

# Arena: Uma ferramenta auxiliar na disciplina “Introdução à Computação” para programação de robôs

Rômulo Nunes de Oliveira<sup>1</sup>

<sup>1</sup>Universidade Federal de Alagoas (UFAL) – Campus Arapiraca  
Rod. AL 115, Km 6,5 – 57.309-005 – Bom Sucesso – Arapiraca – AL – Brasil

romulonunes@gmail.com

**Abstract.** *Students which started an undergraduation course in areas related to Mathematics, Computer Science, and other, have difficulties in the issues which requires knowledge learning in mathematical abstraction and logical reasoning. In addition, in courses not directly related to the Computer Science, the motivation lack exists as in the issues of “Introduction to the Computation” as in the issues that requires machine programming skills. For these reason, the Arena software was built as an aid-didacticism tool aiming at stimulating the study through a game that uses a simple language in structured Portuguese. In this article are presented software characteristics, use tests, and results.*

**Resumo.** *Estudantes que iniciam um curso de graduação em ciências exatas, normalmente encontram dificuldades nas disciplinas que exigem abstração matemática e raciocínio lógico. Associado a isso, para os cursos que não estão relacionados diretamente à Ciência da Computação, existe a falta de motivação para as disciplinas de “Introdução à Computação” ou onde as ementas sugerem tópicos de programação de máquinas. O software Arena foi construído como ferramenta auxílio-didática, com o objetivo de estimular o estudo através de um jogo que usa uma linguagem simples em português estruturado. As características do software, testes de utilização e resultados são apresentados neste artigo.*

## 1. Introdução

A disciplina “Introdução à Computação”, ou equivalente, está presente na grande maioria dos cursos de graduação na área das Ciências Exatas. Isso porque além de uma ferramenta da informática no currículo, o graduando também necessita desenvolver o seu raciocínio lógico para resolução de problemas.

Apesar de sua importância, a disciplina enfrenta o seguinte problema: Os estudantes que iniciam um curso de Graduação nas exatas, normalmente encontram dificuldades nas disciplinas que exigem abstração matemática e raciocínio lógico. Associado a isso, para os cursos que não estão relacionados diretamente à Ciência da Computação, existe a falta de motivação. Afinal, é difícil encontrar alguém recém formado no 2º Grau com maturidade suficiente para perceber que por trás da ementa de uma disciplina existe o interesse de uma formação intelectual no raciocínio lógico para solução de problemas. Além desses fatores, segundo as diretrizes gerais do REUNI [REUNI 2007] (para as universidades federais que aderiram ao programa), a boa qualidade do ensino deve ser mantida, associada ao aumento do número de formandos, diminuição do fator de retenção

etc. Surge então a necessidade de criar modelos, métodos e ferramentas para combater a situação.

Diante desse problema, propomos um software auxiliar para estímulo ao estudo da disciplina através de um jogo. O aluno poderá instruir um robô para uma batalha com uma linguagem em português estruturado, proporcionando nesse momento inicial da disciplina uma resposta (*feedback*) mais atraente que uma tela preta com o texto clássico “Oi mundo!”. Um dos requisitos do software era o de ser de fácil utilização por quem nunca viu programação de máquinas, o que limitou o aprofundamento nos conteúdos da disciplina.

Os conceitos da disciplina que estão presentes no jogo são:

- Algoritmo sequencial (estruturado);
- Linguagem interpretada;
- Respeito à sintaxe de uma linguagem;
- Estrutura de decisão simples com “se” e “senão”;
- Composição com mais de uma estrutura de decisão;
- Raciocínio lógico e estratégia.

O tema “batalha de robôs” foi escolhido por acreditar-se na sua dinamicidade, atratividade e por propiciar uma gama de situações onde o aluno deve atuar “dizendo” o que fazer. Em tempo, é importante frisar que o tema abordado não faz nenhuma apologia ao uso de armas ou guerras, uma vez que alunos de graduação já possuem idade apropriada discernir sobre o assunto.

## **2. Trabalhos relacionados**

O Robocode da IBM é um ambiente desenvolvido em Java sobre o mesmo jogo temático “Batalha entre robôs” com o objetivo de facilitar o aprendizado da própria linguagem Java. No Robocode o aluno também aprende sobre o paradigma Orientação a Objetos e pode incluir ferramentas da Inteligência Artificial em seus robôs. Tudo isso porque a programação de cada robô se dá na própria linguagem Java, o que inclui o uso de heranças, métodos, e também suporta o armazenamento de informações, associação com outras ferramentas desenvolvidas em java. O Robocode tem se mostrado ser uma ferramenta didática importante, porém apenas para usuários que no mínimo possuem uma noção básica de programação e orientação a objetos. Além disso, o uso da linguagem Java normalmente se dá apenas nos cursos relacionados a Ciência da Computação, Sistemas de Informação e afins.

Um outro ambiente de programação tem se mostrado uma importante ferramenta didática durante anos. O LOGO foi criado na década de 60 por Seymour Papert no Instituto de Tecnologia de Massachussets, e desde então tem sido usado como ferramenta para o desenvolvimento da lógica matemática e no estudo da abstração para resolução de problemas [Martins and Correia 2003]. Algumas instituições de ensino fundamental e médio utilizam o LOGO em aulas de informática, devido sua fácil integração com as disciplinas de matemática e ciências, principalmente em questões espaciais. A tartaruga interativa do LOGO é um dos motivos para o sucesso da linguagem, onde o aluno aprende “ensinando a tartaruga” a se locomover.

Apesar de usar uma linguagem simples, do auxílio visual proporcionado no ambiente, o LOGO não cativa o público adulto, como observado no estudo de caso por

Scheila Martins [Martins and Correia 2003]. Os trabalhos de Daniele Martins e Ken Kahn [Martins and Júnior 2007, Kahn 1996] discutem a utilização de games no processo ensino-aprendizagem por fazerem parte do gosto consensual entre os jovens. O trabalho de Vânia Martins e outros [Martins et al. 2007] discute a utilização dos games em EAD na tentativa de diminuir a evasão nos cursos à distância.

O Arena surgiu da necessidade de um ambiente para auxílio no ensino dos conceitos da programação básica estruturada, que o Robocode não contempla. Entretanto, o tema “Batalha de robôs” abordado pelo Robocode foi inspiração para o ambiente Arena.

Quanto à linguagem utilizada no software Arena pelos alunos, certamente tem a característica de ser de rápida assimilação. Isso porque a disciplina “Introdução à Computação” possui uma carga horária curta e que o Arena é um software “introdutor” aos conceitos da programação. Uma linguagem estruturada tradicional, como C ou Pascal, continua fazendo parte do estudo aprofundado.

Assim, no Arena encontramos a mesma característica de simplicidade de linguagem utilizada no LOGO. Entretanto, enquanto o LOGO é uma ferramenta rica e adaptável à diversos contextos, o ARENA é um game, com um objetivo fixo, onde a aprendizagem do aluno se dá na constante otimização (Aprendizagem por reforço) das instruções fornecidas aos robôs.

Outras linguagens e ambientes como nos trabalhos de Roland Untch [Untch 1990], Eike Anderson [Anderson and McLoughlin 2006], Ken Kahn [Kahn 2004], Mauro Mattos [Mattos et al. 1999] e Patrícia Petry [Petry 2005], também surgiram ao longo do tempo, todas com o objetivo de facilitar o entendimento da abstração exigida para programação de computadores. Entretanto, não possuem o conjunto de características desejadas para ferramenta introdutória em algoritmos, a saber, simplicidade nas instruções, tema atrativo e atual, uso da metodologia aprendizagem por reforço.

### 3. Descrição do ambiente

No software Arena os robôs já existem, bastando apenas dar-lhes um nome para que sejam carregados no jogo. Porém, apesar de existirem e estarem carregados no jogo, não fazem nada sem instruções. O Arena fornece robôs “vazios”, mas capazes de executar instruções. Os alunos deverão instruí-los para um batalhas em arena. A Figura 1 mostra a tela principal do software.

Formalmente, a BNF (*Backus-Naur Form*) que descreve a linguagem utilizada pelos alunos para programar o robô está descrita abaixo.

```
<Bloco Condicional> ::= <Bloco Condicional>
                        <linha branco>
                        <Bloco Condicional>
                        | se <condição> <comando>
                        [ senao <comando> ]
                        fim <FDL>

<Linha Branco>       ::= <FDL> <FDL>

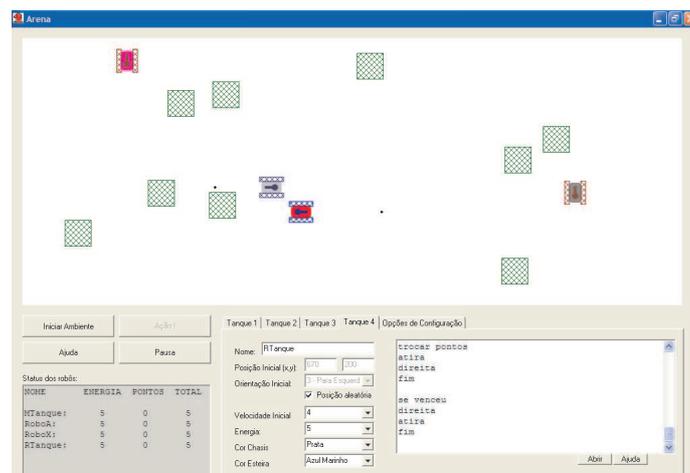
<FDL>                 ::= ? US-ASCII character 13 ?
```

```

<Condição> ::= atingido <FDL>
              | mirando <FDL>
              | tiro aproximando <FDL>
              | colisao com parede <FDL>
              | colisao com alvo <FDL>
              | acertou <FDL>
              | anergia zero <FDL>
              | venceu <FDL>
              | evento aleatorio <FDL>
              | evento raro <FDL>
              | evento rarissimo <FDL>

<Comando> ::= <Comando> CRLF <Comando>
              | <Bloco Condicional> <FDL>
              | atira <FDL>
              | esquerda <FDL>
              | direita <FDL>
              | ladotiro <FDL>
              | viraaleatorio <FDL>
              | pare <FDL>
              | trocarpontos <FDL>
              | +velocidade <FDL>
              | -velocidade <FDL>

```



**Figura 1. Tela de interface principal do software Arena.**

Em outras palavras, para que possam instruir seus robôs, os alunos devem conhecer a sintaxe e semântica dos comandos (Tabela 1), além de conhecer os eventos pré-estabelecidos monitorados pelo robô. O jogo possui um manual de usuário com a lista completa e detalhada de todos eles.

Os eventos pré-estabelecidos são booleanos (Tabela 2) e usados com o comando condicional “se”, opcionalmente incluindo o “senao”. A instrução “fim” delimita cada bloco de comandos. A sintaxe do comando “se” está representada abaixo.

```

se <condição pré-definida>
  <conjunto de instruções caso condição pré-definida verdadeira>
[ senao
  <conjunto de instruções caso condição pré-definida falsa> ]
fim

```

Comando	Semântica da instrução
direita	Faz o robô mudar a sua direção. Vira para direita.
viraaleatorio	Faz virar aleatoriamente para esquerda ou direita.
atira	Faz o robô disparar um tiro com seu canhão
pare	Faz o robô parar (velocidade 0)
+velocidade	Faz o robô aumentar sua velocidade em 1 unidade/seg, sendo que a velocidade máxima é de 6 unidades/seg.

**Tabela 1. Alguns comandos conhecidos pelo robô**

Evento booleano	Semântica do Evento
mirando	Informa se existe alvo na mira do canhão do robô.
colisao com parede	Informa se houve colisão com uma das paredes da arena.
colisao com alvo	Informa se houve colisão com alvo fixo ou móvel.
energia zero	Informa se o robô está sem energia.
atingido	Informa se o robô foi atingido e perdeu energia.

**Tabela 2. Alguns eventos conhecidos pelo robô**

Segue um exemplo com dois blocos de instruções e eventos pré-definidos:

```
se mirando
  pare
  atira
fim
```

```
se colisao com parede
  direita
  +velocidade
fim
```

#### 4. Testes de utilização prática

A utilização prática em sala de aula se deu no início do período letivo de 2008.1, com duas turmas da disciplina “Introdução à Computação”. A turma *A* é composta por 33 alunos do fluxo padrão, que nunca cursaram a disciplina. Já a turma *B*, composta por 13 alunos, foi formada por aqueles que já reprovaram na disciplina.

Em ambas as turmas foram realizadas demonstrações do software no primeiro dia de aula e um agendamento de uma atividade para aula seguinte. A atividade foi individual e consistia na elaboração de instruções para um robô no Arena. Para motivação de todos, no segundo dia de aula os robôs participariam de um torneio, cujas premiações eram pontos na disciplina. No torneio houve uma participação média de 95% em ambas as turmas.

As regras da competição foram:

- Tempo cronometrado de combate em 3 minutos;
- Em cada jogo do torneio competiam dois alunos. Cada aluno competia com duas cópias do seu robô que não se destruíam. Eram portanto 4 robôs por competição;
- Os valores adotados para energia e velocidade inicial de cada robô eram os valores padrões do jogo;

- Os robôs iniciavam em posições aleatórias, assim como os alvos fixos;
- A primeira fase do torneio era por *chaves independentes*, onde buscava-se a melhor pontuação e conseqüente classificação para segunda fase. A segunda fase reunia os dois melhores de cada chave individual, no total de 8 alunos, e era de caráter eliminatório.
- A pontuação no placar de cada jogador era o somatório dos pontos totais dos seus robôs na partida, afora os robôs totalmente destruídos.

Os campeões de cada turma, robôs com algoritmos diferentes, conseguiram vencer invictos todos os jogos do torneio. Na turma *A* o vencedor competiu em 9 jogos, e na turma *B* o vencedor competiu 5 vezes. Essas informações são fortes evidências de que os algoritmos desenvolvidos pelos alunos realmente influenciam os resultados dos jogos.

Um resumo do impacto no aprendizado pode ser observado na Tabela 3, que mostra o comparativo entre as turmas *A* e *B* que utilizaram o software em 2008 e a turma *X* que não utilizou o software em 2007. Vale ressaltar que os alunos da turma *B* foram os reprovados na turma *X*. Um outro dado obtido foi o índice de aprovação nas turmas. Enquanto a turma *X* teve 28,6% de aprovados, a turma *A* teve um índice de 58,4%. A turma *B*, de repententes, teve um índice de 84,6% de aprovação.

Questionamentos	Turma X	Turma A	Turma B
Exemplo de algoritmo em português	94%	100%	100%
O que é linguagem interpretada	90%	95%	100%
O que é um erro de sintaxe	73%	74%	95%
Questão envolvendo estrutura de decisão simples	62%	77%	90%
Questão envolvendo estrutura de decisão composta	25%	41%	81%

**Tabela 3. Quadro dos índices de acerto das turmas.**

Acreditamos que a maior dificuldade nas disciplinas que envolvem o raciocínio lógico é vencer os tabus, os convencionalismos formados em relação ao cálculo e raciocínio lógico. Muitos já enxergam a dificuldade antes mesmo de cursar/estudar a disciplina. Nas nossas turmas de “Introdução a Computação” não acontece de forma diferente. Entretanto, o software apresentado no início da disciplina ajudou a ilustrar o oposto dessa imagem negativa.

A participação de 95% dos alunos nas competições no início do semestre letivo e o resgate, pelos próprios alunos, de exemplos em aulas com as características do software mostram que houve um índice satisfatório de aceitação ferramenta.

## 5. Considerações Finais

Apesar do melhoramento no índice de aprovação, cabe frisar que não consideramos aqui outros fatores externos que também podem ter contribuído com isso. O estudo de variáveis como a média de QI (Quociente de inteligência) das turmas, o índice de concorrência nos exames de vestibular e nível de dificuldade dos testes não fizeram parte desta avaliação. Dessa forma, temos apenas fortes indícios de um influência positiva através dos números apresentados e pela estratégia didática incentivadora.

Um trabalho futuro promissor poderia resolver a questão de como incluir outros conteúdos da disciplina sem prejudicar a simplicidade de utilização, ou seja, como incluir

funcionalidades sem prejudicar a compreensão indutiva do usuário. Outros melhoramentos de programação também devem ser alvo de estudos, como por exemplo a questão da disponibilidade em plataformas diferentes, poder ser utilizado em rede (Internet e Intranet) etc.

Atualmente o Curso de Ciência da Computação desta IES planeja utilizar a ferramenta como meio de divulgação do curso em escolas da região, também com o objetivo de desmistificar os preconceitos relacionados as ciências exatas.

## Referências

- Anderson, E. F. and McLoughlin, L. (2006). Do robots dream of virtual sheep: Rediscovering the “karel the robot” paradigm for the “plug and play generation”. In *In Proceedings of the Fourth International Game Design and Technology Workshop and Conference (GDTW2006)*, Liverpool, UK.
- Kahn, K. (1996). Toontalk - an animated programming environment for children. *Journal of Visual Languages and Computing*, 7(2).
- Kahn, K. (2004). *A Learning Zone of One's Own: Sharing Representations and Flow in Collaborative Learning Environments*, chapter ToonTalk: Steps Towards Ideal Computer-Based Learning Environments. Ios Pr Inc.
- Martins, D. M. and Júnior, D. R. C. (2007). Jovens jogadores de videogames e produção de sentidos: Contribuições para se pensar em práticas educativas alteritárias. In *Anais da 30 Reunião Anual da ANPEd*, Minas Gerais.
- Martins, S. W. and Correia, L. H. A. (2003). O logo como ferramenta auxiliar no desenvolvimento do raciocínio lógico: Um estudo de caso. In *Proceedings of International Conference on Engineering and Computer Education - ICECE*.
- Martins, V., Hauguenauer, C., Clua, E., and Cunha, G. (2007). Design de games para utilização em ead. *Colabora*, 4(1):1 – 12.
- Mattos, M. M., Fernandes, A., and López, O. C. (1999). Sistema especialista para apoio ao aprendizado de lógica de programação. In *VII Congresso Iberoamericano de Educación Superior en Computación (CIESC99)*, Asunción, Paraguay.
- Petry, P. G. (2005). Um sistema para o ensino e aprendizagem de algoritmos utilizando um companheiro de aprendizagem colaborativo. (dissertação de mestrado), Universidade Federal de Santa Catarina, Florianópolis.
- REUNI (2007). *Programa de Reestruturação e Expansão das Universidades Federais*. Ministério da Educação (MEC) - Governo Federal, On-line <http://portal.mec.gov.br>.
- Untch, R. H. (1990). Teaching programming using the karel the robot paradigm realized with a conventional language. On-line <http://www.mtsu.edu/untch/karel/karel90.pdf>.