desmistificação desta tecnologia que oferece suporte para outras áreas do conhecimento. Segundo Barker e Ansorge (2007) citado por Alves et al. (2011), a robótica é uma ferramenta pedagógica e um objeto didático visto que proporciona um aprendizado ativo em dois aspectos, o primeiro, no uso da robótica para educação no ensino de conceitos particulares ou gerais de disciplinas envolvidas com a própria robótica e o segundo, no uso da robótica na educação como elemento motivador.

O GISDI (Grupo de Integração de Sistemas e Dispositivos Inteligentes), entre outras contribuições, apresenta o AEDROMO como uma ferramenta com potencial para o ensino de robótica. O AEDROMO é um Ambiente Experimental e Didático com Robôs Móveis, formado por uma área de trabalho, objetos passivos, um ou mais robôs, dois computadores, uma câmera global do tipo webcam e um transmissor. As premissas na concepção e desenvolvimento desse ambiente são a flexibilidade de adaptação, o uso por várias disciplinas e o baixo custo (FERASOLI et al. 2006).

Visando unir a desmistificação da tecnologia, em particular o uso da robótica por meio do computador, com uma educação mais moderna, motivadora e contextualizada com a realidade dos alunos atualmente, este trabalho propõe o desenvolvimento um software denominado LOGIKID, aderente AEDROMO, para auxiliar na resolução da problemática discutida anteriormente.

## **1.1 Objetivo**

O objetivo deste trabalho foi o desenvolvimento de um software denominado LOGIKID, que como cliente do AEDROMO, pode ser utilizado como ferramenta complementar no ensino da lógica para alunos de ensino fundamental e médio. O software visa auxiliar o desenvolvimento nos alunos de um raciocínio lógico na resolução de desafios propostos.

Consequentemente, os objetivos específicos do trabalho foram:

- Especificar uma linguagem de programação em português que contenha conceitos básicos de matemática, de lógica de predicados de primeira ordem, e de robótica;
- Construir um interpretador para a linguagem especificada;
- Desenvolver um ambiente de desenvolvimento integrado que facilite a programação dos robôs usando a linguagem especificada.

## **1.2 Justificativa**

Avanzato (2000) afirma que as abordagens tradicionais de ensino superior não cativam a atenção dos alunos por darem ênfase aos aspectos teóricos das disciplinas. Neste sentido, a introdução de novas tecnologias no processo de aprendizado pode fornecer resultados positivos. O autor afirma também que uma abordagem que tem apresentado êxito em diferentes níveis de ensino é a robótica que como ferramenta pedagógica para o ensino de engenharia, apresenta muitos aspectos interessantes. Dentre esses aspectos estão: qualidade intrinsecamente multidisciplinar, estimuladora do trabalho em equipe e promotora um retorno visual real de forma estimulante e motivadora. Entretanto, o alto custo dos robôs é uma dificuldade encontrada no uso da robótica em sala de aula.

Papert (1980) defende a robótica como um recurso tecnológico auxiliar utilizável no processo educacional que pode contribuir para o desenvolvimento cognitivo do aluno e para habilidades intelectuais específicas. Afirma que a robótica deve ser explorada e encarada como ferramenta pedagógica interessante sobre vários aspectos.

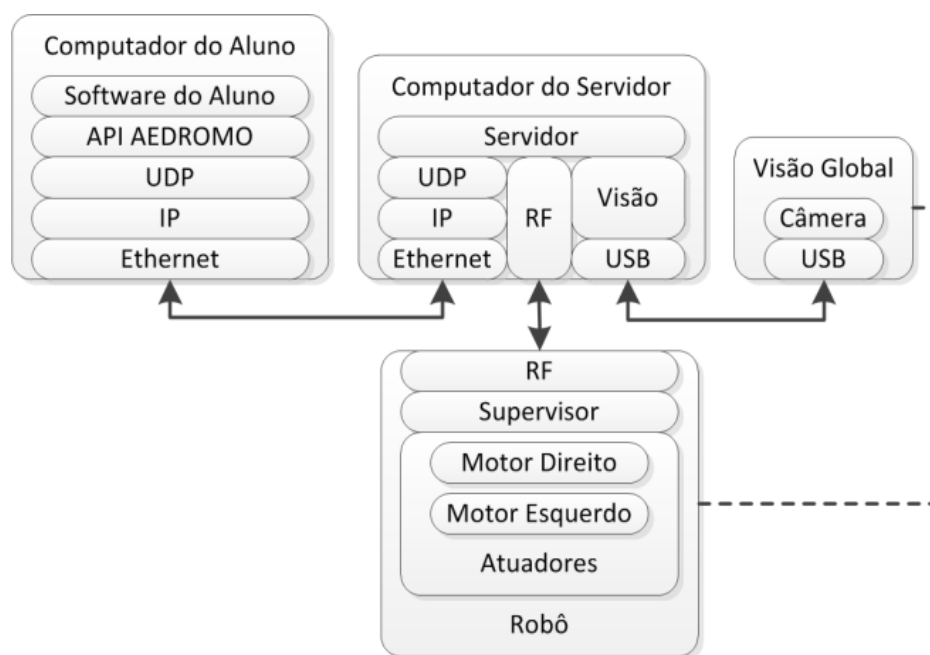
Nesse sentido, o Grupo de Integração de Sistemas e Dispositivos Inteligentes, (GISDI), do Departamento de Computação (DCo) da UNESP de Bauru desenvolve iniciativas de baixo custo, flexíveis e alinhadas para o ensino de Engenharia e da própria Robótica (ALVES et al. 2011).

O AEDROMO - sistema que foi desenvolvido pelo grupo GISDI - é utilizado neste trabalho como uma ferramenta para o uso do software educacional elaborado, o LOGIKID.

Na arquitetura do AEDROMO há dois tipos de computadores: o Computador do Servidor, responsável pela implementação do sistema de visão computacional e pela comunicação com os robôs, e o Computador do Aluno, onde o aplicativo de controle (software do Aluno) é desenvolvido (ALVES et al., 2011). Ainda segundo Alves et al. (2011, p. 4):

Este aplicativo [software do aluno] é um cliente que recebe do servidor as posições de todos os objetos presentes na área de trabalho e envia ao servidor os comandos de acionamento de um robô, para que o servidor encaminhe estes comandos ao robô. A comunicação entre o cliente e o servidor é dada através de uma rede Ethernet utilizando o protocolo UDP/IP (*User Datagram Protocol over Internet Protocol*), permitindo que seja escrito um aplicativo de controle para cada robô. Adicionalmente, os aplicativos de controle podem ser executados no mesmo computador ou em diferentes máquinas.

Figura 1: Arquitetura do AEDROMO



Fonte: Alves et al. (2011).

Alves et al. (2011) citam que esta escolha de arquitetura (mostrada na Figura 1) aproxima o aluno da robótica e ao mesmo tempo reduz o custo do sistema. Dentro da arquitetura, o PC (computador pessoal) desempenha um papel fundamental, visto que é familiar a maior parte dos alunos que têm acesso à PCs nas universidades ou mesmo em casa, sendo que o computador os aproxima da robótica por meio de ferramentas já conhecidas por eles. Ainda afirmam que, ao mesmo tempo, é o PC que confere a flexibilidade da arquitetura através de softwares modulares e abertos além de já estar inserido na infraestrutura de computação existente nas universidades, que dispõem de computadores nos seus laboratórios de informática.

Podemos inferir esse mesmo raciocínio para outros níveis de ensino, visto que muitas escolas dispõem de computadores em seus laboratórios de informática. Diante disso, enfatiza-se aqui a importância da presente pesquisa que visa a elaboração de um software auxiliar para o ensino, que faria com que professores e alunos se familiarizassem com a tecnologia dos computadores e da robótica.

### **1.3 Estrutura da monografia**

Esta monografia está estruturada da seguinte forma:

- Capítulo 2 – Fundamentação teórica: Apresenta sínteses de trabalhos, utilizando a Ciência da Computação e a robótica, cujos objetivos englobaram não apenas o ensino de conceitos de computação, mas também o ensino de conceitos de outras disciplinas por intermédio da computação.
- Capítulo 3 – LOGIKID: Relata sobre a elaboração do software desenvolvido para ser utilizado como técnica complementar no ensino da lógica para alunos do ensino fundamental e médio.
- Capítulo 4 – Experimentos realizados: Apresenta atividades de lógica desenvolvidas cuja aplicação, em conjunto com o software LOGIKID,

pode auxiliar no desenvolvimento de um pensamento computacional e lógico pelos alunos

- Capítulo 5 – Conclusões e considerações finais: Apresenta as conclusões e considerações finais da pesquisa e a intenção de trabalhos futuros envolvendo a aplicação do software no contexto educacional.

## CAPÍTULO 2 - FUNDAMENTAÇÃO TEÓRICA

O aumento na oferta de oportunidades de aprendizagem dos fundamentos da Ciência da Computação foi definido como estratégico para o progresso científico e tecnológico de qualquer país por um relatório apoiado pela *Association for Computing Machinery* – ACM (Denning, 2004), entidade internacional que discute os rumos da Ciência da Computação, assim como por uma recomendação conjunta divulgada pela Sociedade Brasileira de Computação – SBC e o Ministério da Educação – MEC (SBC, 2011). O desenvolvimento do Pensamento Computacional seria uma questão primordial para a formação de profissionais e cidadãos com melhores condições de empregarem as tecnologias na transformação do seu próprio cotidiano (SANT'ANA; NEVES, 2012).

Entretanto, no Brasil, o aprendizado dos fundamentos da Ciência da Computação é reservado principalmente àqueles que optam por cursos de graduação na área (FRANÇA et al., 2012). Em outros países como os Estados Unidos, Canadá e Israel, o ensino de programação já é aplicado também no ensino básico. O Grupo de Licenciatura em Computação da Sociedade Brasileira de Computação (SBC) tem defendido a proposta de incluir conteúdos de informática no Ensino Médio (PEREIRA JUNIOR et al., 2005).

Dentro da Ciência da Computação, a robótica vem causando impactos em várias áreas do conhecimento em virtude de sua característica inovadora. A robótica educativa não é recente, visto que surgiu por volta da década de 1960, quando seu pioneiro Seymour Papert desenvolvia sua teoria do construcionismo e defendia o uso do computador como recurso atrativo para as crianças no contexto escolar.

Segundo Gomes et al. (2010) a robótica educativa pode ser definida como:

[...] um conjunto de conceitos tecnológicos aplicados à educação, em que o aprendiz tem acesso a computadores e softwares, componentes eletromecânicos como motores, engrenagens, sensores, rodas e um ambiente de programação para que os componentes acima possam funcionar. Além de envolver conhecimentos básicos de mecânica, cinemática, automação, hidráulica, informática e inteligência artificial, envolvidos no funcionamento de um robô, são utilizados recursos pedagógicos para que se estabeleça um ambiente de trabalho escolar agradável. Aí se simula uma série de acontecimentos, muitas vezes

da vida real, com alunos e professores interagindo entre si, buscando e propiciando diferentes tipos de conhecimentos [...]

Para D'abreu (2014), ainda que não se tenha ainda alcançado a universalização da prática da Robótica Pedagógica<sup>1</sup>, a mesma não pode mais ser considerada uma novidade nas escolas de educação infantil, fundamental e médio. Para ele, a robótica pedagógica aponta para uma mudança de paradigma no processo de ensino e aprendizagem:

[...] tem como objetivo “o aprendizado de ciências de forma lúdica e dessa maneira despertar o interesse dos alunos nas áreas tecnológicas, envolvendo: Concepção: Discussão e troca de ideias sobre o desenvolvimento de um robô; Construção: Montagem do robô e Automação: Elaboração de programas para automação e controle do robô (D'ABREU, 2014, p. 80).

Portanto, trata-se de uma atividade de construção do conhecimento que pode possibilitar o desenvolvimento das inteligências corporal sinestésica e musical, lógico matemático, linguística, espacial, intrapessoal e interpessoal. Além disso, a robótica pedagógica é uma atividade que possibilita a criação de situações de ensino e aprendizagem interdisciplinar que possibilitam o entendimento de outras culturas e outros modos de compreensão da realidade. De um ponto de vista científico-tecnológico significa entender os princípios básicos de funcionamento de muitas tecnologias que fazem parte do nosso cotidiano (D'ABREU, 2014).

Além da recomendação conjunta do Ministério da Educação (MEC) e da Sociedade Brasileira de Computação (SBC), várias pesquisas como as relatadas nos Anais do V Workshop de Robótica Educacional (Editadas pela Sociedade Brasileira de Computação) apontam como muito positivo o uso de softwares no contexto de ensino aprendizagem e justificam isso por meio de resultados positivos de experiências realizadas em vários níveis de ensino.

Nascimento et al. (2014) verificaram resultados positivos com relação ao desempenho na disciplina de Matemática após a realização de oficinas de robóticas para educandos do 6º e 7º ano de uma Escola Estadual em Natal, Rio Grande do Norte. Segundo os pesquisadores, tais alunos estavam desmotivados

---

<sup>1</sup> Considera-se nesse trabalho robótica pedagógica e robótica educativa com o mesmo significado.



e com baixo rendimento escolar e durante o trabalho com as oficinas demonstraram gostar de construir o robô e após essas atividades, apresentaram aumento do interesse e motivação com relação aos conteúdos dados em sala de aula. Defendem a eficácia da robótica no meio escolar como meio motivador, de reforço e facilitador na mediação, construção, aquisição e ampliação do conhecimento no processo ensino-aprendizagem dos alunos. Além disso, afirmam que houve aumento da autoestima dos educandos e o desenvolvimento da autonomia e de competências e habilidades no uso da lógica na resolução de problemas.

Jesus et al. (2014) utilizaram a linguagem de programação do ArduAlg com alunos iniciantes em lógica de programação e algoritmos. Segundo eles, o ArduAlg permite que o estudante desenvolva seus algoritmos sem se preocupar com detalhes técnicos inerentes a tecnologia do Arduino. Constataram que uma das principais vantagens dessa utilização é permitir que a robótica educacional seja aplicada de forma mais fácil, barata e flexível.

Segundo Angonese et al. (2014) as equipes do SIRLab (Laboratório de Sistemas Inteligentes e Robótica) competem em olimpíadas de robótica, dentre elas futebol de robô, e essas e outras experiências mostram que o laboratório tem sido um ponto de integração e motivação para os alunos do ensino médio/técnico e superior, contribuindo para uma formação de qualidade de futuros profissionais de carreiras tecnológicas.

Barros et al. (2014) concluíram que com o uso das tecnologias educacionais assistivas de baixo custo auxiliam a acessibilidade da robótica educacional para os deficientes visuais. Também defendem que a robótica educacional é uma ferramenta que capacita pessoas alterando suas perspectivas sobre a resolução de problemas de lógica em geral. Complementam afirmando que a assimilação de conteúdos de lógica, facilita a assimilação de outros conteúdos relacionados ao currículo escolar tradicional.

Pensando na dificuldade de disponibilizar um número mínimo de robôs por número de alunos, além do espaço físico requerido, Shimabukuro e Pegoraro (2014) defendem o uso de simuladores que permitem que os alunos testem suas soluções sem utilizar robôs reais. Afirmam que depois de ter entendido o funcionamento do robô e do ambiente, o aluno pode fazer os testes em um ambiente real. Propõe o uso de um simulador com visualização tridimensional

para o Ambiente Experimental Didático com Robôs Móveis (AEDROMO) juntamente com uma biblioteca destinada ao usuário e a utilização da linguagem Processing.

O simulador usa uma interface muito simples e apresenta uma imagem dinâmica do que está ocorrendo no espaço da arena. Diante da melhoria oferecida pela visualização tridimensional, os alunos sentem maior proximidade com o ambiente real, tornando mais concreto suas experiências obtidas por meio do simulador. Além disso, a biblioteca possibilita que um maior número de pessoas possa utilizar o AEDROMO, visto que não exige delas conhecimentos específicos de comunicação e controle dos robôs. Em virtude disso, a pessoa pode se concentrar na solução da atividade proposta. Concluem que a biblioteca em conjunto com o simulador apresentados para o AEDROMO podem ampliar as práticas educativas deste ambiente, proporcionando um maior alcance desta ferramenta para o ensino da robótica e assuntos relacionados à programação (SHIMABUKURO; PEGORARO, 2014).

Foram apresentados brevemente diversos exemplos nos quais a aplicação de pressupostos da Ciência da Computação e da Robótica no contexto de Ensino-Aprendizagem não se limitou a nenhum nível de ensino nem as dificuldades/limitações individuais dos estudantes ou ainda a dificuldades financeiras ou relacionadas à falta de espaço.

Diante disso, foram buscados trabalhos correlatos, cujos objetivos englobam não apenas o ensino de conceitos de computação, mas também o ensino de conceitos de outras disciplinas por intermédio da computação. Portanto apresenta-se a seguir um levantamento bibliográfico de alguns softwares amplamente utilizados no ambiente educacional como LOGO, SCRATCH e LIGHTBOT.

## **2.1 Software LOGO**

O LOGO foi desenvolvido no Instituto de Tecnologia de Massachussets (MIT) por Seymour Papert, na década de 60. Uma das características mais importantes do LOGO é a de não possuir objetivo delimitado, ou seja, pode ser utilizada em ampla gama de atividades (FERRUZZI, 2001).

### 2.1.1 Pressupostos

A linguagem de programação LOGO foi concebida especialmente para a utilização das crianças a partir dos 5 anos, permitindo desse modo, o contato das mesmas com o computador. Nessa linguagem, uma tartaruga desempenha o papel central visto que é o ser a quem são transmitidos as instruções e os comandos dados. O aprendiz pode manipular a tartaruga de modo que esse processo auxilia na formação de um ambiente de aprendizagem ativa (PAPERT, 1980).

LOGO é uma linguagem de programação, ou seja, um meio de comunicação entre o computador e o seu usuário. O diferencial do LOGO e outras linguagens de programação consiste no fato de que foi desenvolvida para ser utilizada por crianças e para que estas possam, por meio dela, aprender outras coisas. A linguagem LOGO vem embutida em uma filosofia da educação não diretiva, inspirada por Piaget, na qual a criança aprende explorando o seu ambiente – mais especificamente, criando "microambientes" ou "micromundos" com regras impostas por ela mesma (PROJETO LOGO, 2009).

A “geometria da tartaruga”, que é utilizada no LOGO, é um estilo diferente de geometria. Tem-se um cursor que é representado por uma tartaruga dinâmica, que possui uma posição e, o que é muito importante, uma orientação. Esta “tartaruga” aceita ordens ou comandos que são fornecidos pelas crianças. Além disso, a linguagem utilizada no LOGO é uma linguagem procedural, isto é, é fácil criar novos termos ou procedimentos. Os comandos básicos são termos do cotidiano da criança (FERRUZZI, 2001).

Dentre os comandos básicos do LOGO, a tartaruga “sabe” executar certas tarefas utilizando as funções primitivas. Por exemplo, ela sabe andar para frente ao receber o comando PF  $n$  <Enter> o qual significa andar para frente  $n$  passos. O deslocamento da tartaruga é realizado a partir da posição onde a mesma se encontra e na direção da sua cabeça (RODRIGUES, 1993).

Quando se inicia o trabalho em LOGO, a tartaruga está “em casa” que trata-se do centro da tela e corresponde ao ponto (0,0) e orientação 0 graus. Outros comandos básicos são: PT  $n$  (andar para trás  $n$  passos), VD  $m$  (virar à direita  $m$  graus), VE  $m$  (virar à esquerda  $m$  graus) (RODRIGUES, 1993).

Após algum tempo de trabalho, a tela se torna com um aspecto um tanto confuso. Para limpá-la é necessário utilizar o comando LE que significa limpa a tela. Como primitiva pode-se citar também o comando de repetição que economiza muito trabalho em diversas ocasiões, como por exemplo Repete  $n$  [...] (RODRIGUES, 1993).

Com relação ao campo de movimento da tartaruga, a mesma está limitada a uma região de trabalho retangular ligeiramente inferior às dimensões da tela. Quando a mesma se desloca, deixa um traço na tela, como se um lápis fosse riscando o chão por onde ela se movimentasse. A tartaruga pode levantar o lápis por meio dos comandos LC (lápis em cima) e LB (lápis embaixo) (RODRIGUES, 1993).

Para voltar para a casa, a tartaruga tem um meio rápido que é no ponto de coordenadas (0,0) digitando EC (em casa). Além disso, a tartaruga poderá ou não ser visível utilizando-se os comandos ET (esconde tartaruga) e MT (mostra tartaruga). Também é possível que a tartaruga apague os traços existentes como se tivesse uma borracha na ponta (comando LA – lápis para apagar) (RODRIGUES, 1993).

### **2.1.2 Pesquisas com software LOGO**

Souza e Pereira (2012) apresentam resultados de um projeto de extensão realizado em duas escolas da rede pública de São João del-Rei. Após a execução do projeto, por meio do uso do software Kturtle nas aulas de linguagem LOGO, foi possível perceber que os alunos apresentaram, além de uma melhora expressiva em conceitos de informática, uma melhora significativa com relação à motivação e ao interesse em sala de aula. Os pesquisadores concluíram que a interação das escolas de ensino fundamental com a universidade por meio da robótica educacional contribuiu de maneira significativa na consolidação de conceitos relativos à informática, à figuras geométricas e à operações matemáticas, tendo como consequência melhorias no desempenho escolar dos alunos.

Ferruzzi (2001) apresenta resultados de uma pesquisa realizada no Instituto de Educação Infante Juvenil de Londrina. Afirma que o uso do LOGO no contexto de ensino-aprendizagem auxilia em noções de mensuração,

visto que quando os alunos começam a trabalhar com o deslocamento da tartaruga, são levados a compreender as relações mensuráveis, maior/menor, grande/pequeno. Além disso, o uso do LOGO privilegia atividades grupais que estimulam a descoberta e exploram as potencialidades dos sujeitos participantes. Conclui que a pesquisa se mostrou muito produtiva, visto que foi possível verificar a construção de processos de ensino-aprendizagem diferentes do estilo tradicional.

## **2.2 Software SCRATCH**

É fornecido gratuitamente e foi projetado pelo grupo Lifelong Kindergarten no Media Lab do MIT (SCRATCH, 2015). O SCRATCH é ideal para pessoas que estão começando a programar por não exigir o conhecimento prévio de outras linguagens de programação. Foi desenvolvida para auxiliar pessoas acima de 8 anos no aprendizado de conceitos matemáticos e computacionais. Com essa linguagem é possível criar histórias animadas, jogos e outros programas interativos.

### **2.2.1 Pressupostos**

O SCRATCH permite que seu usuário programe suas próprias histórias interativas, jogos e animações, além de permitir o compartilhamento de suas criações com outros membros da comunidade online. O SCRATCH auxilia os jovens a aprender a pensar de maneira criativa, refletir de maneira sistemática e trabalhar de forma colaborativa, habilidades essenciais que devem ser desenvolvidas durante sua educação formal e/ou informal. (SCRATCH, 2015).

Com o SCRATCH é possível misturar mídias, tais como imagens, gráficos, sons e músicas, para a criação, por exemplo, de histórias interativas, de jogos ou de animações (NETO, 2013). A programação é feita arrastando-se blocos de comandos que devem ser encaixados uns aos outros (FRANÇA et al., 2013).

De acordo com o Manual do Software (EDUSCRATCH, 2011), esta ferramenta educacional promove o desenvolvimento de diversas habilidades em seu aprendiz e usuário como fluidez tecnológica, gestão da informação,

comunicação e colaboração, criatividade e inovação, pensamento crítico, resolução de problemas e tomada de decisões.

O Software SCRATCH é considerado por especialistas um programa inovador, visto que permite exercitar a criatividade e o raciocínio científico, lógico e matemático. Fornece condições ao educando de ser projetista-criador da aprendizagem e do conhecimento no momento em que disponibiliza ferramentas de programação de informática simplificadas (MICHELON, 2012).

Michelon (2012, p. 21) afirma que:

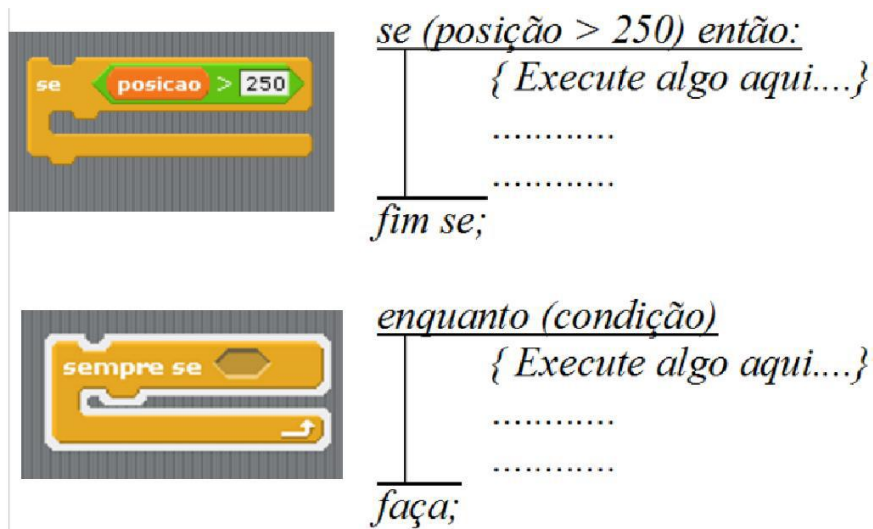
Através do Software SCRATCH educandos e educadores esboçam seus projetos e após exploração do ambiente e produção de conteúdos, podem disponibilizar em ambiente virtual suas construções. As produções disponibilizadas no site SCRATCH auxiliam outros alunos e professores que buscam conteúdos educacionais e recursos pedagógicos que auxiliam em sala de aula.

Nesse mesmo sentido, Mélo et al. (2011) apontam o SCRATCH como uma excelente ferramenta para o ensino de conceitos de Lógica de Programação visto que apresenta uma interface visual amigável e simples levando, de forma intuitiva, às principais estruturas de uma linguagem como: variáveis, operadores, estruturas de decisão e de repetição, e outros.

Por não trabalhar com linhas de código (usa-se somente interface), é possível realizar a criação de programas de maneira mais simples e dinâmica, além de estimular o raciocínio lógico, e de permitir a visualização gráfica da execução do programa criado (NETO, 2013).

Um script do SCRATCH guarda similaridade com formas tradicionais de estruturas de algoritmo, com a identificação caracterizada nos blocos de estruturas de decisão e de repetição. Na Figura 2 mostram-se aplicações dos blocos “se ... senão” e o “sempre se...” do SCRATCH, com as suas formas correspondentes em algoritmo ("Português Estruturado"). O conceito de variáveis e matrizes unidimensionais também é introduzido de forma muito intuitiva, com comandos para a sua criação e blocos para operações básicas sobre elas.

Figura 2: Blocos do SCRATCH e fragmentos de algoritmo.



Fonte: Neto (2013).

## 2.2.2 Pesquisas com software SCRATCH

Dullius (2008) pesquisou sobre as possibilidades de utilização do SCRATCH no Ensino fundamental com ferramenta de apoio para a aprendizagem. O estudo foi desenvolvido com um grupo de professores de uma escola municipal por meio da realização de oficinas. Os relatos dos sujeitos mostraram que o software seria uma ferramenta interessante para ser explorada com os alunos nos laboratórios de informática, sendo que essa exploração pode contemplar diferentes componentes curriculares se mostrando uma alternativa desafiadora e divertida de apoio no processo de aprendizagem.

Silva et al. (2014) relatam uma experiência vivenciada na aplicação de oficinas para professores da rede estadual e municipal do Rio Grande do Norte. Visaram, com essas oficinas, oportunizar aos educadores a utilização do software SCRATCH como ferramenta auxiliar na prática docente e também de atividades lúdicas para ensinar computação. Os resultados foram significativos sendo que as atividades auxiliaram o aprendizado de conceitos de fundamentos da computação. Além disso, as oficinas fortaleceram os vínculos de interação entre professores e alunos de forma lúdica e interativa, reforçando a troca de conhecimentos decorrentes dessas interações.

Em virtude da grande evasão de alunos do Proeja Informática (educação de jovens e adultos, ensino médio e ensino técnico) do Sertão-PE, principalmente devido às disciplinas de Lógica de Programação, Alencar et al. (2014) sugeriram uma proposta para o alto índice de reprovação nessa disciplina. Para isso, utilizaram o software SCRATCH que segundo eles, apresenta como ponto inovador o fato de usar novos paradigmas de programação sendo de fácil entendimento. Ao final da intervenção observaram que os alunos desenvolveram o jogo proposto com facilidade e, diante disso, concluíram que o uso do software foi eficaz para o ensino/aprendizagem de conceitos básicos de programação, fixando o conteúdo de forma mais dinâmica e em um nível de compreensão mais lúdico.

## **2.3 Software LIGHTBOT**

O LIGHTBOT foi criado em 2008 por Daniel Yaroslavski, estudante de graduação da Universidade de Waterloo no Canadá. Posteriormente, reconstruiu-se uma nova versão para iOS e Android que é uma maneira fácil para auxiliar as crianças a aprender conceitos como loops. O jogo começou como uma espécie de jogo de Q-Bert, que permite que você conduza um pequeno robô para cima e para baixo em uma série de blocos. Mais tarde, Yaroslavski acrescentou aspectos mais educacionais, incluindo a necessidade de usar conceitos de programação para resolver quebra-cabeças (BIGGS, 2013).

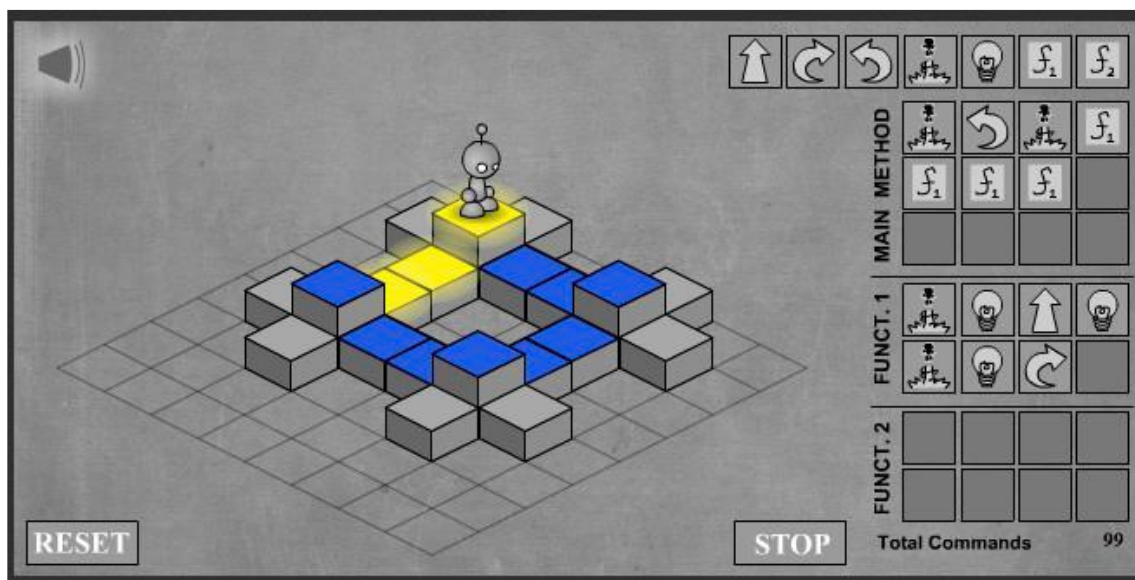
### **2.3.1 Pressupostos**

Santos (2010) afirma que vale citar o LIGHTBOT como exemplo de jogo educativo (Figura 3). O objetivo deste jogo é mover um robô por um tabuleiro e ligar certas casas deste tabuleiro. Ao ligar todas as casas marcadas o jogador avança de nível. Para isso o jogador deve criar um “script” utilizando uma quantidade restrita ações, de um conjunto de comandos disponíveis. Pode-se usar uma função “*main*” e duas funções adicionais ( $f_1$  e  $f_2$ ). Também é possível usar condicionais e recursividade e outros conceitos de programação.



Segundo Yaroslavski, sua criação é um jogo de vídeo game com uma natureza mais atraente e menos óbvia para o ensino da ciência da computação, mascarando conceitos como mecânica de jogo e focando mais na lógica de programação do que na codificação direta. Afirma que os professores dos Estados Unidos da América, da Rússia e de outros países tem usado o LIGHTBOT em suas salas de aula para introduzir conceitos de programação. Ainda segundo Yaroslavski, é a maneira perfeita de deixar as crianças experimentarem a programação sem sentá-los na frente de uma tela de texto (BIGGS, 2013).

Figura 3: Jogo LIGHTBOT.



Fonte: Santos (2010).

Enquanto o LOGO e o SCRATCH são linguagens de programação e não possuem um objetivo delimitado, o LIGHTBOT é um jogo que possui um objetivo que deve ser concluído para passar de fase. Em relação à forma de se programar, o LIGHTBOT é programado por meio de botões de ação e há uma limitação no uso destes botões que variam de acordo com a fase a ser jogada, o LOGO é programado utilizando uma linguagem escrita e simplificada e o SCRATCH utilizando blocos de montar.

## 2.4 LEGO MINDSTORMS

Os kits MINDSTORMS fazem parte de uma linha do brinquedo LEGO®, lançada comercialmente em 1998 e voltada para a educação tecnológica. Inicialmente, o projeto foi inspirado por Seymour Papert e desenvolvido em uma parceria entre o Media Lab do Massachusetts Institute of Technology (MIT) e o Lego Group (FRIEDRICH et al., 2012).

### 2.4.1 Pressupostos

Segundo Friedrich et al. (2012, p.03):

O sistema *LEGO MINDSTORMS* é formado por quatro tipos de sensores, três motores e um controlador central, além de um conjunto de peças da linha tradicional (tijolos cheios, placas, rodas) e da linha *Lego Technic* (tijolos vazados, motores, eixos, engrenagens, polias e correntes). Cada componente tem suas funcionalidades específicas: os motores são os responsáveis por movimentar a estrutura das montagens; os sensores são os responsáveis pela coleta de informação junto ao meio externo; o controlador central, também conhecido por módulo RCX (*Robotic Command Explorer*) é responsável pela parte inteligente, contendo o *software* que gerencia o sistema.

Além desses sensores, o conjunto LEGO MINDSTORMS inclui peças de montar, permitindo criar humanoides, veículos, guindastes e animais (Figura 4).

Figura 4: Possibilidades de robôs projetáveis do sistema LEGO.



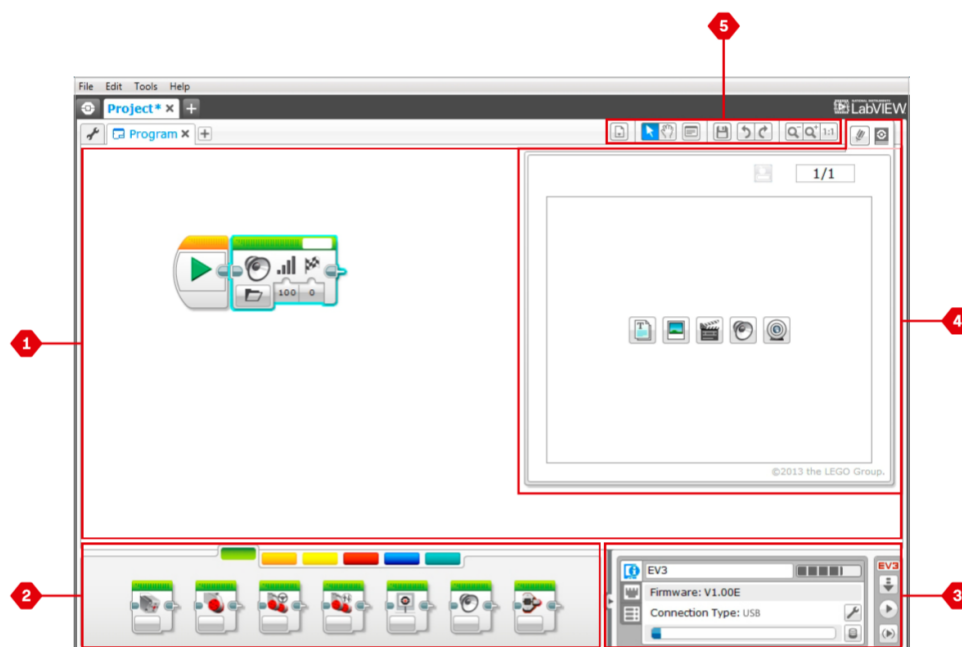
Fonte: Indy Telecom & Industrial Media (2006) apud Guedes e Kerber (2010).

Os conjuntos do LEGO MINDSTORMS são utilizados, além da função lúdica, para o desenvolvimento de projetos de pequeno e médio porte,

estimulando a criatividade e a solução de problemas do cotidiano por parte dos alunos (FRIEDRICH et al., 2012).

Segundo Knudsen (1999), o kit LEGO MINDSTORMS é um dos dispositivos robóticos de baixo custo mais populares. Price et al. (2003) defendem que a disponibilidade de robôs LEGO MINDSTORMS, pequenos e de preço acessível viabiliza o desenvolvimento de projetos de robótica simples mesmo quando desenvolvidos de maneira independente. Isto se deve, em parte, ao fato dos robôs serem objetos físicos cujos projetos de construção apresentam resultados observáveis. A Figura 5 mostra o software EV3 que faz parte do Kit LEGO MINDSTORMS.

Figura 5: Tela do software EV3 do Kit LEGO MINDSTORMS.



Fonte: Guia do Usuário – LEGO MINDSTORMS EV3.

O ambiente de programação EV3 consiste nas seguintes áreas principais:

1. Área de programação: local onde o programa será montado através do encaixe de blocos de programas.
2. Paleta de programação: são os blocos de construção para o programa.
3. Área de hardware: onde ocorre a ligação dos componentes do robô como sensores e motores

4. Editor de conteúdo: um caderno integrado ao software onde é possível obter instruções ou documentar o projeto usando textos, imagens e vídeos.
5. Barra de ferramentas: ferramentas básicas para trabalhar com o programa.

O software EV3 possui diversos blocos programáveis divididos em categorias: blocos de ação, blocos de fluxo, blocos de sensor, blocos de dados, blocos avançados e blocos mais utilizados. Isso permite criar diversas sequências de programação, fazendo com que o LEGO MINDSTORMS possa se comportar de distintas formas possíveis.

A programação consiste no encaixe de blocos funcionais semelhantes ao do SCRATCH. Para criar um programa é só clicar no bloco desejado e arrastá-lo até a área de programação de forma a conectá-los com o bloco de start. Basta adicionar quantos blocos forem preciso que o programa será executado de acordo com a ordem em que seus blocos estão dispostos na tela, da esquerda para a direita (MANDAI, 2014)

#### **2.4.2 Pesquisas com software MINDSTORMS**

Guedes e Kerber (2010), com o intuito de promover o conhecimento de novas metodologias e práticas de aprendizagem para professores e para a comunidade escolar em geral, objetivaram a inserção da robótica no meio educacional. Desenvolveram a pesquisas nas disciplinas de Matemática e Ciência, tendo como base o robô LEGO MINDSTORMS, que foi testado em uma escola de Ensino Fundamental São Lourenço. Concluíram após a experiência, que o uso do conjunto Lego Mindstorm NXT levou os alunos ao desenvolvimento da criatividade e imaginação na procura de soluções para a realização dos experimentos.

Friedrich et al., (2012), propõem uma metodologia pedagógica para ambientação e inserção à lógica de programação para crianças de sete a dez anos de idade, utilizando como meio auxiliar de ensino o programa LOGO e o Projeto LEGO MINDSTORMS. A pesquisa foi realizada em uma turma de uma

escola pública de Cachoeira do Sul – Rio Grande do Sul, e os resultados obtidos mostraram que todas as atividades do plano de ensino foram concluídas com êxito pelas crianças, que se mostraram receptivas e felizes por estarem participando do trabalho.

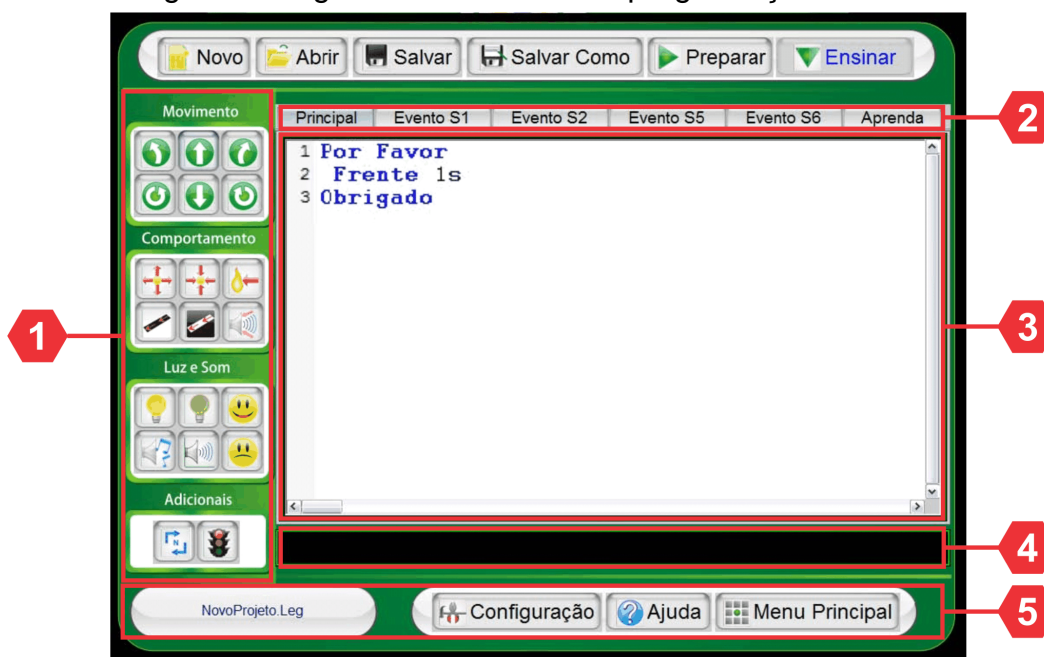
## 2.5 LEGAL

O ambiente de desenvolvimento LEGAL foi desenvolvido pela PNCA Robótica e Eletrônica e faz parte de um kit de desenvolvimento conhecido como KIT ALFA, que se trata de um sistema modular para projetos de robôs, dispositivos eletrônicos, mecânicos e computacionais.

### 2.5.1 Pressupostos

LEGAL é uma interface e uma linguagem de programação que usa uma sintaxe em Português, o que facilita seu uso no Brasil. Foi criada para estimular o raciocínio lógico, respeitando a capacidade cognitiva de usuários em diferentes faixas etárias. Segundo o próprio desenvolvedor, essa linguagem é ideal para ser utilizada no ensino. O ambiente de programação LEGAL, pode ser dividido em cinco regiões conforme ilustra a Figura 6.

Figura 6: Regiões do ambiente de programação LEGAL.



Fonte: <http://www.unoeste.br/Fipp/robotica/programacao.aspx>

1. Paleta de botões com ícones autoexplicativos dos comandos de movimento, comportamento, luz e som e adicionais da linguagem LEGAL.
2. Barra de botões que servem para gerenciar os arquivos, preparar e enviar os programas para o módulo de controle.
3. Região conhecida como área de programação, onde os códigos são escritos.
4. O “quadro” negro é usado para informar possíveis erros do programa.
5. A primeira barra contém o botão para o nome do programa que está sendo escrito e a segunda barra os botões de comando de configuração, ajuda e menu principal.

A linguagem de programação LEGAL é notação formal que descreve algoritmos a serem executados pelo MC2, que é o cérebro de seu robô.

A linguagem LEGAL é uma linguagem de programação, por isso ela possui as principais estruturas comuns a todas as linguagens como: comandos condicionais, comandos de repetição, entre outros (PNCA, 2016).

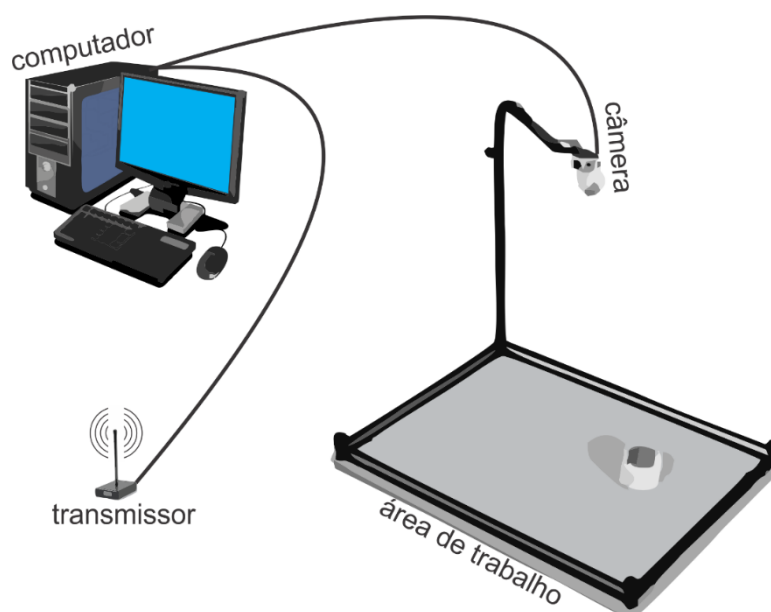
Além disso, a linguagem Legal também tem características próprias voltadas para o controle de robôs e outros dispositivos mecatrônicos. Entre estas características pode-se destacar: comandos para controlar motores e servos motores; comandos para fazer a leitura de sensores externos; comportamentos inteligentes pré-programados; comandos para coletar e armazenar dados remotamente (PNCA, 2016).

## **2.6 AEDROMO**

O AEDROMO - Ambiente Experimental e Didático com Robôs Móveis - é formado por uma área de trabalho, objetos passivos, um ou mais robôs, dois computadores, uma câmera global do tipo *webcam* e um transmissor (Figura 7). As premissas na concepção e desenvolvimento deste ambiente são a flexibilidade de adaptação, uso por várias disciplinas e o baixo custo. Os experimentos, neste ambiente, podem ser motivados para fins de pesquisa, aprendizagem ou, entretenimento (FERASOLI et al. 2006). Destaca-se ainda

que o AEDROMO pode ser programado em qualquer linguagem que suporte sockets.

Figura 7: Ambiente do AEDROMO.



Fonte: Autor.

Existem dois tipos de computadores na arquitetura do AEDROMO: o Computador Servidor e o Computador Aluno. O primeiro é responsável pela implementação do sistema de visão computacional e pela comunicação com os robôs. Por outro lado, no computador aluno é onde o software de controle dos robôs será desenvolvido pelo usuário (ALVES et al. 2011).

O uso de uma câmera global confere grande versatilidade ao sistema, uma vez que permite a pequenos robôs reagirem com o ambiente de forma complexa, sem a necessidade de inclusão de sensores sofisticados. Ao mesmo tempo, ela atribui ao ambiente uma ampla variedade de aplicações pedagógicas tanto para ensino quanto para pesquisa (SOUSA, 2015)

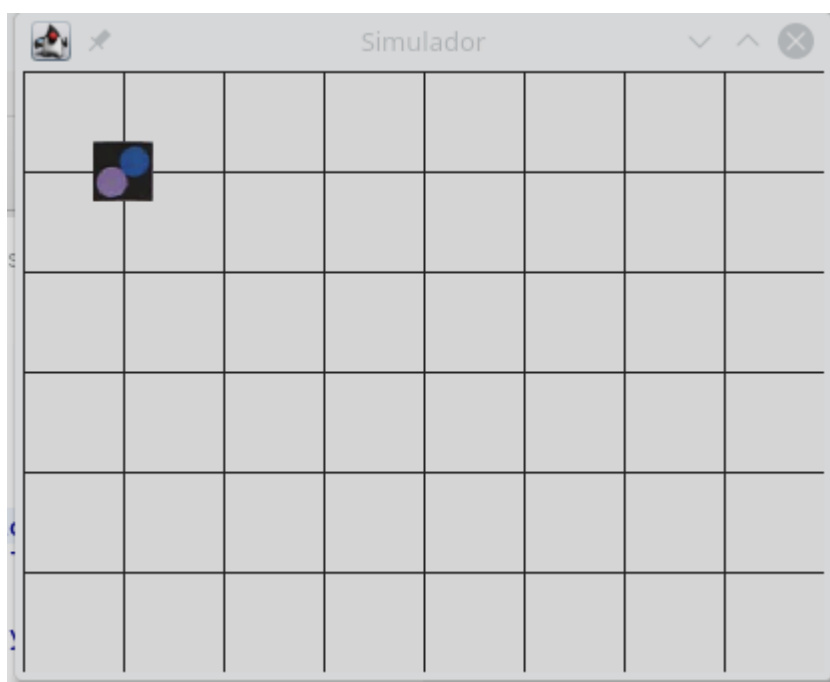
A câmera global captura as imagens e as envia ao computador servidor. Neste computador, as imagens são adquiridas e processadas. O processamento determina as posições cartesianas ( $x, y$ ) e o ângulo  $\theta$  dos objetos (robôs, bolas, blocos, etc.) inseridos na área de trabalho (ALVES et al. 2011).

Foram adotados marcadores coloridos para facilitar o reconhecimento do robô. Esses marcadores consistem em duas etiquetas de cores diferentes coladas sobre os robôs (ALVES et al. 2011).



Além do servidor que permite controlar os robôs, o AEDROMO também conta com um servidor de simulação, cuja finalidade é melhorar sua acessibilidade por não necessitar do hardware físico. Graças a arquitetura cliente-servidor adotada, o código do aluno (cliente) não necessita ser alterado para que funcione com o simulador (servidor) (Figura 8), pois o protocolo de comunicação é o mesmo utilizado com o servidor dos robôs reais.

Figura 8: Simulador.



Fonte: Autor.

Como o AEDROMO permite que um software do aluno seja desenvolvido em qualquer linguagem que suporte sockets, dentre as diversas existentes, optou-se pelo Processing pelas vantagens apresentadas por Shimabokuro e Pegoraro (2014). O Processing, além de ser uma linguagem de programação, é também um ambiente de desenvolvimento integrado de código aberto que entra na categoria de software livre, capaz de gerar arquivos executáveis para vários sistemas operacionais como Windows, MAC OS, Linux e Android. Tem como base a linguagem JAVA sendo que usa sua mesma sintaxe.

No levantamento bibliográfico realizado em pesquisas e artigos publicados relacionados ao uso dos computadores no ensino de lógica e



programação. Foi possível encontrar vários ambientes diferentes para o ensino de lógica e programação, sendo que alguns como o LOGO e SCRATCH necessitavam apenas do computador para a sua utilização, enquanto outros como o LEGAL e o MINDSTORMS utilizavam-se de acessórios robóticos para operarem em conjunto com o computador.

O Quadro 1 é comparativo entre trabalhos pesquisados e o LOGIKID.

<b>SISTEMA</b>	<b>O QUE É?</b>	<b>PÚBLICO</b>	<b>PROGRAMAÇÃO</b>	<b>OBJETIVO</b>
<b>LOGO</b>	Linguagem de programação	A partir dos 5 anos de idade	Linguagem escrita	Ensinar conceitos básicos de programação
<b>SCRATCH</b>	Ambiente de desenvolvimento	A partir dos 8 anos de idade	Blocos de encaixe	Ensinar conceitos básicos de programação
<b>LIGHTBOT</b>	Jogo eletrônico	A partir dos 4 anos	Botões de comandos	Introduzir conceitos básicos de programação
<b>EV3 – LEGO MINDSTORMS</b>	Ambiente de desenvolvimento/Kit robótico	A partir dos 7 anos de idade	Blocos de encaixe	Viabilizar o desenvolvimento de projetos de robótica simples
<b>LEGAL</b>	Ambiente de desenvolvimento/Kit robótico	Não encontrado	Linguagem escrita	Viabilizar o desenvolvimento de projetos de robótica
<b>LOGIKID</b>	Software para o ensino de lógica e programação	A partir dos 7 anos de idade	Botões de escolhas	Auxiliar o ensino de lógica e programação no ensino fundamental e médio

Quadro 1 – Características comparativas entre os trabalhos pesquisados e o LOGIKID.

## CAPÍTULO 3 – LOGIKID

### 3.1 Estrutura do LOGIKD

O LOGIKID segue a mesma ideia do LOGO de comandar um agente por meio de um programa escrito numa linguagem simples. No entanto, por utilizar o AEDROMO, o agente do LOGIKID é um robô específico. O emprego dos robôs físicos aproxima o LOGIKID de soluções como o LEGOMINDSTORM, uma iniciativa motivadora para o uso da robótica, e o LEGAL, que possui uma linguagem de programação em português. O desenvolvimento da linguagem do LOGIKID também foi influenciado pelo software LIGHTBOT, que utiliza funções e laços simplificados, e pelo software SCRATCH, que utiliza uma sintaxe simples e de fácil entendimento baseada em programação procedural. Estes dois últimos softwares, apesar de não fazerem parte do software atual, nortearam o desenvolvimento, para que suas características sejam concluídas futuramente.

Como as pesquisas para o ensino utilizando a robótica envolvendo o LEGO MINDSTORMS são muito favoráveis, uma vez que usa robôs reais, optou-se pelo uso do AEDROMO que pode ser usado tanto com robôs reais como no ambiente virtual (simulador).

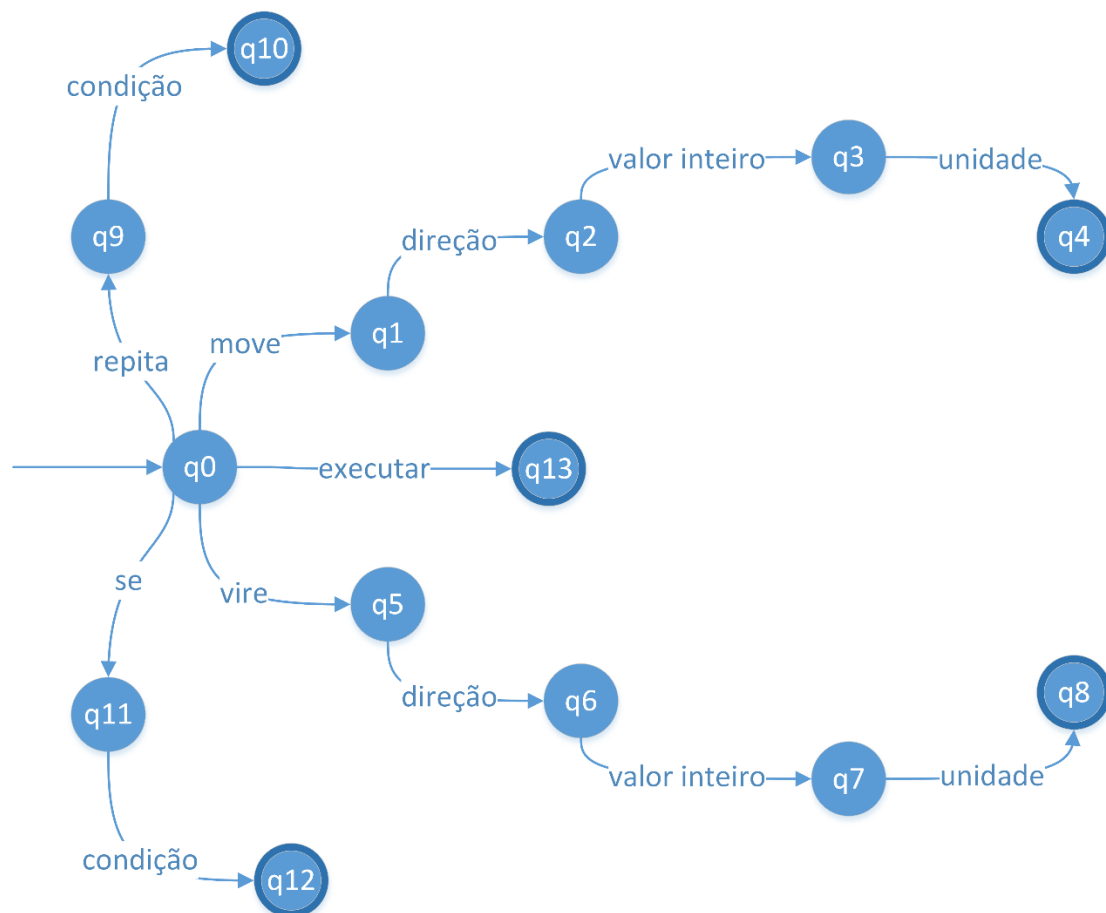
Em relação ao LIGHTBOT, por apresentar funções e laços simplificados, o LOGIKID seguiu essa mesma ideia, simplificando a sua sintaxe quando se utiliza os botões para as ações.

O LOGIKID interage com o usuário através de perguntas feitas pelo software e respostas dirigidas escolhidas pelo usuário. Exemplifica-se um tipo de funcionamento: o programa irá destacar uma pergunta, por exemplo, “o que você deseja fazer”. O aluno irá fazer uma escolha de acordo com os botões que aparecem na tela e após a sua escolha outra tela irá aparecer, como por exemplo se ela escolher “frente”, aparecerá uma tela de “por quanto tempo?”. Diante disso o aluno escolherá o tempo. Consecutivamente o aluno fará as suas escolhas e quando terminar poderá clicar no botão de ação e o robô irá fazer o que ela mandou.

O diagrama da Figura 9 é um autômato finito determinístico com 14 estados (q0 até q13), onde o estado inicial é o q0 e os estados finais são q4, q8, q10, q12 e q13. As transições entre os estados representam a leitura de um

símbolo de entrada. Os comandos determinados pelo usuário são definidos a partir do estado inicial (transação do estado q0 para os estados q1, q5, q9 e q11). Em seguida são preenchidos os parâmetros (demais transações) dos respectivos comandos até um dos estados finais. O autômato é executado toda vez que o usuário realiza uma ação. Desta forma, os comandos são alocados em uma fila até serem executados quando se lê o símbolo "executar". Os comandos "se" e "repita" não foram implementados no projeto atual.

Figura 9: Diagrama de estados.



Fonte: Autor.

### 3.2 Palavras reservadas e rotinas do LOGIKID

As palavras reservadas e rotinas para a linguagem do LOGIKID são apresentadas nos Quadros de 2 a 4.

<b>Palavra Reservada</b>	<b>Significado</b>
cm, centímetros	indica que a unidade é “centímetros”
s, seg, segundos	indica que a unidade é “segundos”
graus	indica que a unidade angular é “graus”
rad, radianos	indica que a unidade angular é “radianos”
bola	objeto bola, com posição (x,y)
obstaculo	objeto, com posição (x,y)
gol	objeto gol, com posição (x,y)
borda	objeto borda, consiste nos limites do campo
robo	posição (x,y) do robô.
frente	indica o sentido e direção (frente do robô)
atras	indica o sentido e direção (atrás do robô)
esquerda, anti-horario	faz um giro de à esquerda do robô, mas precisa dizer o quanto o robô irá girar neste sentido.
direira, horário	faz um giro de à direita do robô, mas precisa dizer o quanto o robô irá girar neste sentido.
direção	armazena a direção do robô em relação a arena do AEDROMO
termina	finaliza o laço e pula para próxima linha depois dele
parado	estado do robô em repouso
fim	finaliza o programa

Quadro 2: Palavras reservadas para o LOGIKID.

<b>Operadores lógicos</b>
nao, !
e, &&
ou,
ou_exclusivo
maior, >
menor, <
maior_ou_igual, >=
menor_ou_igual, <=
diferente, !=
igual, ==

Quadro 3: Operadores lógicos.

<b>Rotina</b>	<b>Significado</b>
<b>move dir qtd unid</b>	move o robô para a direção escolhida por determinada quantidade de tempo ou distancia
<b>repita</b> {comandos} <condição> <b>volta</b> ou <b>termina</b>	Repete uma sequência arbitrária de comandos
<b>vire dir qtd unid</b>	vira o robô para a direção escolhida por determinados graus ou radianos (no eixo central do robô)
<b>gire dir qtd,unid</b>	sinônimo de vire
<b>vai x y</b>	vá para a posição xy da arena
<b>se cond se não fim</b>	se condição então

Quadro 4: Rotinas.

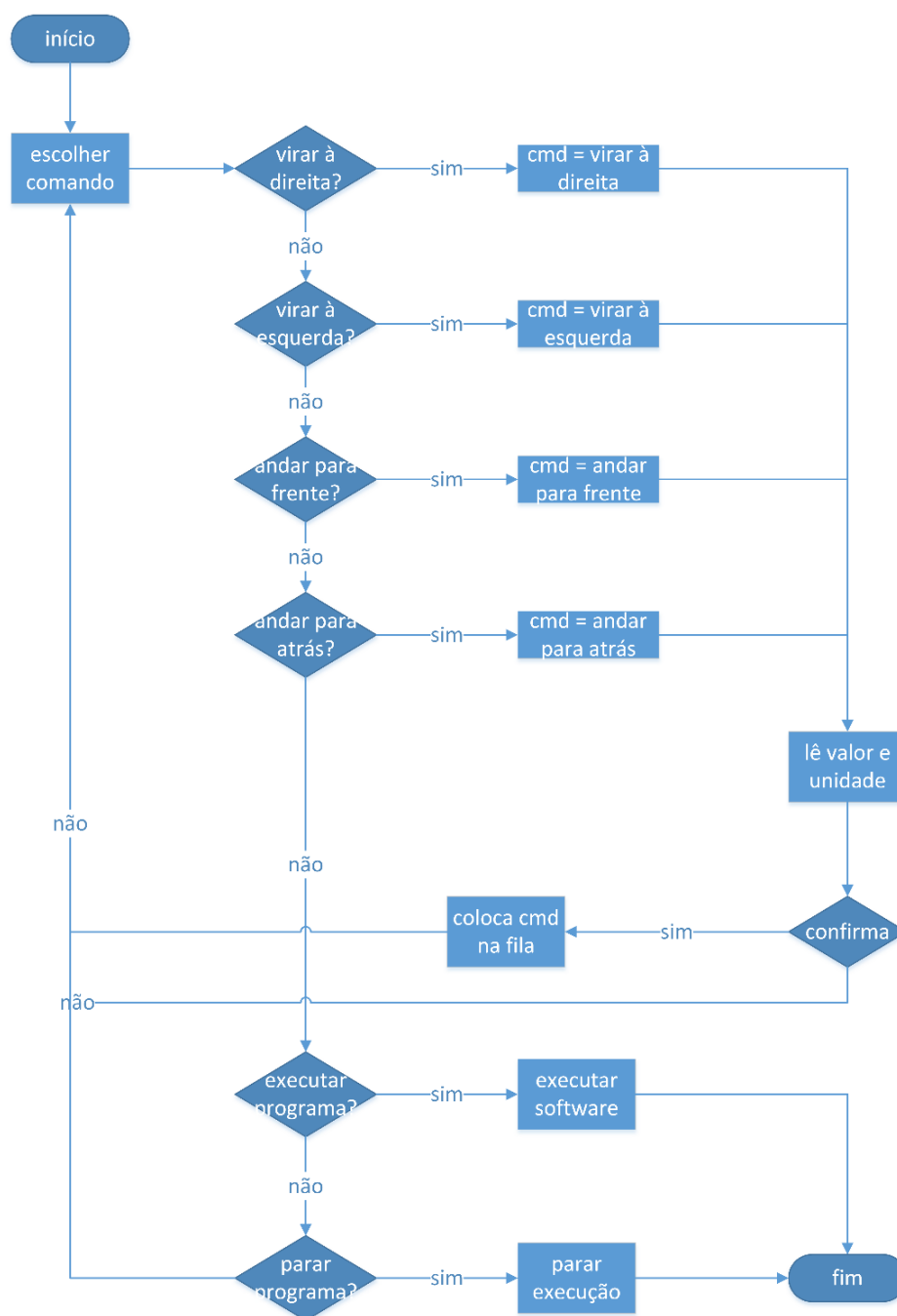
Essas rotinas são salvas pelo LOGIKID em um vetor de caracteres. Ao ser executado o código, são convertidas para a linguagem do interpretador e salvas em um arquivo de rotina com a extensão .rot que será utilizada pelo interpretador. O interpretador executa essas rotinas realizando as funções a serem utilizadas pelo robô controlando a comunicação com o AEDROMO.

### 3.3 Interface do LOGIKID

A interface do LOGIKID com o usuário possui uma tela interativa, para uma melhor interação com o aluno. Seu objetivo principal é realizar – por meio das escolhas dos alunos - movimentos por um robô dentro de um campo visual.

A lógica de comandos refere-se à utilização de botões de escolha do usuário.

Figura 10: Fluxograma dos comandos dos botões.



Fonte: Autor.

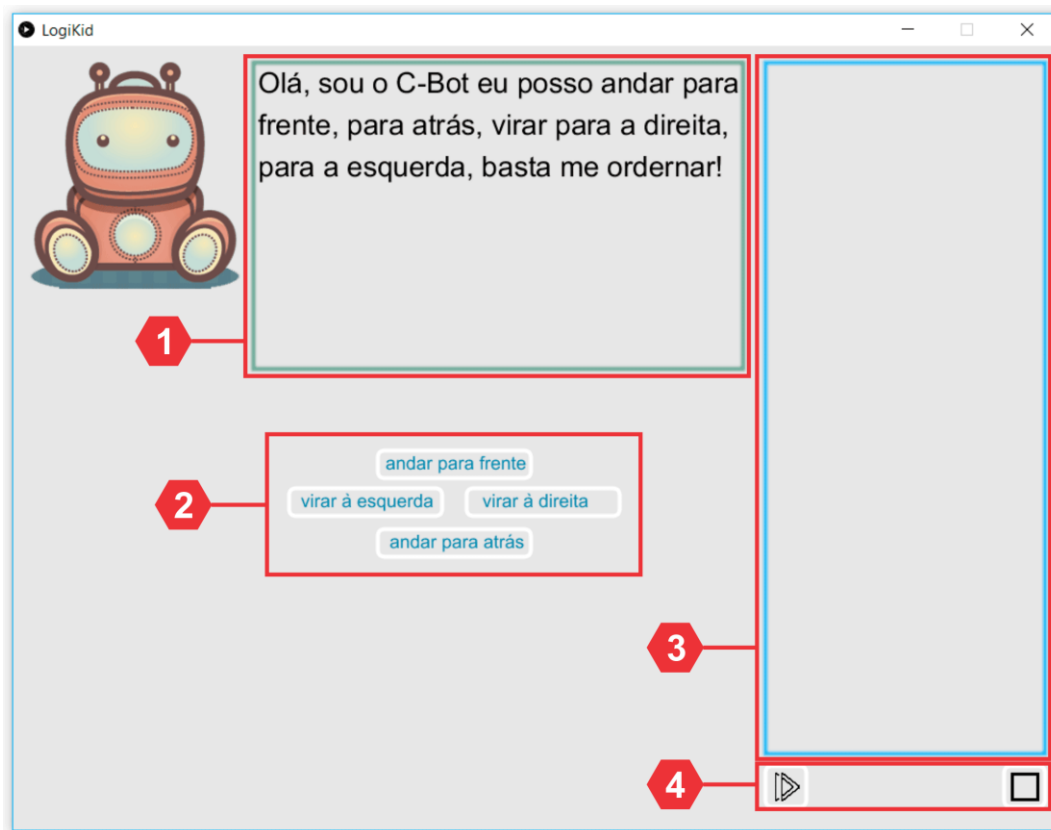
O fluxograma (Figura 10) refere-se aos comandos dos botões que funcionarão no LOGIKID.

Quando se escolhe algum comando, o LOGIKID deverá fazer perguntas e o aluno respondê-las até chegar a hora da execução do programa, onde pode-se esperar a sua conclusão ou parar a sua execução.

As escolhas dos alunos podem ser efetuadas por meio de botões, sendo que após cada clique, surgirá uma pergunta e após confirmada a sua resposta, o programa irá gerar uma linha código e mostra-lo na área de códigos ao lado.

A interface é demonstrada na Figura 11.

Figura 11: Interface desenvolvida no Processing.



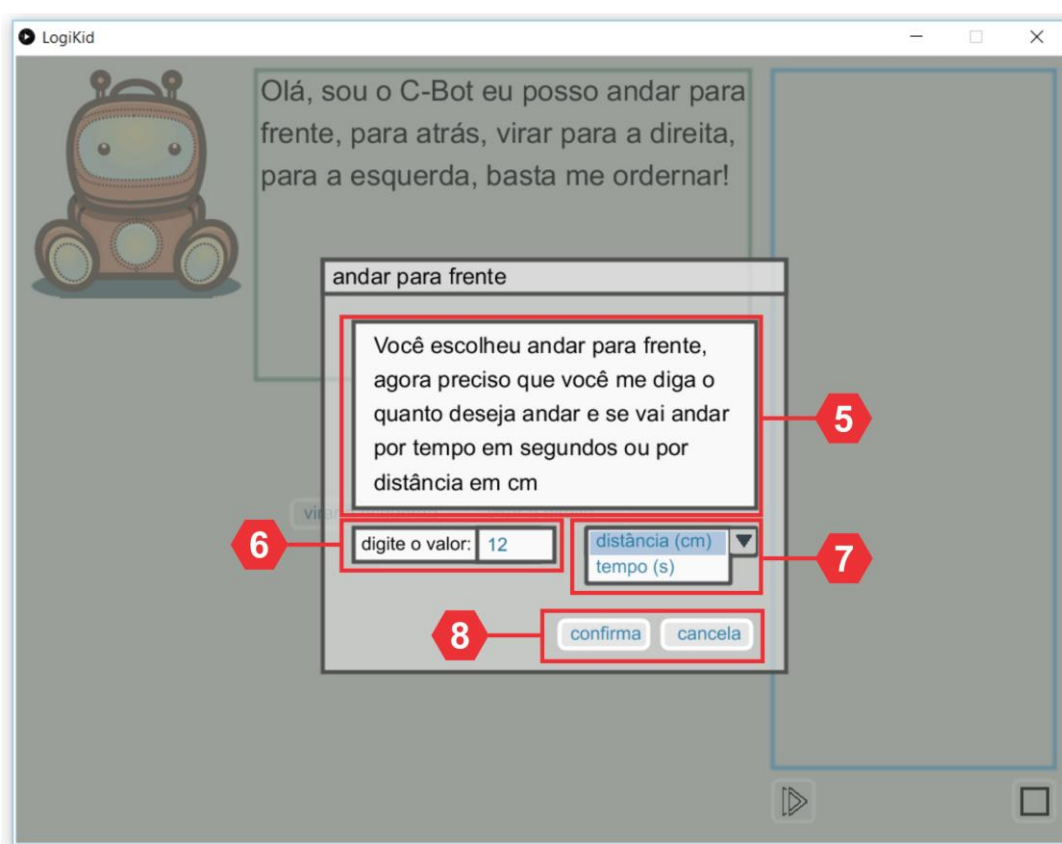
Fonte: Autor.

1. Área de texto informativo e de perguntas: local onde o LOGIKID irá iniciar o “diálogo” com o aluno.
2. Botões de comandos: são os botões responsáveis pelas decisões de movimentação e giro do robô escolhidas pelo aluno.
3. Área de códigos: Após cada escolha confirmada pelo aluno, um código de programação (na linguagem do LOGIKID) representando o botão escolhido irá aparecer.
4. Botões para executar ou parar o programa.

Cada sequência de perguntas gera uma linha de comandos. Essa interação poderá ser repetida diversas vezes até que todos os comandos necessários para a realização da tarefa sejam completamente especificados.

Após completar o código, o aluno poderá clicar no botão iniciar e o robô irá se movimentar de acordo com as suas escolhas. Escolhendo-se por exemplo o comando “andar para frente”, abre-se uma janela a qual o aluno escolherá um valor e uma unidade de medida que poderá ser tempo (s) ou distância (cm) (Figura 12).

Figura 12: Interface com opção de comando "andar para frente"



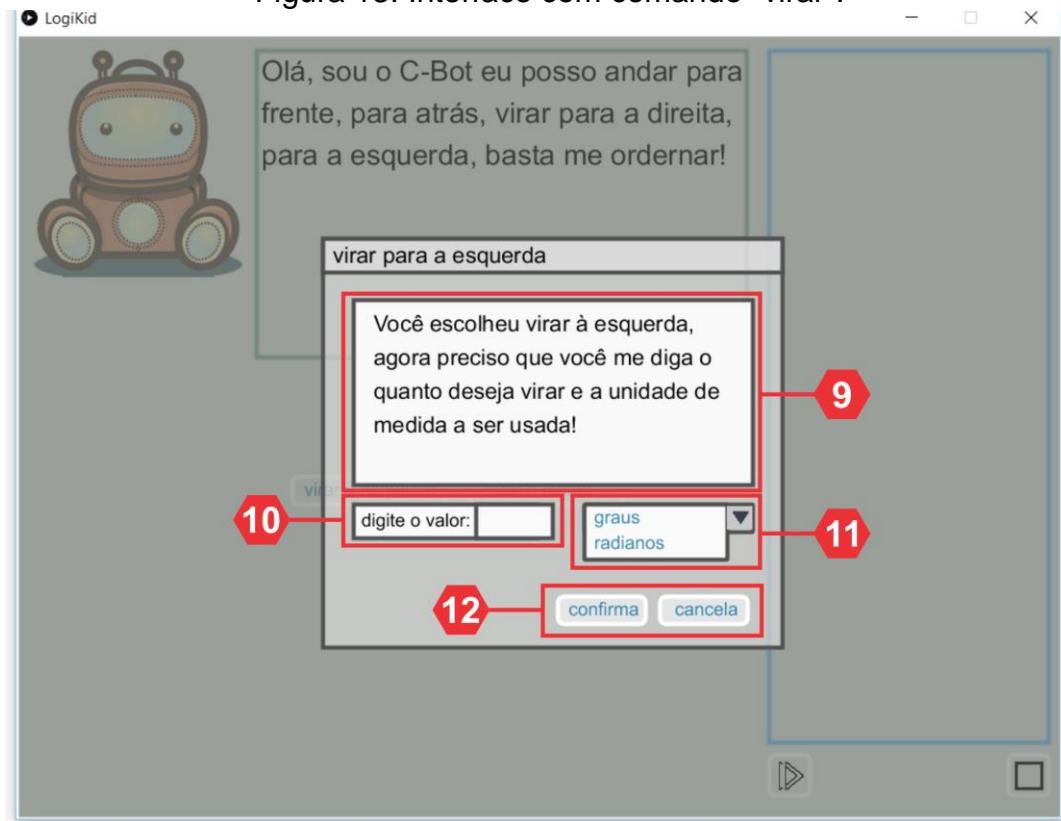
Fonte: Autor.

5. Área de texto informativo e perguntas: local onde há “diálogo” entre o LOGIKID e o aluno.
6. Valor (número inteiro) da unidade a ser escolhida.
7. Combobox: para selecionar o tipo de unidade a ser usada na movimentação.
8. Botões para confirmar ou cancelar a escolha.



Escolhendo-se por exemplo o comando “virar”, abre-se uma janela a qual o aluno escolherá um valor e uma unidade de medida que poderá ser graus ou radianos (Figura 13).

Figura 13: Interface com comando "virar".

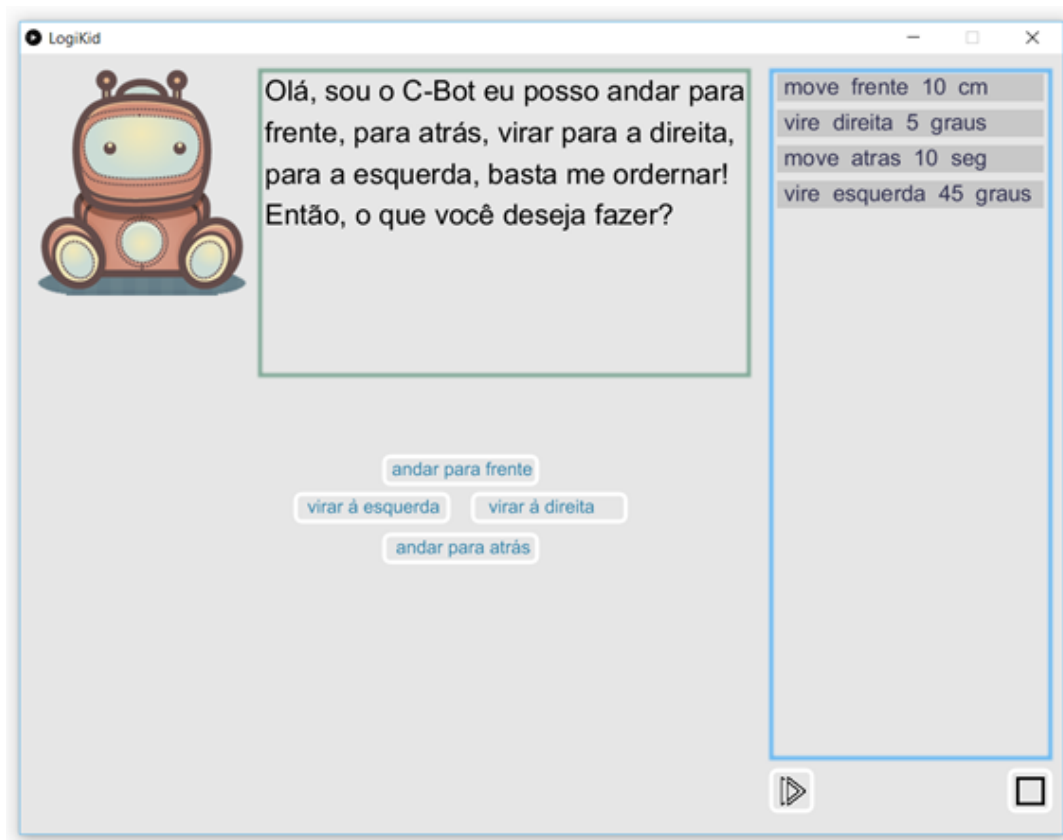


Fonte: Autor.

9. Área de texto informativo e perguntas: local onde há “diálogo” entre o LOGIKID e o aluno.
10. Valor (número inteiro) da unidade a ser escolhida.
11. Combobox: para selecionar o tipo de unidade a ser usada no giro do robô.
12. Botões para confirmar ou cancelar a escolha.

Após determinar-se a sequência de comandos que o robô irá executar, deve-se clicar no botão executar. A partir daí o robô irá executar os passos linha por linha. Se o botão de “parar” for clicado o robô irá parar qualquer execução, não sendo possível continuar de onde parou. A Figura 14 a mostra a interface com 4 rotinas a serem executadas.

Figura 14: Interface após alguns comandos.



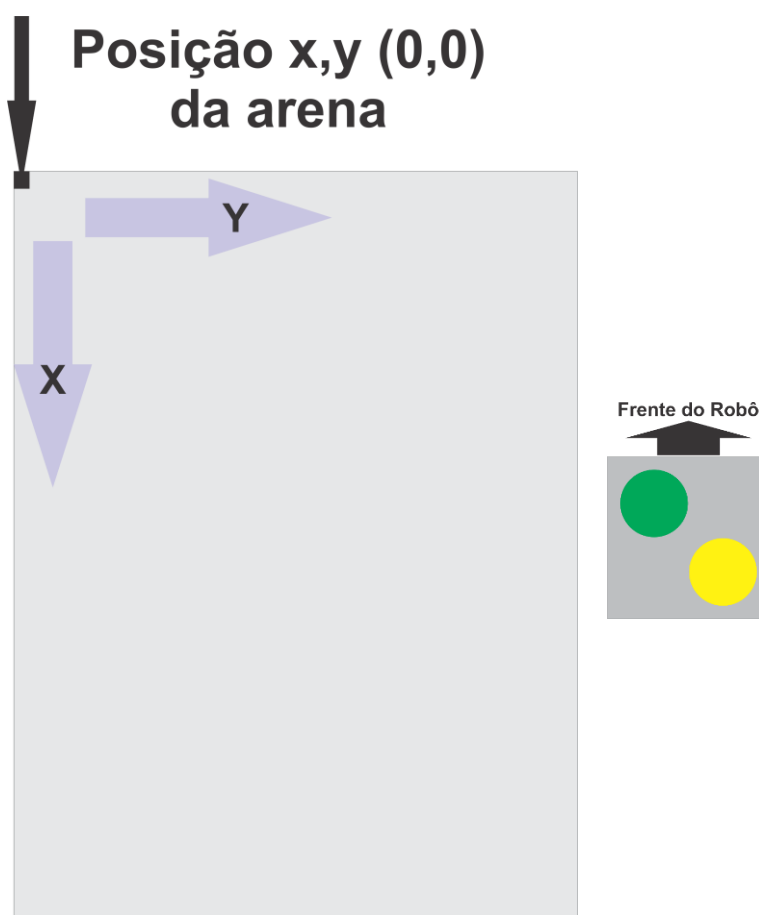
Fonte: Autor.

O software apresenta fases ou desafios de lógica com níveis de dificuldade variáveis. O aluno poderá usar os botões ou o código e ainda poderá ver uma lista de códigos para adicionar em sua respectiva aba, sendo que sempre que clicar em um botão, seu código será gerado.

## CAPÍTULO 4 – EXPERIMENTOS REALIZADOS

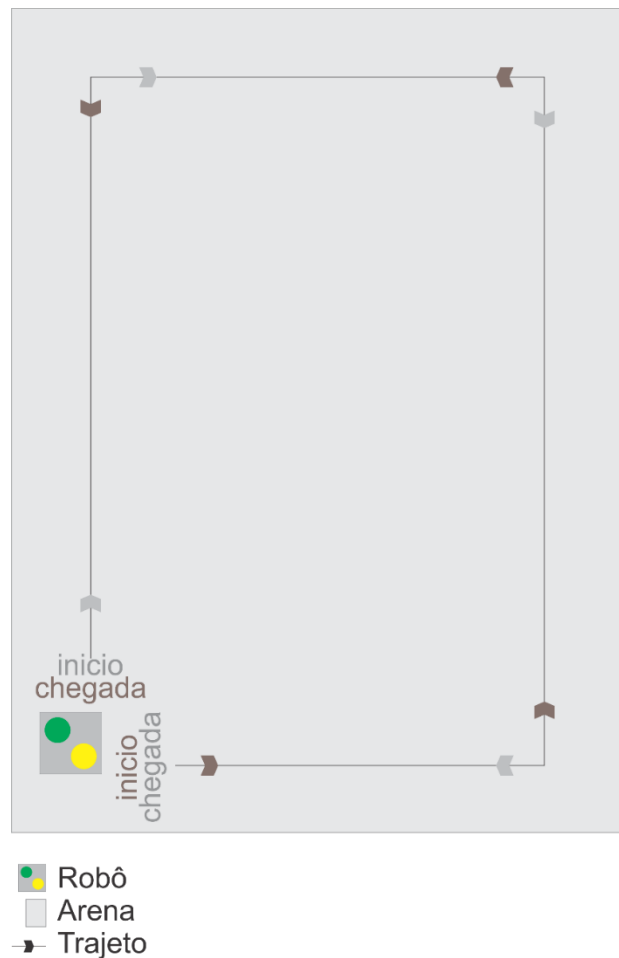
A seguir são apresentadas algumas atividades de lógica desenvolvidas cuja aplicação – em conjunto com o software LOGIKID – possa auxiliar no desenvolvimento de um pensamento computacional e lógico pelos alunos. Salienta-se que atividades propostas podem servir de base ou modelo para que os professores ou os próprios alunos desenvolvam outros exercícios envolvendo esse tipo de lógica ou até mesmo abordando outras áreas. O professor que avaliará a qualidade da solução da atividade realizada pelo aluno. As atividades são apresentadas em ordem crescente de complexidade. Nas situações, tanto o robô quanto a arena estão em escala. A posição origem  $x,y$  (0,0) da arena e a frente do robô são sinalizadas na Figura 15.

Figura 15: Sinalização da arena e do robô (fora de escala).



Fonte: Autor.

Figura 16: Atividade 1 - Exercitando o básico da lógica.

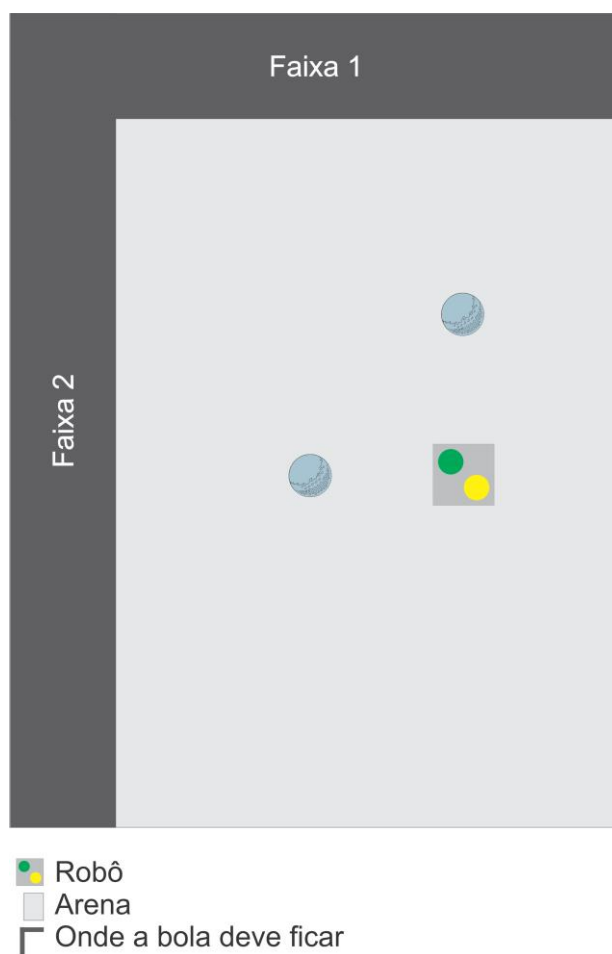


Fonte: Autor.

### Atividade 1 - Seguindo um trajeto

A atividade 1 visa familiarizar o aluno com a interface do LOGIKID e com os comandos básicos de programação. Nessa atividade, o aluno deverá fazer com que o robô dê uma volta na arena, fazendo o trajeto, como mostrado na Figura 16, seguindo o sentido das setas; deverá sair do ponto inicial fazendo caminho de início até o caminho de chegada. Diante disso, o aluno terá possibilidade de ter uma noção tanto dos algoritmos básicos do movimento como a noção da distância que o robô caminha assim como da velocidade. Também fornece noção de movimento angular de acordo o ângulo em graus ou radianos (área mais voltada à matemática). Uma possibilidade futura é o aluno escrever o código sem o auxílio dos botões do LOGIKID.

Figura 17: Atividade 2 – Deslocamento de bolas

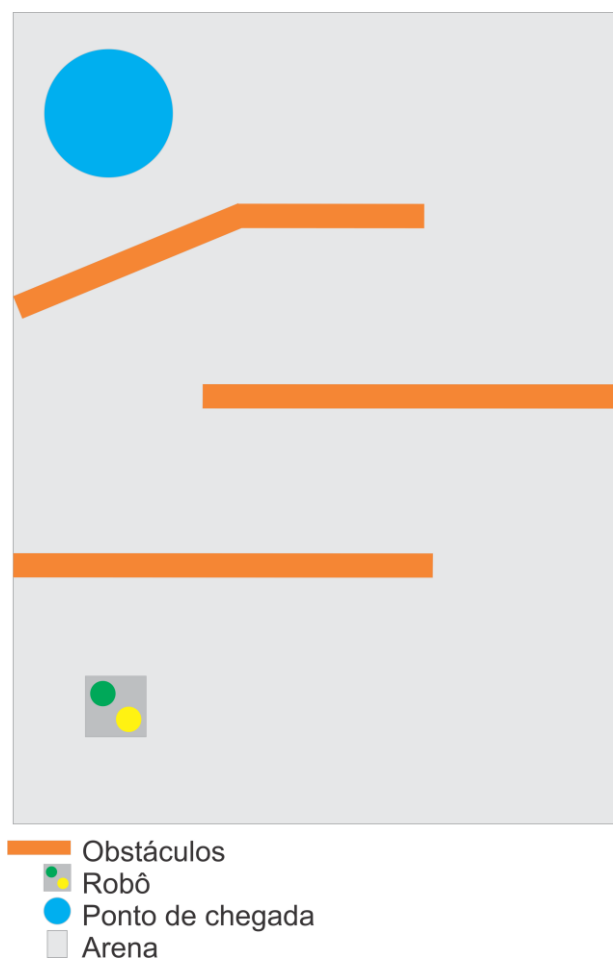


Fonte: Autor.

#### Atividade 2 – Deslocamento de bolas

Na atividade 2 (Figura 17) há inclusão de duas bolas que poderão ser de pingue-pongue, tênis, golfe, dentre outras. A ideia desta atividade é fazer com que os alunos elaborem e executem por meio do software os movimentos necessários para que o robô empurre as bolas (independente da ordem) até as faixas de modo com que cada bola fique em uma das faixas. A dificuldade consiste em deixar cada bola em sua respectiva faixa, visto que durante o movimento podem quicar ou rolar para fora do seu destino.

Figura 18: Atividade 3 – Desviar dos obstáculos.



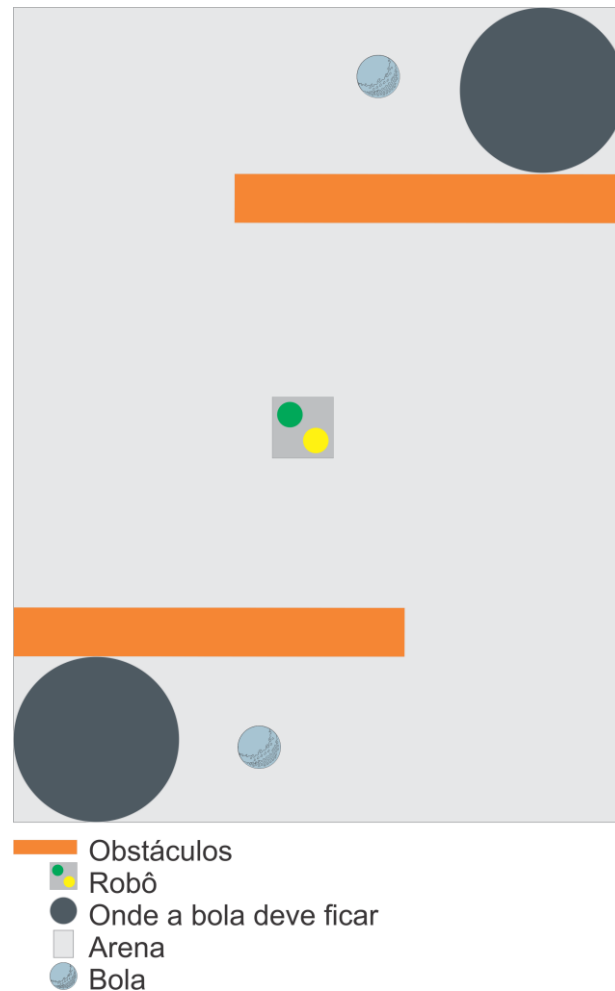
Fonte: Autor.

### Atividade 3 – Desviar dos obstáculos

Na atividade 3 (Figura 18) há inclusão de obstáculos que poderão ser qualquer tipo de material como madeira ou caixas. Ressaltando que as cores dos obstáculos não podem interferir na visão do AEDROMO, ou seja, não podem ter a mesma cor dos marcadores do robô. A ideia desses obstáculos é fazer com que os alunos elaborem mentalmente e depois, por meio do software, concretizem movimentos com o robô para se desviar desses materiais, com a posição inicial igual à figura ele deve chegar à bola azul. Essa atividade, além de reforçar os conceitos do exercício anterior, permite que o código seja avaliado de acordo com a precisão dos movimentos do robô como, por exemplo, a presença ou ausência de colisão com os obstáculos, assim como a realização

parcial ou integral desses movimentos com o robô pelo aluno. O tempo de execução também é uma variável que pode ser analisada.

Figura 19: Atividade 4 – Deslocamento de bolas com manobras.

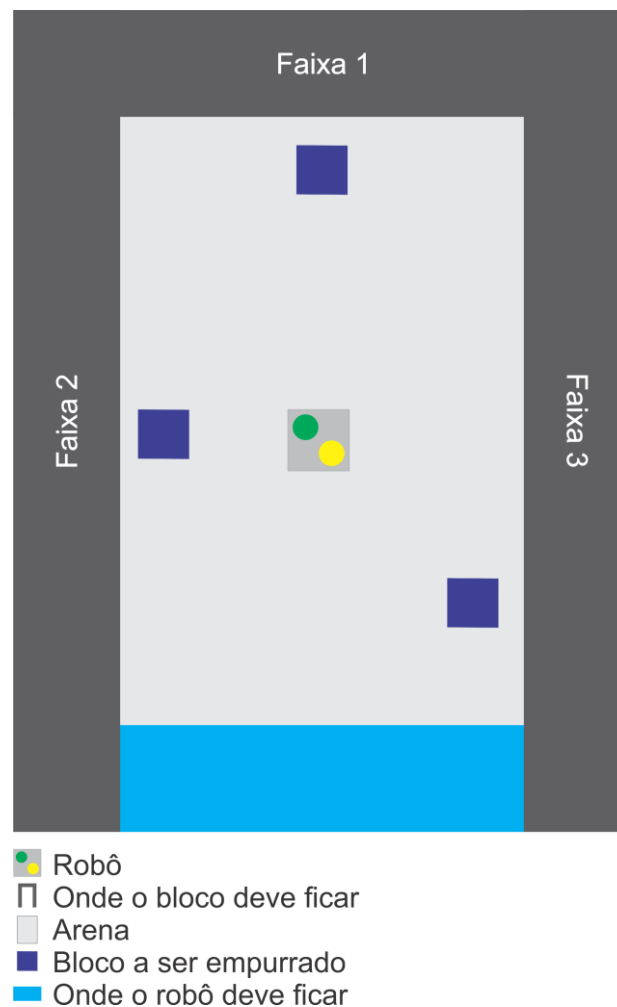


Fonte: Autor.

#### Atividade 4 – Deslocando de bolas com manobras

Na atividade 4 (Figura 19) o robô deve se locomover até a bola, deslocando-a até o local onde a mesma deve ficar (círculo cinza), lembrando que será necessário fazer isso duas vezes visto que existem duas bolas. Nesta atividade é possível perceber que a dificuldade do deslocamento do robô é maior e também que a área a qual ele deve levar a bola é menor em relação à atividade problema 2.

Figura 20: Atividade problema 5 – Sequência de passos.



Fonte: Autor.

### Atividade 5 – Movimento de blocos

Na atividade 5 (Figura 20) o aluno deve fazer com que o robô movimente cada bloco a uma área de chegada (faixas cinza numeradas) obedecendo determinada ordem e posteriormente se encaminhar para a área azul onde deve ficar. A dificuldade desta atividade consiste em se lidar com três objetos (blocos) que deverão ser colocados em ordem (faixa 1,2 e 3) de modo que cada um deles fique em sua respectiva faixa.



## **CAPÍTULO 5 – CONCLUSÕES E CONSIDERAÇÕES FINAIS**

O software criado, além de ser baseado em pressupostos de softwares utilizados e avaliados positivamente no contexto de ensino-aprendizagem como LOGO, SCRATCH e LIGHTBOT, conta também com o desenvolvimento de uma linguagem de programação integrada a robôs.

As características do LOGIKID foram baseadas numa estrutura existente que é o AEDROMO e a sua interface que gera o código de programação de acordo com a escolha do aluno, mediante uma “conversa” ou jogo de perguntas e respostas, cujo resultado será um programa em uma linguagem específica fazendo o robô agir de acordo com as escolhas do aluno.

O sistema LOGIKID não verifica se a tarefa foi cumprida corretamente, o que é tarefa do professor. Futuramente o sistema deverá identificar se alguma penalidade grave ocorreu, como por exemplo, choque o robô com algum obstáculo.

O software LOGIKID foi desenvolvido com intuito de possibilitar o desenvolvimento, nos alunos, de um pensamento computacional assim como da lógica na resolução de situações propostas, além competências. O aluno irá desenvolver esse raciocínio por meio de quatro operações básicas de movimento como frente, atrás, esquerda e direita.

O diferencial do LOGIKID é o uso de botões para escrever códigos de programação que controlam um robô. Destaca-se ainda que o software desenvolvido pode ser aplicado em instituições de ensino ou mesmo para uso particular não exigindo custos elevados.

O custo da montagem de um AEDROMO fica em torno de R\$200,00, pois seus componentes podem ser facilmente encontrados em lojas virtuais, porém esse custo pode ser reduzido visto que algumas partes do robô como os motores, os fios, as rodas e a parte em acrílico podem ser substituídas por sucatas.

Para trabalhos futuros com o LOGIKID, pretende-se: permitir a edição direta ou correção dos comandos gerados pela interface, desenvolver uma descrição formal da tarefa que permita um acompanhamento desta pelo sistema LOGIKID autonomamente e inserir novos comandos de interação dos robôs com o ambiente, pois atualmente o sistema é puramente deliberativo.

Figura 21: Apresentação do LOGIKID no evento Campus Party Brasil 2016.



Local: Anhembi (São Paulo-SP) Site do evento: <http://brasil.campus-party.org/>

## REFERÊNCIAS BIBLIOGRÁFICAS

ALENCAR, G.A.; FREITAS, A.K.; PESSOA, M.S.; MARTINS, D.J.S. Utilizando o SCRATCH nas aulas de Lógica de Programação do Proeja: um relato de experiência. **Nuevas Ideas en Informática Educativa** TISE 2014.

ALVES, S. F. R. ; FERASOLI FILHO, H. ; PEGORARO, R. ; CALDEIRA, M. A. C. ; YONEZAWA, W. M. ; ROSÁRIO, J. M. . Ambiente Educacional de Robótica Direcionado a Aplicações em Engenharia. In: **X Simpósio Brasileiro de Automação Inteligente**, 2011, São João del-Rei. Anais do X Simpósio Brasileiro de Automação Inteligente, 2011.

ANGONESE, A.T.; KREMPER, E.; ROSA, P.F.F. SIRLab: Uma Evolução do Projeto Engenharia-Escola para Competições Robóticas. In: V Workshop de Robótica Educacional. São Carlos. **Anais** – Editora Sociedade Brasileira de Computação, 2014. 17-22.

AVANZATO, R. Mobile robotics for freshman design, research, and high school outreach. In **2000 IEEE International Conference on Systems, Man, and Cybernetics**. 2000.

BARKER, B.S. ; ANSORGE, J. Robotics as means to increase achievement scores in an informal learning environment. **Journal of Research on Technology in Education**, 39(3), 2007. p.229- 243.

BARROS, R.P.; TORRES, V.P.; BURLAMAQUI, A.M.F. TardBot: Tecnologias assistivas para imersão de deficientes visuais na robótica educacional. In: V Workshop de Robótica Educacional, 2014 – São Carlos, **Anais**, Editora Sociedade Brasileira de Computação, 11-16.

BIGGS, J. **Light-Bot Teaches Computer Science With A Cute Little Robot And Some Symbol-Based Programming**. Posted Jun 26, 2013. Retirado de <<http://techcrunch.com/2013/06/26/light-bot-teaches-computer-science-with-a-cute-little-robot-and-some-symbol-based-programming/>>

D'ABREU, J.V.V., Robótica Pedagógica: percurso e perspectivas. In: V Workshop de Robótica Educacional, 2014 – São Carlos, **Anais**, Editora Sociedade Brasileira de Computação, 79-83.

DENNING, P. J. "Great principles in computing curricula". In: **XXXV ACM Technical Symposium on Computer Science Education** – SIGCSE, p.336-341, Norfolk, Virginia, USA, 2004.

DULLIUS, S.R. **O software SCRATCH como uma ferramenta de apoio ao processo de aprendizagem**. Universidade Federal do Rio Grande do Sul, UFRGS, Brasil. 2008

FERASOLI FILHO, H., PEGORARO, R., CALDEIRA, M.A.C., ROSÁRIO, J., **AEDROMO - An Experimental and Didactic Environment with Mobile Robots**. In: **Proceedings of The 3rd International Conference on Autonomous Robots and Agents**, 2006.

FERRUZZI, E.C. Considerações sobre a linguagem de programação LOGO. **Seminário Apresentado no GEIAAM** – Grupo de Estudos de Inteligência Artificial Aplicada à Matemática- UFSC - Setembro/2001.

FRANÇA, R. S.; AMARAL, H. J. C. “Proposta Metodológica de Ensino e Avaliação para o Desenvolvimento do Pensamento Computacional com o Uso do SCRATCH”. In: **Workshop de Informática na Escola – WIE**. p.179-188, Campinas – SP. 2013.

FRANÇA, R.; SILVA, W.; AMARAL, H. “Ensino de Ciência da Computação na Educação Básica: Experiências, Desafios e Possibilidades”. In: **XX Workshop de Educação em Computação - WEI**. Curitiba, PR. 2012.

FRIEDRICH, R.V.; SANTOS, D.S.; KELLER, R.D. Proposta Metodológica para a Inserção ao Ensino de Lógica de Programação. Anais do 23º **Simpósio Brasileiro de Informática na Educação** (SBIE 2012), ISSN 2316-6533 Rio de Janeiro, 26-30 de Novembro de 2012.

GOMES, Cristiane Grava. ; SILVA, F. O. ; BOTELHO, J. C. ; SOUZA, A. R. . A robótica como facilitadora do processo ensino aprendizagem de Matemática no Ensino Fundamental. In: Nelson Antonio Pirola. (Org.). **Ensino de ciências e matemática IV: temas de investigação**. 1ed.SÃO PAULO: CULTURA ACADÊMICA EDITORA, 2010, v. IV, p 205-221.

GUEDES, A. L. ; KERBER, F. M. . Usando a robótica como meio educativo. **Revista Unoesc & Ciência** - Área das Ciências Exatas e da Terra, v. 1, p. 199-207, 2010

INDY TELECOM & INDUSTRIAL MEDIA. **LEGO MINDSTORMS**. 2006.

Disponível

em: <<http://www.metlin.org/content/lego/>>

JESUS, A.M.D.; FERREIRA, L.A.C.; SANTOS, F.M.C.; RUFO, M.C.B.; AMBRÓSIO, G.D. ArduAlg: Ambiente de Programação Fácil para Robótica na Plataforma Arduino. In: V Workshop de Robótica Educacional, 2014 – São Carlos, **Anais**, Editora Sociedade Brasileira de Computação, 29-34.

KALTPOST. **Scratch-sherpa**. 2012. Retirado de <  
[http://gpio.kaltpost.de/?page\\_id=1558](http://gpio.kaltpost.de/?page_id=1558)>

KNUDSEN, J. **The unofficial guide to Lego Mindstorms robots**. Sebastopol, CA: O'Reilly. 1999.

MANDAI, L.M. **Manual de Instruções da Linguagem de Blocos LEGO**. Laboratório de Sistemas Embarcados e Críticos (LaSEC). Centro Paula Souza. FATEC, Lins. 2014.

MANUAL SCRATCH – **Computação criativa: uma introdução ao pensamento computacional baseada no conceito de design**: tradução por EduScratch, 2011.

MÉLO, F.E.N.; CUNHA, R.R.M.; SCOLARO, D.R.; CAMPOS, J.L. Do Scratch ao Arduino: uma proposta para o ensino introdutório de programação para cursos superiores de tecnologia. **XXXIX Congresso Brasileiro de Educação em Engenharia – COBENGE 2011**. Blumenau, SC. 2011.

MICHELON, N. **O uso do software educacional como suporte de produção e autoria no ensino fundamental**. Porto Alegre: Universidade Federal do Rio Grande do Sul – Centro Interdisciplinar de Novas Tecnologias na Educação, 2012. Trabalho de Conclusão de Curso, Curso de Especialização em Mídias na Educação, Porto Alegre: 2012.

NASCIMENTO, M.C.D.; FONSECA, C.A.G.; GONÇALVES, L.M.G. Experiência de Robótica com Materiais Alternativos na Escola Estadual Professor Luís Soares. In: V Workshop de Robótica Educacional, 2014 – São Carlos, **Anais**, Editora Sociedade Brasileira de Computação, 23-28.

NETO, V.S.M. A utilização da ferramenta SCRATCH como auxílio na aprendizagem de lógica de programação. **II Congresso Brasileiro de Informática na Educação (CBIE 2013)**.

NETTO, L.F. Tecnologia Educacional entre universitários: Análise de um levantamento de perfil perante a Educação na Sociedade da Informação. **2º Contecsi – Congresso Internacional de Gestão da Tecnologia e Sistemas de Informação/ International Conference on Information Systems and Technology Management**. Junho-2005. São Paulo.

PAPERT, S., **Mindstorms: children, computers, and powerful ideas**, Basic Books, Inc: New York, NY, USA. 1980.

PEREIRA JÚNIOR, J.; RAPKIEWICZ, C. E.; DELGADO, C.; XEXEO, J. A. M. “Ensino de Algoritmos e Programação: Uma Experiência no Nível Médio”. **XIII Workshop de Educação em Computação - WEI**. São Leopoldo, RS. 2005. PRETTO, Nelson de Luca. **Escritos sobre educação, comunicação e cultura**. Papirus: Campinas, SP. 2008.

PNCA, 2016. <http://pnca.dominiotemporario.com/>

PRICE, B. A.; RICHARDS, M.; PETRE, M.; HIRST, A.; JOHNSON, J. Developing robotics e-teaching for teamwork. **International Journal of Continuous Engineering Education and Lifelong Learning**, 13(1/2), 190-205. 2003.

PROJETO LOGO. **Por que Logo?** Atualizado em 2 Ago. 2009. Disponível em: <<http://projetologo.webs.com/texto2.html>>

RODRIGUES, S.R. **Noções básicas de programação: LOGO**. Nota didática. Instituto de Ciências Matemáticas de São Carlos, Universidade de São Paulo ILTC. São Carlos, Maio de 1993.



SANT'ANNA, H. C.; NEVES, V. B. "SCRATCH Day UFES: oficina itinerante de introdução a programação para professores". In: **IV Simpósio Hipertexto e Tecnologia na Educação**, Recife – PE. 2012

SCRATCH. ABOUT SCRATCH (SCRATCH Documentation Site). Disponível em: <[http://info.scratch.mit.edu/About\\_Scratch](http://info.scratch.mit.edu/About_Scratch)>. Acesso em 12/08/2015.

SANTOS, F.K. JAM - **Um Jogo de Aprendizagem Multidisciplinar**. UNIOESTE. Universidade Estadual do Oeste do Paraná. Trabalho de Conclusão de Curso. Curso de Bacharelado em informática. 2010.

SBC – Sociedade Brasileira de Computação. "A Computação como educação básica". In: **Revista da Sociedade Brasileira de Computação**. Porto Alegre: Giornale Comunicação Empresarial. 2011.

SHIMABUKURO, M.; PEGORARO, R. Simulador do Ambiente Educacional Didático de Robôs Móveis – AEDROMO. In: V Workshop de Robótica Educacional, 2014 – São Carlos, **Anais**, Editora Sociedade Brasileira de Computação, 73-78.

SILVA, T. R. ; ARAUJO, G. G. ; ARANHA, E. H. S. . **Oficinas Itinerantes de SCRATCH e Computação Desplugada para Professores como apoio ao Ensino de Computação**: um Relato de Experiência. In: 20º Workshop de Informática na Escola - WIE, 2014, Dourados - MS. Anais do III Congresso Brasileiro de Informática na Educação - CBIE, 2014. p. 380-389

SOUSA, A.M. **Área de trabalho dinâmica para o ambiente experimental e didático com robôs móveis**. Trabalho de Conclusão de Curso. Ciência da Computação. Faculdade de Ciências. Universidade Estadual Paulista Júlio de Mesquita Filho. 50f, 2015.

SOUZA, L.F.F.; PEREIRA, E.B.P. **A Robótica Educacional Como Meio de Integração Entre o Ensino Fundamental e de Graduação Pelo Uso da Linguagem LOGO**. In: Workshop de Robótica Educacional, 2012 – UFRN.

V Workshop de robótica educacional 20 e 21 outubro de 2014. São Carlos. **Anais**. Instituto de Ciências Matemáticas e de Computação – ICMC, Universidade de São Paulo – USP: Editora Sociedade Brasileira de Computação – SBC, 2014. 83p.

LEGO. <http://mindstorms.lego.com> <Acesso em 10/07/2015>.

ADORO ROBÓTICA. <http://www.adororobotica.com/programacaobasica.pdf> <Acesso em 10/07/2015>.