

José Weliff Ferreira Lima
Alandson Mendonça R. Meireles

RESUMO

Na área da ciência da computação um raciocínio lógico e uma rápida análise de dados são pré-requisitos obrigatórios para uma tomada de decisão eficaz. A primeira disciplina que envolve o raciocínio lógico, a Lógica de Programação, é uma das disciplinas com maiores índices de reprovação nos cursos de Tecnologia de Informação. Aprender a programar é um processo difícil e exigente, diante de tal afirmação foi feito um estudo sobre diversas metodologias / ferramentas de ensino que abordam a aprendizagem de lógica de programação. Cada uma dessas metodologias foram descritas e as vantagens e desvantagens de sua utilização foram enumeradas, demonstrando um exemplo prático. Ao final, concluímos que para um melhor processo de ensino de lógica de programação sugere-se a combinação dessas metodologias para absorver o melhor conteúdo de cada.

Palavras-chave: lógica de programação. Ensino. Estudo. Metodologias.

ABSTRACT

In the area of Computer Science a logical reasoning and a quick analysis of data are required for an effective decision making. The first discipline that involves logical reasoning, logic programming, is one of the subjects with higher failure rates in Information Technology courses. Learning to program is a difficult and demanding process, facing this statement a study on various methodologies / tools that discuss teaching learning logic programming was done. Each of these methods were described and the advantages and disadvantages of their use were listed, demonstrating a practical example. At the end, we concluded that a better process for teaching programming logic suggests the combination of these methodologies to the best absorb the contents of each.

Keywords: Logic programming. Teaching. Study. Methodologies

1 INTRODUÇÃO

Na ciência da computação e desenvolvimento de *softwares* demandam que os desenvolvedores e analistas de sistemas tenham um raciocínio rápido. Para ser um desenvolvedor é importante ter um bom raciocínio lógico e uma rápida análise de dados para ser capaz de tomar decisões com mais eficácia. Essa essência é obtida durante os estudos e formação dos profissionais na academia. Entretanto, muitos não conseguem ter uma boa abstração, pois é necessária uma base ou dedicação dos alunos nos estudos, assim, pessoas que querem seguir na área de programação e criação de *softwares* acabam não alcançando o sucesso esperado no mercado de trabalho. Para Rodrigues Júnior (2004) o fracasso está ligado ao preparo dos alunos para o ingresso na faculdade principalmente nas particulares onde é mais fácil o processo seletivo, assim entrando mais alunos sem experiência.

Nos cursos que abordam a programação de computadores, a disciplina lógica de programação é uma das principais, onde se inicia a treinar a mente, seu foco não é aprender apenas uma linguagem de programação, mas sim a lógica em geral usada em qualquer linguagem. Para muitos, aprender lógica de programação é um desafio, além de geralmente a disciplina em questão nunca ter sido cursada anteriormente, muitos desconhecem as ferramentas de apoio à aprendizagem.

Pode-se observar que muitos alunos já no final de algum curso da área da computação pouco sabem trabalhar com a lógica, o que está relacionado ao impacto do primeiro contato com a disciplina e o desconhecimento das ferramentas. Para Gomes e Mendes (2000), “Aprender a programar é um processo difícil e exigente. Para se tornarem bons programadores os alunos têm de adquirir um conjunto de conhecimentos que vão além de conhecer a sintaxe e a semântica de uma linguagem de programação”.

Existem diferentes metodologias/ferramentas de ensino que abordam a aprendizagem de lógica de programação de uma forma simples sem a necessidade de ter logo no início o formalismo de uma linguagem de programação que tem suas expressões em inglês e pode comprometer o aluno a usar uma IDE (*IntegratedDevelopmentEnvironment*), um ambiente integrado para desenvolvimento de software, podendo dificultar o aluno entender a lógica e causar o desânimo. Dessas metodologias, além das linguagens de programação, podemos citar o português, fluxogramas e ensino através de jogos lógicos.

Pseudocódigo ou português (português estruturado) é apenas uma sintaxe, em geral usando o português, adaptada para criação de pequenos algoritmos usando a lógica. Fluxogramas são passos descritos em formas

de figuras e em uma sequência para o desenvolvimento de um processo ou resolução de um problema. Jogos educacionais também vêm ganhando espaço no ensino de lógica de programação. De acordo com Rapkiewicz (2006) os jogos têm poder de motivação, pois divertem o aluno enquanto aprende, além de ter uma retenção maior do que é ensinado.

Esse artigo irá mostrar a importância dessas metodologias mostrando suas vantagens e desvantagens e no final sugerir um ponto de equilíbrio entre todas elas para que chegue ao resultado satisfatório durante o processo de aprendizagem de lógica de programação.

2 SÍNTESE DE INFORMAÇÕES

Aprender lógica de programação é um processo difícil, Gomes e Mendes (2000) afirmam que aprender lógica de programação é um processo que exige muito, tanto dos alunos como dos professores ou instrutores. Uma vez que temos insucesso em uma disciplina de lógica, iremos ter dificuldades em todo processo de um curso de computação nas próximas disciplinas de programação.

Uma vez que a aprendizagem de programação ocorre praticamente, durante todo o curso, o baixo índice de assimilação dos estudantes nas disciplinas cujos requisitos exigem o conhecimento de programação tem sido um grande problema enfrentado em muitas instituições (SANTIAGO e DAZZI, 2004).

Hoje nos deparamos com uma grande evasão de alunos na área de Tecnologia da Informação, causando um déficit de profissionais de TI no mercado de trabalho. Rocha (1991) afirma que estamos tendo um fracasso no ensino de programação. “O bom aprendizado de lógica de programação é essencial para a formação do profissional de informática que busca o desenvolvimento de sistemas de qualidade” (ALMEIDA, 2002).

Alunos que não dominam lógica de programação costumam ter grandes dificuldades em cadeiras de linguagem de programação como Estrutura de Dados, Programação Orientada a Objetos, Paradigmas de Linguagens de Programação. Quando não são estimulados a estudarem além da sala de aula, e os professores não procuram motivar os alunos o processo de aprendizagem torna-se difícil. “Isto torna difícil e desagradável o processo de aprendizagem que deveria ser na medida do possível fácil e prazeroso” (MEDEIROS e DAZZI 2002).

2.1 FERRAMENTAS

As ferramentas são recursos que podem ajudar tanto o professor a passar o conteúdo e métodos passados em sala como também estimular os alunos. “A utilização de uma ferramenta computacional para os alunos

confeccionarem seus algoritmos, permitindo aos mesmos testarem suas soluções visualizando o resultado gerado por elas, é antes de tudo, um grande motivador do processo de ensino aprendizagem” (DAZZI, MIRANDA, SOUZA 2000; SANTIAGO e DAZZI 2003). Gomes e Mendes (2000) informam que as ferramentas permitem a construção de resoluções para problemas, bem como a sua simulação, observação e análise num maior nível de facilidade. Caso a solução não seja adequada, o aluno poderá detectar e corrigir os erros cometidos, de forma a encontrar uma solução satisfatória. Este processo de detecção e correção de erros é fundamental para o desenvolvimento nos aprendizes de programação.

Nesta seção serão mostradas três ferramentas mais comuns de ensino de lógica de programação: Fluxogramas, Pseudocódigos e Jogos educacionais. Cada uma mostrando suas vantagens e desvantagens enfatizadas também por outros autores e um exemplo de resolução para um problema simples com cada ferramenta, assim tornando mais visível à diferença e aproveitamento de cada uma.

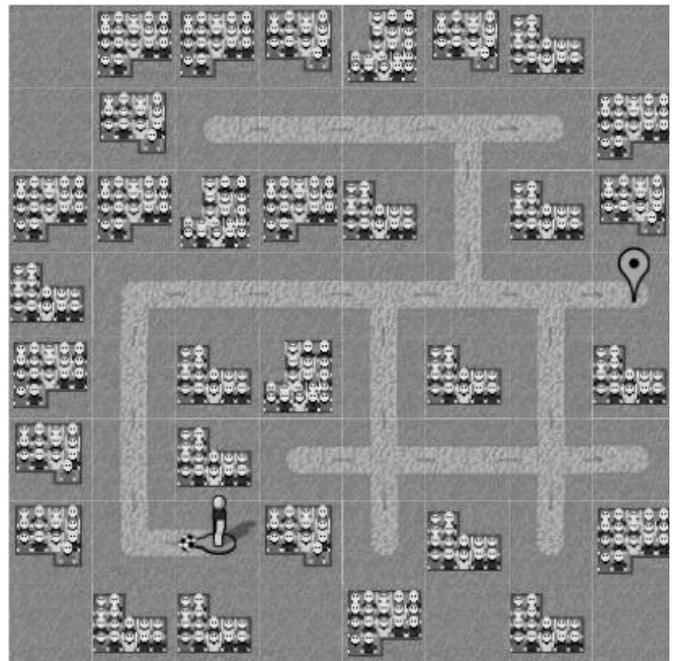
2.1.1 Fluxogramas

Fluxogramas são representados em formas gráficas ordenados em uma sequência lógica representando um algoritmo.

Vantagens: De acordo com Forbellone (2006) as formas gráficas são mais puras por serem fiéis ao raciocínio original substituindo uma grande quantidade de palavras (como usado nos pseudocódigos) por conversões de desenhos e terem clareza no seu fluxo de execução.

Desvantagens: Forbellone (2006) ainda ressalta que apesar dos fluxogramas terem suas vantagens, eles requerem conhecimento das convenções gráficas usadas na produção dos fluxogramas. “Apesar de simples não são naturais, pois estamos mais condicionados a nos expressar por palavras” (FORBELLONE, 2006). Além disso, se torna mais trabalhoso quando a questão é fazer um desenho.

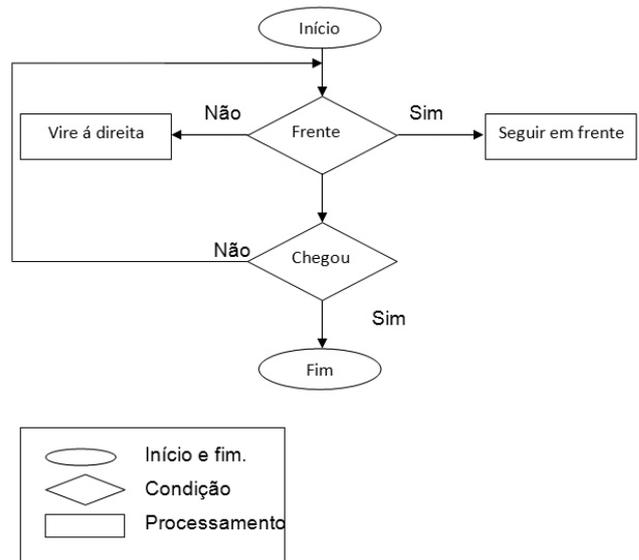
Figura 1 – Exemplo de problema



Fonte: <http://www.anodocodigo.org.br/blockly/apps/maze/index.html?level=18>

A imagem acima representa um problema que usa a lógica para sua resolução, onde o jogador com a bola deverá andar até o ponto de chegada indicado na imagem utilizando estruturas lógicas de programação. Em fluxogramas seria da seguinte forma:

Figura 2 – Resolução do problema lógico com fluxogramas.



2.1.2 Pseudocódigos

Vantagens: Os pseudocódigos são escritos no nosso idioma nativo, que permitem uma boa eficiência

na hora de entender e formular os algoritmos, sem ter que se preocupar com a sintaxe de uma linguagem de programação. No momento do erro pode fazer uma alteração ou correção facilmente. Os textos descrevem uma ordem organizada de execução de um algoritmo.

Desvantagens: Apesar de serem mais fáceis de usar, os pseudocódigos são menos puros, o que exige uma grande quantidade de palavras na formulação de seus algoritmos e pode se tornar cansativo à medida que o algoritmo aumenta. Forbellone (2006) cita como uma das principais desvantagens a ambiguidade do nosso idioma, onde uma frase pode ter vários sentidos. Rapkiewicz (2006) fala de outro problema, a questão da efetividade, tanto nos pseudocódigos como nos fluxogramas são difíceis de fazer testes e validar seu funcionamento.

Para a resolução do exemplo de problema da Figura 1, utilizando Pseudocódigos, ficaria da seguinte forma:

- ñ Início
- ñ Repita até a chegada
 - o Se puder seguir em frente
 - o Siga em frente
 - o Se não
 - o Vire a direita
- ñ Fim

2.1.3 Jogos educacionais

Os jogos estão tornando-se uma das ferramentas mais utilizadas para motivação e estimulação pessoais a aprenderem programação. Grandes empresas vêm aderindo a essa ferramenta para buscar novas pessoas na área da computação, em maio de 2014 a Microsoft lançou um jogo na web que envolve programação, nomeado de Conde Hunt¹ e em 2013 a Microsoft e Facebook juntamente com outras grandes empresas, como a Google, lançaram o Code.org com o intuito de atingir mais de 10 milhões de estudantes. Empresas brasileiras também seguiram a tendência e criaram o Ano do Código² com apoio do [GUJ](#), a [Locaweb](#), [Alura](#), [Globo.com](#), [Code Miner](#), [Caelum](#), e [Casa do Código](#).

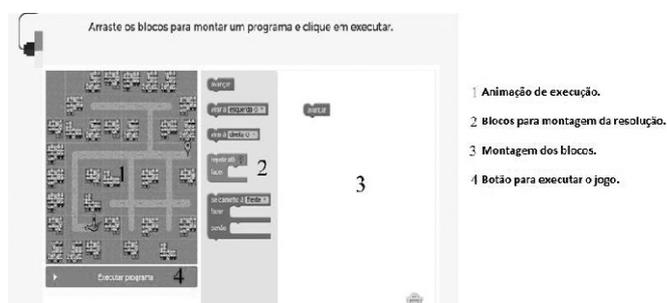
Vantagens: As vantagens de usar jogos na aprendizagem são várias. Jogos forçam o aluno a aprender essências que são importantes no ambiente de desenvolvimento como entender, decidir e priorizar. Tornam o conhecimento divertido e motivador, considerando que o aluno estará em contato com equipamentos multimídias. Jogos usam alternativas criativas para aplicar

o ensino, exigindo o desenvolvimento de habilidades como tomada de decisão e a persistência.

Desvantagens: Jogos educacionais têm o objetivo de assimilar o jogo com o conteúdo da disciplina, pode se tornar difícil a aprendizagem quando o aluno tem um baixo índice de percepção, e não consegue ver semelhança entre a lógica do jogo com a lógica de programação, podendo também se tornar um vício e causar o desinteresse da disciplina.

O exemplo da Figura 1 foi retirado do Ano do Código³ para dar abrangência à iniciativa brasileira e na Figura3 mostram as opções para resolução desse problema e como se estrutura o jogo.

Figura 3 – Estrutura do jogo



Fonte: <http://www.anodocodigo.org.br/blockly/apps/maze/index.html?level=18>

A Figura4 mostra como deveriam ser montadas as estruturas dadas para a resolução do problema. Assim o jogador irá andar até o ponto de chegada indicado na animação.

Figura 4 – Resolução do problema lógico no jogo.



Fonte: <http://www.anodocodigo.org.br/blockly/apps/maze/index.html?level=18>

¹ <https://www.codehunt.com>
² <http://www.anodocodigo.org.br>

3 CONSIDERAÇÕES FINAIS

Para o processo de ensino de lógica de programação, o professor pode utilizar essas metodologias citadas aproveitando o melhor de cada uma, assim construindo um cenário de aprendizagem que possa envolver o máximo de alunos, pois, com mais opções, o aluno irá ter diversas metodologias para assimilar um mesmo conteúdo. Todavia, é fundamental também que os alunos não deixem de ter um contato já nessa disciplina com alguma linguagem de programação, já que será com elas que o aluno irá trabalhar durante sua fase de desenvolvedor nas outras disciplinas e no mercado de trabalho.

REFERENCIAS

Almeida, E. S. ET AL. AMBAP: Um Ambiente de Aprendizado de Programação. In: **ANAIS DO XXII CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO – X WEI**, Florianópolis, Brasil, 2002.

Borges, H. **Portugol –No Ensino de Ciências da Computação**. Disponível em: <<http://www.devmedia.com.br/portugol-no-ensino-de-ciencias-da-computacao/21464>>. Acesso em: 03 julho 2013.

Dazzi, R. L. S.; Miranda, E. M.; Souza, E. M. S. SAEL: Sistema de Apoio ao Ensino de Lógica de Programação. In: **WORKSHOP DE INFORMÁTICA APLICADA À EDUCAÇÃO**, Araraquara, Brasil, 2000.

FORBELLONE, A. L. V. **Lógica de Programação: A Construção de Algoritmos e Estrutura de Dados**. 3. ed. São Paulo: Pearson Prentice Hall, 2005.

Gomes, A. J; Mendes, A. J. N. Suporte à Aprendizagem da Programação com o Ambiente SICAS. In: **V CONGRESSO ILBERO-AMERICANO DE INFORMÁTICA EDUCATIVA**, Viña del Mar, Chile, 2000.

Medeiros, C. L.; Dazzi, R. L. S. Aprendendo Algoritmos com Auxílio da WEB. In: **CONGRESSO BRASILEIRO DE COMPUTAÇÃO**, Itajaí, Brasil, 2002.

Rapkiewicz, C. E; Falkembach, G.; Seixas, L.; Dos Santos Rosa, N.; Da Cunha, V. V.; Klemann, M. Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais, Rio Grande do Sul, v. 4, n. 2, 2006.

Rodrigues J. M. C; Experiências Positivas para o Ensino de Algoritmos. In: **II Workshop de Educação em Computação e Informática. Feira de Santana**, Brasil. 2004.

Rocha, H. V. **Representações computacionais auxiliares ao entendimento de conceitos de programação**. Campinas, 1991, 461p. Tese de Doutorado em En-

genharia Elétrica, Universidade Estadual de Campinas, Brasil, 1991.

Santiago, R; Dazzi R. L. S, **Ferramenta de Apoio ao Ensino de Algoritmos**. Disponível em: <<http://www.inf.furb.br/seminco/2004/artigos/96-vf.pdf>>. Acesso em 10 julho 2013.

Santiago, R; Dazzi, R. L. S. Ferramentas que Auxiliam o Desenvolvimento da Lógica de Programação. In: **SEMINCO –SEMINÁRIO DE COMPUTAÇÃO**, Blumenau, Brasil, 2003. p.113-120.

SOBRE OS AUTORES

José Weliff Ferreira Lima

Acadêmico do curso de Sistemas de Informação na Faculdade Católica Rainha do Sertão – FCRS.

Contato: *weliff.14@gmail.com*

Alandson Mendonça Ribeiro Meireles

Mestre em Computação Aplicada na área de Sistemas de Apoio à Decisão na Universidade Estadual do Ceará (2012), graduado em Ciência da Computação pela Universidade de Fortaleza (2008). Professor da Faculdade Católica Rainha do Sertão. Analista de Sistemas e desenvolvedor Java com experiência na área de Ciência da Computação, com ênfase em Engenharia de Software e Pesquisa Operacional.

Contato: *alandsonmeireles@fcrs.edu.br*