
O Uso de Softwares Educativos para Introdução de Lógica de Programação no Ensino de Base e Superior

The Use of Educational Software to Introduce Programming Logic in Basic and Higher Education

GRACILENE AMÉRICO VIANA

Universidade Federal do Pará (UFPA)

CARLOS DOS SANTOS PORTELA

Universidade Federal do Pará (UFPA)

Resumo: A disciplina de Lógica de Programação costuma ter altos níveis de evasão e reprovação. Isto porque a maioria dos alunos não teve um contato anterior com essa disciplina e seus conceitos, como raciocínio lógico e algoritmos. Alguns pesquisadores brasileiros defendem a inserção da Lógica de Programação no ensino de base, assim como ocorre em outros países. Diante deste contexto, esta pesquisa aborda o uso de softwares educativos para apoiar a introdução à lógica e algoritmos tanto no ensino de base quanto no superior. Assim, apresenta-se uma análise de trabalhos que relatam o uso dos softwares Scratch, Robocode e Visualg. Os resultados desta análise apontaram que alguns desses softwares são mais adequados para o ensino fundamental e médio e outros para a graduação. Por fim, sugerem-se propostas de abordagens para adoção dessas ferramentas em sala de aula.

Palavras-chave: Softwares Educativos. Lógica de Programação. Abordagens de Ensino.

Abstract: The Programming Logic course usually has high levels of avoidance and disapproval. This is because most of the students did not have prior contact with this course and its concepts, such as logical reasoning and algorithms. Some Brazilian researchers defend the inception of Programming Logic in basic education, as it happens in other countries. Given this context, this research addresses the use of educational software to support the introduction to logic and algorithms in both basic and higher education. Thus, we present an analysis of researches that report the use of Scratch, Robocode and Visualg. The results of this analysis pointed out that some of these tools are more suitable for elementary and high school and others for undergraduate. Finally, we present proposals of adopting methodologies for these tools in the classroom.

Keywords: Educational Software. Programming Logic. Teaching Methodologies.

1 Introdução

Lógica de Programação é uma disciplina da área da Computação que tem como propósito ensinar aos alunos meios para resoluções de problemas. No entanto, apesar de sua importância, é uma disciplina que costuma ter altos níveis de evasão e reprovação (FALCÃO; JUNIOR, 2015). Fabri (2007) destaca que, os cursos introdutórios de Lógica de Programação, oferecidos aos alunos das diversas áreas do conhecimento, iniciam com uma média de 50 alunos e em poucos meses, constata-se que taxa de reprovação (ou desistência) chega a 60%. Isto deve-se ao fato de que existe uma grande dificuldade em compreender e aplicar certos conceitos abstratos de programação.

Diversas pesquisas vêm sendo desenvolvidas na área de ensino de Computação, com especial preocupação para o ensino de programação de computadores. A maioria destas pesquisas são voltadas especialmente para estudantes novatos em programação que apresentam grandes dificuldades na aprendizagem de Ciências Exatas (LEAL, 2014). Neste contexto, alguns especialistas discutem as vantagens de ensinar código a crianças e defendem o ensino da Ciência da Computação ao lado de outros cursos elementares, como Matemática, Biologia, Física e Química (MEDEIROS; BRASIL; ARANHA, 2014).

Adicionalmente, Monsalve, Werneck e Leite (2010) defendem que é necessário despertar o interesse do aluno para a aprendizagem através do uso de metodologias ativas, transformando os conteúdos em vivência. Uma destas metodologias é a utilização de softwares educativos, como jogos e simuladores. Para Pietruchinski et al. (2011), a utilização de jogos no processo de ensino-aprendizagem é uma ferramenta capaz de auxiliar no processo educativo, desde que seja planejada e trabalhada de uma forma crítica, que possibilite a aprendizagem de uma maneira significativa ao aprendiz.

Neste sentido, acredita-se que uma criança ou adolescente que comece a ter interações com softwares educativos, que objetivam fomentar o aprendizado de algoritmo e lógica de programação a partir do ensino básico, poderá desenvolver de maneira positiva seu raciocínio lógico e, com isso, ser atraído para a área da Computação que possui uma demanda crescente de profissionais.

Diante desse contexto, esta pesquisa se propõe a fazer uma análise do uso de três softwares educativos que auxiliam o ensino de Lógica de Programação: Scratch, VisuAlg e Robocode. Essa análise foi realizada a partir de artigos com relatos de aplicação destes em sala de aula. Além desta seção introdutória, a Seção 2 apresenta a fundamentação teórica em que se baseou a pesquisa e a Seção 3 os procedimentos metodológicos adotados. Na Seção 4 são descritas as ferramentas selecionadas neste processo e uma análise de usabilidade segundo a ISO/IEC 9126 (2003). A Seção 5 apresenta propostas para aplicação dessas no ensino de base e superior. Por fim, a Seção 6 expõe as considerações finais e os trabalhos futuros dessa pesquisa.

2 Ensino de Lógica de Programação

Atualmente, o ensino de Lógica de Programação no Brasil se limita à educação superior ou pós-graduação. A aplicação da Computação como uma ciência básica é ainda incipiente quando o contexto é a educação básica brasileira. Sendo assim, de acordo com Oliveira et al. (2014), é evidente a importância de viabilizar o conhecimento tecnológico-científico e estimular o desenvolvimento do pensamento lógico pelos estudantes de ensino básico.

A Sociedade Brasileira de Computação (SBC) concorda que os princípios e habilidades introdutórias da Computação devem ser desenvolvidos pelos estudantes desde os primórdios da educação básica, como já ocorre em alguns países (OLIVEIRA et al., 2014). Sabe-se que no Brasil esta realidade é distante, mas acredita-se que ferramentas computacionais podem ser elementos motivadores para o ensino-aprendizagem de disciplinas como Matemática e Física, por exemplo. Em especial, destacam-se as ferramentas computacionais que oferecem

ambientes lúdicos e interativos e com a capacidade de atrair a atenção de crianças e jovens, fomentando o interesse e favorecendo a aprendizagem de conteúdos escolares (AMARAL; SILVA; PANTALEÃO, 2015).

Com isso, o uso de softwares educativos como ferramentas de apoio ao ensino-aprendizagem, em diversas áreas, está cada vez mais frequente (MARCOLINO; BARBOSA, 2015). Na área da Computação, por exemplo, pode-se perceber um aumento significativo no desenvolvimento de softwares que visam melhorar o ensino em conteúdos de difícil compreensão. De acordo com a pesquisa realizada por Marcolino e Barbosa (2015), entre os anos de 2004 a 2014, foram encontrados 81 softwares voltados para apoiar o ensino-aprendizagem em Programação.

Dentre esses, alguns softwares se destacam pela quantidade de estudos e relatos de aplicação publicados em eventos nacionais na área de Ensino e Informática, como no Congresso Brasileiro de Informática na Educação (CBIE), Simpósio Brasileiro de Informática na Educação (SBIE) e Workshop sobre Educação em Computação (WEI), em revistas como a Informática na Educação: teoria & prática, Revista Brasileira de Informática na Educação e Novas Tecnologias na Educação. Adicionalmente, buscou-se relatos de experiências em eventos regionais de escolas técnicas. Não se caracteriza como objetivo desta pesquisa criticar os trabalhos selecionados e nem os comparar, mas sim identificar as abordagens de uso adotadas pelos autores para inserção dos softwares educativos em sala de aula, bem como o nível de ensino: fundamental, médio ou superior.

A seguir, apresentam-se os procedimentos metodológicos seguidos durante a execução dessa pesquisa.

3 Procedimentos Metodológicos

Inicialmente, foi realizada uma revisão da literatura sobre softwares educativos nos anais de eventos e revistas nacionais na área de Ensino e Informática. Esses eventos e periódicos foram selecionados por serem indexados pelo CAPES¹ na área de Ciência da Computação. Dada a baixa quantidade de trabalhos retornados, principalmente relacionados à aplicação desses softwares no ensino fundamental e médio, optou-se por abranger eventos regionais de escolas técnicas, como institutos federais. Assim, identificaram-se 7 (sete) trabalhos sobre o Scratch, 2 (dois) relatos sobre o Visualg e 7 (sete) estudos relacionados ao Robocode.

Após a identificação desses softwares, realizou-se o download e testes das ferramentas no ambiente do grupo de pesquisa <omitido para revisão>. A fim de avaliar a interface e funcionalidades, foi realizada uma análise de usabilidade segundo a ISO/IEC 9126 (2003), descrita na Subseção 4.4.

Em seguida, realizou-se a análise das abordagens relatadas nos estudos selecionados, visando identificar as etapas seguidas e o nível de ensino contemplado. Por fim, essas abordagens foram replicadas no contexto do grupo de pesquisa <omitido para revisão> e oficinas foram planejadas para, futuramente, disciplinas da Universidade <omitido para revisão> no curso de Sistemas de Informação e em turmas do ensino fundamental e médio de escolas da rede pública do município de <omitido para revisão>, a fim de validá-las.

A seção a seguir descreve o objetivo e as principais funcionalidades dos três principais softwares adotados no ensino de Lógica identificados nos eventos e revistas analisados.

4 Softwares Educativos

Para Mota et al. (2014), softwares educativos são programas de computador que visam atender necessidades e possuem objetivos pedagógicos bem estabelecidos. Assim, todo software pode ser considerado educacional, desde que sua utilização esteja inserida num contexto e em uma situação de ensino-aprendizagem, favorecendo seu desenvolvimento e propiciando meios que enriquecem a prática docente.

¹ <http://qualis.capes.gov.br/>

Buscando atingir os objetivos dessa pesquisa, inicialmente realizou-se uma pesquisa bibliográfica em eventos da área de Ensino e Informática sobre a adoção de ferramentas de apoio ao ensino de lógica. Foram selecionados diversos artigos que abordam o uso das ferramentas, destacando-se dentre essas os softwares Scratch, Visualg e Robocode. A partir desses artigos foi feito uma análise dessas ferramentas bem como um estudo dos manuais de cada uma para obter mais detalhes das mesmas. Assim, foi possível coletar os dados necessários para esta análise.

4.1 Scratch

O Scratch é uma ferramenta que permite a criação de histórias interativas, jogos e animações bem como o compartilhamento dessas criações na Web (SCRATCH BRASIL, 2018). Tudo pode ser feito a partir de comandos que são agrupados de modo lógico. Assim, a ferramenta permite estimular a criatividade e a imaginação, não tratando o aprendiz apenas como usuário do software.

Tendo em vista que seu objetivo é auxiliar a aprendizagem de programação, de maneira lúdica e criativa, pode ser usado por crianças a partir de 8 anos de idade e pessoas que não possuem nenhum conhecimento de programação.

Neto (2013) descreve um estudo sobre o uso do Scratch na disciplina de Lógica de Programação. O estudo foi realizado com alunos, entre 16 a 17 anos de idade, do curso Técnico em Informática. Após a aplicação de questionários, concluiu que a utilização da ferramenta trouxe resultados bastante favoráveis, pelo fato de ser um ambiente atraente e motivador, podendo amenizar o problema pertinente à compreensão da lógica computacional e contribuir de forma significativa para a formação do profissional da área. De maneira similar, a oficina realizada por Arantes e Ribeiro (2017) demonstrou as potencialidades do ambiente de programação Scratch na disseminação do pensamento computacional.

Mota et al. (2014) relatam a aplicação de um curso de Lógica de Programação por meio do Scratch para adolescentes do ensino médio em uma escola pública. Antes e depois da realização de oficinas de lógica de programação, foi aplicado um questionário aos alunos participantes a partir do qual se concluiu que os 8 alunos que realizaram a oficina demonstraram muito interesse em continuar o estudo de lógica através dessa ferramenta.

Por fim, Oliveira et al. (2014) relatam a experiência de um projeto de extensão realizado com a finalidade de ensinar conceitos básicos de Ciência da Computação no ensino fundamental. Para tornar possível a compreensão de algoritmos, foi utilizado o Scratch como ferramenta auxiliar pedagógica. Os resultados obtidos mostram que é possível incluir a temática no cotidiano escolar de maneira que desperte o interesse dos alunos da educação básica.

4.2 Visualg

O Visualg é um programa que edita, interpreta e executa algoritmos escritos em pseudocódigo (portugol), criado pelo professor Cláudio Morgado de Souza (APOIO INFORMÁTICA, 2018). Trata-se de um programa livre para uso e distribuição grátis, de domínio público, usado para o ensino de Lógica de Programação em várias escolas e universidades no Brasil. Apresenta-se como uma ferramenta didática que usa o idioma português como base, facilitando a criação e a visualização de algoritmos, permitindo também a checagem de eventuais erros de lógica e digitação de maneira rápida e prática.

De acordo com o seu criador, a linguagem que o VisuAlg interpreta é bem simples: uma versão portuguesa dos pseudocódigos largamente utilizados nos livros de Introdução à Programação, conhecida como "Portugol" (APOIO INFORMÁTICA, 2018).

Leite et al. (2013) descrevem um estudo sobre o potencial do software educacional Visualg no apoio ao ensino da disciplina de Introdução à Programação. Com os resultados obtidos, concluíram que os alunos de programação puderam aprender de maneira mais prática, rápida e agradável os conceitos de algoritmos através das funcionalidades do ambiente, como uma área

para os algoritmos, outra para as variáveis que estão na memória e a visualização dos resultados da execução dos algoritmos.

A partir do estudo de Leite et al. (2013), da análise da documentação e de relatos de experiência, pode-se observar que o Visualg é uma ferramenta bastante eficaz na compreensão dos conceitos de programação e algoritmos por ser prático e mais fácil de usar em relação a outras ferramentas de programação. Adiciona-se a isso o fato de adotar o português como linguagem padrão na criação dos algoritmos, o que facilita a introdução à programação por alunos que não dominam a língua inglesa.

4.3 Robocode

O Robocode é um jogo com um simulador de batalhas entre robôs (tanques de guerra), em que se joga programando em Java (ROBOCODE BRASIL, 2018). Nesse jogo, o aluno (desenvolvedor) pode utilizar as classes bases disponibilizadas pelo ambiente para criação do seu próprio robô e também pode colocá-lo em batalha com outros robôs criados por outros desenvolvedores. Para programar um robô, é necessário estender uma classe nativa do Robocode, podendo-se programar robôs de diversos tipos.

Lima e Santos (2012) descrevem uma proposta para utilização da ferramenta Robocode como uma prática lúdica com o intuito de acelerar o processo de ensino e aprendizagem de Programação Orientada a Objetos (POO). Com base no estudo da ferramenta Robocode, identificaram que a proposta é eficaz, visto que o jogo eletrônico é dinâmico e divertido, possui uma maneira fácil de assimilar conteúdos considerados difíceis, utilizando-se brincadeiras e ao mesmo tempo permitindo superar desafios. Destacam ainda a importância do uso dessa abordagem, que busca tornar o aprendizado de linguagem de programação orientada a objetos (uma tarefa desafiadora) em algo lúdico e motivador.

Já Martins (2015) relata a utilização do Robocode como uma ferramenta capaz de auxiliar na didática e aprendizagem das disciplinas de Programação para Web I e II para alunos do curso Técnico em Informática para Internet, em que foi organizado e realizado um campeonato de Robocode e, posteriormente, desenvolvida uma pesquisa com os alunos em relação à atividade realizada. Os resultados obtidos sugerem fortemente que o Robocode pode ser eficaz na promoção da aprendizagem de POO, visto que a programação se torna mais divertida, pois é efetuada através da criação de robôs para combate em uma arena virtual, mas através de uma atividade prática do mundo real.

4.4 Análise de Usabilidade

Para avaliação dos softwares educativos, antes da inserção deles em sala de aula, os autores avaliaram as 3 ferramentas conforme os princípios de usabilidade da ISO/IEC 9126. As opções de respostas usaram escala Likert (variando de 0 a 4), apresentando a concordância ou discordância dos avaliadores para as perguntas realizadas no que diz respeito à Inteligibilidade, Apreensibilidade, Operacionalidade e Atratividade.

Segundo a ISO/IEC 9126-1 (2003), a Inteligibilidade é a capacidade do produto de software de possibilitar ao usuário compreender se o mesmo é apropriado e como ele pode ser usado para tarefas e condições de uso específicas. Relacionada a essa característica, perguntou-se: "*De acordo com suas impressões iniciais, qual o grau de inteligibilidade da aplicação?*". Apresentaram-se as seguintes opções:

- 0 – Não Apropriado;
- 1 – Parcialmente Apropriado;
- 2 – Apropriado;
- 3 – Largamente Apropriado;
- 4 – Totalmente Apropriado.

A ISO/IEC 9126-1 (2003) define Apreensibilidade como a capacidade do produto de software de possibilitar ao usuário aprender sua aplicação, ou seja, especifica aquilo que é

compreendido facilmente. Foi realizada a seguinte pergunta: "Qual o grau de adequação das funcionalidades da aplicação para o aprendizado do usuário?". As respostas tinham as opções:

- 0 – Não Adequadas;
- 1 – Parcialmente Adequadas;
- 2 – Adequadas;
- 3 – Largamente Adequadas;
- 4 – Totalmente Adequadas.

Já a Operacionalidade é a capacidade do produto de software de possibilitar ao usuário operá-lo e controlá-lo (ISO/IEC, 2003). Abrange aspectos de adequação, modificabilidade, adaptabilidade e capacidade para ser instalado. Operacionalidade corresponde à controlabilidade, tolerância a erros e conformidade com as expectativas do usuário. Assim, indagou-se: "A aplicação permite controle e possui tolerância a erros de acordo com as expectativas do Usuário?". Os valores de 0 (zero) a 4 (quatro) representam as opções:

- 0 – Não Permite;
- 1 – Permite Parcialmente;
- 2 – Permite;
- 3 – Permite Largamente;
- 4 – Permite Totalmente.

Por fim, ISO/IEC 9126-1 (2003) define Atratividade como a capacidade do produto de software de ser atraente ao usuário. Isto se refere a atributos que possuem a intenção de tornar o software mais atrativo para o usuário, como o uso de cores e a natureza do projeto gráfico. A avaliação foi feita a partir da pergunta: "Qual o grau de atratividade do software, considerando o uso adequado de cores e os estilos de interação?". Os valores representam as seguintes opções:

- 0 – Não Atrativo;
- 1 – Parcialmente Atrativo;
- 2 – Atrativo;
- 3 – Largamente Atrativo;
- 4 – Totalmente Atrativo.

A Tabela 1 mostra as notas relacionadas dos 2 avaliadores para cada software e a média obtida.

Tabela 1 – Avaliação de Usabilidade do Scratch, Visualg e Robocode

ISO/IEC 9126	Scratch		Visualg		Robocode	
	Aval. 1	Aval. 2	Aval. 1	Aval. 2	Aval. 1	Aval. 2
Inteligibilidade	4	4	4	4	3	3
Aprensibilidade	4	4	4	4	2	3
Operacionalidade	4	4	4	4	4	4
Atratividade	3	4	4	4	2	2
MÉDIA	3,875		4		2,87	

Fonte: Os Autores.

De acordo com a avaliação realizada, o Visualg é a ferramenta que apresenta melhor usabilidade. Os avaliadores consideraram o software bastante intuitivo, o que facilita o manuseio do usuário com o uso de ícones intuitivos e texto explicativo nos menus. Além disso, oferece manual de ajuda e permite com que o usuário tenha controle de suas operações, pois emite mensagens de confirmação de ação dentro do mesmo e oferece opções alternativas para a realização dessas operações. Por fim, os avaliadores consideraram o Visualg bastante atrativo, devido este permitir com que se configure o tema de interface de acordo com a preferência do usuário, o que faz dele um software versátil e atraente.

Em segundo lugar, fica a ferramenta Scratch que utiliza ícones nos menus, oferece manual de ajuda e possui opção de vários idiomas, facilitando bastante sua adoção em vários países. Seus ícones fazem analogias intuitivas que facilitam o aprendizado do usuário, como por

exemplo o ícone de disquete em Salvar. Adicionalmente, emite mensagens, permitindo com que o usuário tenha controle em casos de erros e oferecendo alternativas na realização de suas funções. A conclusão dos avaliadores foi que o Scratch possui um ambiente bastante atrativo, com uso de cores adequadas e ícones representativos, mas o excesso dessas características acaba “poluindo” um pouco a interface, impactando na sua atratividade.

A ferramenta com pior usabilidade, de acordo com os avaliadores, foi a Robocode, pois apesar de bastante intuitivo, utiliza apenas o idioma inglês, o que pode dificultar um pouco o manuseio por pessoas sem o conhecimento da língua. Por exemplo, os botões de operações não usam ícones para facilitar o aprendizado dos usuários que não entendem inglês. Apesar disso, o aplicativo informa a ocorrência de erros, permitindo realizar operações de correção facilmente. Os avaliadores consideraram que o Robocode usa cores que se contrastam adequadamente, como a tela inicial que possui tela de fundo preta com o nome da ferramenta em verde dando destaque a mesma. No entanto, as demais telas possuem uma combinação de cinza com azul, totalmente diferente da tela inicial, o que torna a interface pouco atrativa.

5 Abordagens de Ensino

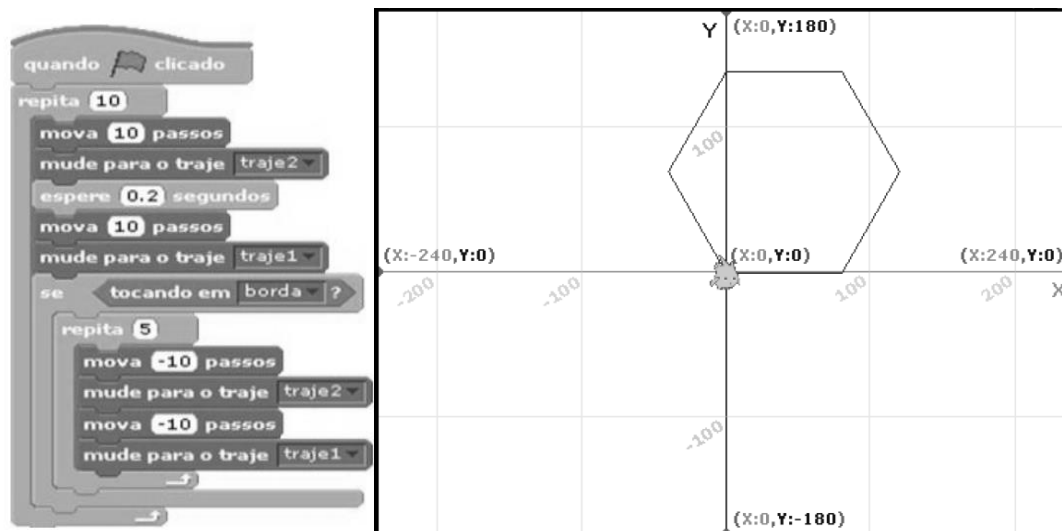
O conceito de abordagem adotada nesta pesquisa é derivado do trabalho de Pantaleão, Amaral e Silva (2017) que define o conjunto de etapas necessárias para o ensino de programação usando um software educativo, abrangendo instruções, diretrizes e sugestões para aplicar essa ferramenta em sala de aula. Assim como em Souza et al. (2013), o uso dessas abordagens objetiva desenvolver nos estudantes as habilidades necessárias para que tenham êxito no desenvolvimento de algoritmos utilizando técnicas de programação.

5.1 Scratch no Ensino Fundamental

Tradicionalmente, uma das disciplinas que os alunos possuem mais dificuldades no ensino de base é a Matemática. De acordo com o Sistema de Avaliação da Educação Básica (SAEB, 2015), os últimos resultados apontaram que grande parte das escolas brasileiras não possuem as competências básicas em Matemática, como a habilidade de resolver problemas. Essa habilidade pode justamente ser desenvolvida através da construção de algoritmos e, mais especificamente no Ensino Fundamental, a partir de um ambiente de programação voltado particularmente para o público infantil, como o Scratch.

A abordagem de aplicação do Scratch sugerida é baseada na proposta de Andrade, Silva e Oliveira (2013). Assim, primeiramente deve-se apresentar o ambiente de trabalho do Scratch, mostrando aos alunos suas funcionalidades e alguns projetos já prontos. Em seguida, deve-se trabalhar com os alunos as primeiras noções de algoritmo usando os blocos de comandos existentes no ambiente. A partir desse ponto, os alunos podem iniciar a criação dos seus primeiros projetos para treinar a linguagem recém-aprendida. Nessa etapa, por exemplo, os alunos podem criar algoritmos onde o resultado é o desenho de figuras geométricas, conforme Figura 1.

Figura 1 – Desenho de Figuras Geométricas no Scratch



Fonte: ANDRADE; SILVA; OLIVEIRA, 2013.

Por fim, os alunos, guiados pelo tutor, podem desenvolver os primeiros jogos tendo a Matemática como tema principal. A recomendação é que possam produzir algoritmos com desafios para seus colegas utilizando conteúdos específicos dessa disciplina, como operações aritméticas ou geometria.

Uma outra possibilidade, apresentada em Wangenheim, Nunes e Santos (2014) e Lisboa, Monte-Alto e Silva (2017), consiste na criação das histórias interativas. De maneira semelhante a abordagem anterior, se faz necessárias explicações do passo-a-passo pelos instrutores mostrando como usar comandos do Scratch no início de cada tarefa e, principalmente, a assistência durante a realização da tarefa, quando os alunos aplicam o conhecimento e buscam suas próprias soluções.

No entanto, Arantes e Ribeiro (2017) afirmam que não se deve solicitar muitos exercícios para não desmotivar os alunos, destacando que o ideal é solicitar um ou dois exercícios de cada vez. Em uma de suas aulas, a instrutora solicitou muitos exercícios de uma só vez, o que não foi produtivo, pois os alunos não conseguiram fazer quase nenhum, porque preferiram explorar todos os exercícios ao invés de manter o foco na solução de um deles.

5.2 Visualg no Ensino Médio

O ensino de Física no Ensino Médio também apresenta grandes dificuldades, por vários fatores, destacando-se dentre eles a falta de motivação dos alunos (COSTA; BARROS, 2015). Segundo Gomes e Castilho (2009), a metodologia de ensino na educação brasileira é comumente vista como transmissão de conteúdos aos alunos, realização de exercícios repetitivos, memorização e definições de fórmulas e conceitos. Assim, ao ingressarem em um curso de graduação em computação, muitos estudantes se decepcionam e perdem a motivação ao se depararem, já no primeiro semestre do curso, com disciplinas que exigem habilidades de abstração e de raciocínio lógico desenvolvidas no ensino fundamental e médio (LEITE et al., 2013).

Neste contexto, a recomendação consiste em utilizar a ferramenta Visualg para resolução de cálculos físicos ou matemáticos, como proposto em Souza et al. (2013), principalmente os exigidos no Exame Nacional do Ensino Médio (ENEM). Por exemplo, uma fórmula física bastante

utilizada nas provas do ENEM é a do Movimento Retilíneo Uniforme (MRU). A Figura 2 apresenta um algoritmo, desenvolvido em Portugol, que implementa essa fórmula e resolve uma questão exemplo.

$$S = S_0 + V.t$$

Figura 2 – Algoritmo da fórmula MRU implementada no Visualg

```

1 Algoritmo "Cálculo"
2 //
3 //
4 // Descrição   : Calcula a fórmula do Movimento Retilíneo Uniforme (MRU)! (função)
5 // Autor(a)    : Gracilene
6 // Data atual  : 07/03/2018
7 Var
8 // Seção de Declarações das variáveis
9 S, So, V, t : real
10
11 Inicio
12 // Seção de Comandos, procedimento, funções, operadores, etc...
13 So <- 0      // Atribui um valor a variavel espaço inicial
14 V <- 80     // Atribui um valor a variavel velocidade
15 t <- 45     // Atribui um valor a variavel tempo
16 S <- So + V*t // Calcula a formula e atribui o resultado a variavel espaço final
17 escreval ("S = ", S) // Imprime o resultado da formula
18
19 Fimalgoritmo
    
```

Fonte: Autores

A abordagem sugerida é semelhante à adotada para o Scratch, onde primeiramente deve-se apresentar o ambiente de trabalho do Visualg, mostrando aos alunos alguns algoritmos básicos de exemplos. Após este momento, os alunos podem iniciar a criação dos seus primeiros algoritmos para treinar a linguagem Portugol. Por fim, os alunos, guiados pelo tutor, podem desenvolver os primeiros algoritmos para aplicar fórmulas físicas ou matemáticas.

Na experiência realizada por Souza et al. (2013), cujo objetivos consistiram em explorar o raciocínio lógico dos estudantes e prepará-los para o estudo dos algoritmos e analisar a influência do idioma do software (português estruturado) na aprendizagem. Os resultados apontaram avanços no desenvolvimento de algoritmos e que o idioma inglês, geralmente nativo das linguagens de programação, não é o problema chave no ensino-aprendizagem de algoritmos. As maiores dificuldades apontadas pelos estudantes foram o entendimento do problema matemático e a capacidade de raciocinar logicamente.

5.3 Robocode na Graduação

Aprender uma linguagem de programação é algo complexo, principalmente se o paradigma for desconhecido dos alunos. Um dos principais paradigmas ensinados na graduação é a Programação Orientada a Objetos (POO), cujo Robocode adota os conceitos através da linguagem Java. Neste sentido, o jogo é baseado em eventos, por exemplo: ver um robô inimigo e atirar. A ferramenta possibilita, ainda, a utilização de troca de mensagens, encapsulamento, herança e polimorfismo. Todos esses conceitos são referentes à POO. Por este motivo, sugere-se a adoção do Robocode somente no nível superior, para alunos que já tenham uma noção consolidada de algoritmos e lógica de programação.

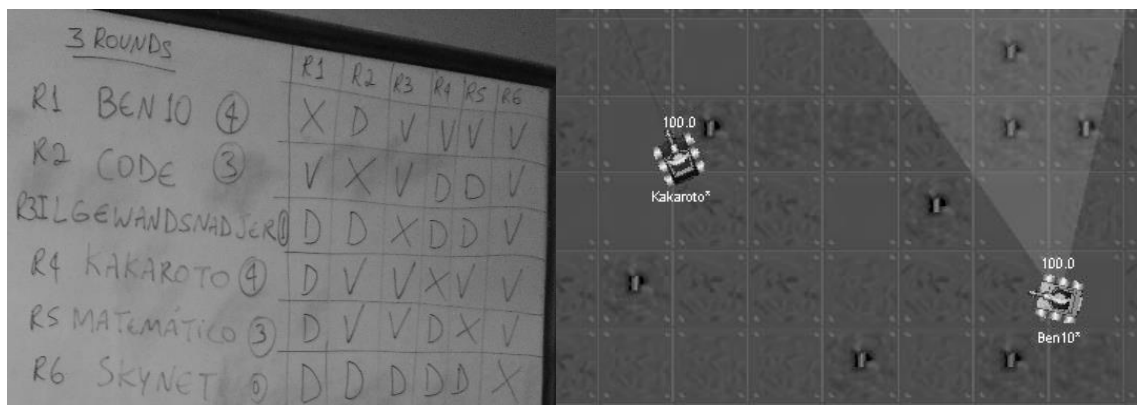
A abordagem de adoção proposta é baseada nas experiências de Ferreira et al. (2012), Lima, Dos Santos e Pais (2017) e Martins (2018), onde inicialmente deve-se realizar o estudo do ambiente do Robocode e dos códigos-fonte dos robôs. Posteriormente, os alunos devem alterar o código-fonte base disponibilizado pela ferramenta, com o auxílio de um monitor, e implementar o seu próprio robô. Por fim, realiza-se um torneio de batalhas entre robôs, através

de grupos onde todos se enfrentam e os melhores se classificam ou “mata-mata” (dependendo da quantidade de alunos), onde ao final deve haver um campeão.

De acordo com a experiência de Martins (2018), após o torneio, os alunos relataram que houve uma interação maior entre eles, que se sentiram desafiados com a atividade e isso contribuiu para sua permanência no curso de Sistemas de Informação. Alguns dos benefícios citados pelos alunos foram: o trabalho em equipe, a melhor compreensão do conteúdo da matéria e ter agregado conhecimento ao currículo. Já no relato de Lima, Dos Santos e Pais (2017), os alunos mostraram-se interessados em aprender a usar o Robocode e ficou evidente que estavam aprendendo a programar enquanto se divertiam. Com isso foi possível constatar que a ferramenta pode auxiliar o aprendizado dos alunos de uma forma lúdica.

A Figura 3 mostra um torneio de Robocode realizado pelos autores deste artigo, seguido essa abordagem, numa turma do primeiro semestre do curso de Bacharelado em Sistemas de Informação, turma de 2018 da Universidade <omitido para revisão>.

Figura 3 –Torneio de Robocode.



Fonte: Autores

6 Considerações Finais

Neste artigo são apresentados os resultados de uma análise de uso dos softwares Scratch, Robocode e Visualg através de um estudo em que identificou o apoio de cada um desses softwares à Introdução de Lógica de Programação. As principais contribuições desta pesquisa consistiram na identificação das abordagens de ensino para adoção desses softwares educativos em sala de aula e na análise do nível de formação em que esses vem sendo adotados com resultados satisfatórios.

Essa análise aponta que essas ferramentas podem ser fortes aliadas no ensino-aprendizagem, desde que adotadas no nível de formação adequado. Observou-se, também, que cada uma dessas ferramentas possui um perfil de aprendizagem mais adequado para introduzir conceitos de lógica em um determinado grau de formação dos alunos.

O Scratch, por exemplo, é uma ferramenta que apresenta bons resultados quando utilizados para introduzir conceitos de programação a alunos do ensino fundamental (ANDRADE, SILVA e OLIVEIRA, 2013) (WANGENHEIM, NUNES e SANTOS, 2014) (LISBÔA, MONTE-ALTO e SILVA, 2017) (ARANTES e RIBEIRO, 2017). Já o Visualg oferece um nível de aprendizado mais avançado que vem se mostrando bastante eficiente quando utilizado por alunos de nível médio e de cursos técnico da área (LEITE et al., 2013) (SOUZA et al., 2013). Assim como esses autores, acredita-se que a adoção do Scratch e do Visualg para estudantes do ensino fundamental e médio se constitui como uma estratégia essencial de captação de alunos para a área da Computação, pois favorecem o desenvolvimento do raciocínio lógico. Esse contato

prematureo com algoritmos e lógica e programação pode prepará-los para cursar as disciplinas introdutórias dos cursos de Computação.

Por fim, o Robocode vem sendo muito utilizado como elemento motivador para introduzir conceitos de Programação Orientada a Objetos a alunos do nível superior (FERREIRA et al., 2012) (LIMA, DOS SANTOS e PAIS, 2017) (MARTINS, 2018). O uso de jogos educativos na graduação, como relatado em Monsalve, Werneck e Leite (2010), Medeiros, Brasil e Aranha (2014) e Falcão e Junior (2015), traz os componentes motivação e engajamento, auxiliando no ensino de POO em Java, paradigma no qual os calouros dos cursos de Computação apresentam bastante dificuldades (LIMA e SANTOS, 2012) (MARTINS, 2015).

Diante disso, como trabalhos futuros, pretende-se aplicar estudos de caso utilizando o Scratch com alunos do ensino fundamental, o VisuAlg com alunos do ensino médio e o Robocode com alunos iniciantes do curso de Sistemas de Informação a fim de verificar se essas ferramentas, no ensino básico, irão atrair alunos do fundamental e médio a ingressar na área de Computação e, no ensino superior, se irão motivar alunos da graduação a aprender lógica, algoritmos e linguagens de programação de maneira mais satisfatória que a abordagem tradicional de ensino.

Referências

AMARAL, L.; SILVA, G.; PANTALEÃO, E. Plataforma Robocode como Ferramenta Lúdica de Ensino de Programação de Computadores-Extensão Universitária em Escolas Públicas de Minas Gerais. Em: Simpósio Brasileiro de Informática na Educação (SBIE), p. 200-208, 2015.

ANDRADE, M.; SILVA, C.; OLIVEIRA, T. Desenvolvendo games e aprendendo matemática utilizando o Scratch. Em: Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames), 2013.

APOIO INFORMÁTICA. A Linguagem de Programação do VisuAlg. Disponível em <http://www.apoioinformatica.inf.br/produtos/visualg>. Acesso em Agosto de 2018.

ARANTES, F.; RIBEIRO, P. Desenvolvimento do Pensamento Computacional com Valores da Ética Hacker. *Informática na Educação: teoria & prática*, Porto Alegre, v. 20, n. 2, p. 188-206, mai./ago. 2017.

COSTA, L.; BARROS, M. O Ensino da Física no Brasil: Problemas e Desafios. Em: VII Congresso Nacional de Educação (EDUCERE), 2015.

FABRI, J. O Ensino de Lógica de Programação e o Desenvolvimento de Jogos Educacionais: Um Caso Aplicado aos Alunos do Curso de Licenciatura Plena em Matemática. *Revista Tecnologias, Sociedade e Conhecimento*, 2007.

FALCÃO, E.; JUNIOR, J. Desenvolvimento de jogos eletrônicos como metodologia de ensino de Programação para alunos do curso de informática do Instituto Federal Catarinense – Campus Camboriú. Em: VIII Mostra Nacional de Iniciação Científica e Tecnológica Interdisciplinar (MICTI), 2015.

FERREIRA, M.; FILHO, R.; RODRIGUES, F.; SÁ, S.; SILVA, J. Robocode: um auxílio no aprendizado da linguagem Java. Em: Encontros Universitários, UFC Quixadá, 2012.

GOMES, J.; CASTILHO, W. Uma Visão de como a Física é ensinada na Escola Brasileira, e a experimentação como estratégia para mudar essa realidade. Em: 1ª Jornada de Iniciação Científica e Extensão do IFTO, 2009.

LEAL, A. Ensino de Programação no Ensino Médio Integrado: Uma Abordagem Utilizando Padrões e Jogos com Materiais Concretos. 2014. Dissertação (mestrado) - Instituto de Informática, Universidade Federal de Goiás, 2014, Goiânia, BR-GO.

LEITE, V.; SENEFONTE, H.; BARBOSA, C.; SEABRA, R. VisuAlg: Estudo de Caso e Análise de Compilador destinado ao ensino de Programação. *Nuevas Ideas en Informática Educativa*, p. 637-640, 2013.

LIMA, C.; SANTOS, M. A utilização da ferramenta Robocode no aprendizado de linguagem de programação orientada a objetos. Em: III Jornada de Iniciação à Docência (PIBID), 2015, Vitória, BR-ES.

- LIMA, J.; DOS SANTOS, K.; PAIS, S. Robocode no Ensino de Programação de Computadores. Em: VI Congresso Estadual de Iniciação Científica e Tecnológica do IF Goiano, 2017, Urutaí, BR-GO.
- LISBÔA, MONTE-ALTO E SILVA. Clubes de Programação com Scratch nas Escolas e a Interdisciplinaridade. Em: Congresso Brasileiro de Informática na Educação (CBIE - Workshops), p. 1174-1178, 2017.
- MARCOLINO, A.; BARBOSA, E. Softwares Educacionais para o Ensino de Programação: Um Mapeamento Sistemático. Em: Simpósio Brasileiro de Informática na Educação (SBIE), p. 190-199, 2015.
- MARTINS, E. Utilização do Robocode como Ferramenta de Auxílio ao Ensino de Programação Orientada a Objeto na Disciplina de Programação para Web. Em: III Jornada de Iniciação à Docência (PIBID), 2015, Vitória, BR-ES.
- MARTINS, E. Uso do Robocode no ensino de programação em um curso médio e superior. *Tecnia – Revista de Educação, Ciência e Tecnologia do IFG*, p. 198-211, 2018.
- MEDEIROS, T.; BRASIL, P.; ARANHA, E. Um framework para criação de jogos voltados para o ensino de lógica de programação. Em: Congresso Brasileiro de Informática na Educação (CBIE), p. 200-208, 2014.
- MONSALVE, E.; WERNECK, V.; LEITE, J. Simules-W: Um Jogo para o Ensino de Engenharia de Software. Em: Fórum de Ensino em Engenharia de Software (FEES), 2010, Salvador, BR-BA.
- MOTA, F.; RIBEIRO, N.; EMMENDORFER, L.; BUTZEN, P.; MACHADO, K.; ADAMATTI, T. Desenvolvendo o Raciocínio Lógico no Ensino Médio: uma proposta utilizando a ferramenta Scratch. Em: Simpósio Brasileiro de Informática na Educação (SBIE), p. 377-381, 2014.
- NBR ISO/IEC 9126-1. Engenharia de software - Qualidade de produto. Parte 1: Modelo de qualidade. Associação Brasileira de Normas Técnicas (ABNT), 2003.
- NETO, V. (2013). A utilização da ferramenta Scratch como auxílio na aprendizagem de lógica de programação. Em: Congresso Brasileiro de Informática na Educação (CBIE - Workshops), p. 260-269, 2013.
- OLIVEIRA, M.; SOUZA, A.; BARBOSA, A.; BARREIROS, E. Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência. Em: Workshop sobre Educação em Computação (WEI), p. 1525-1534, 2014.
- PANTALEÃO, E.; AMARAL, L.; SILVA, G. Uma abordagem baseada no ambiente Robocode para ensino de programação no Ensino Médio. *Revista Brasileira de Informática na Educação*, V. 25, N. 3, p. 95-111, 2017.
- PIETRUCHINSKI, M.; NETO, J.; MALUCELLI, A.; REINEHR, S. Os jogos educativos no contexto do SBIE: uma revisão sistemática de Literatura. Em: Simpósio Brasileiro de Informática na Educação (SBIE), 2011.
- ROBOCODE BRASIL. Tutorial Robocode. Disponível em <http://www.robocodebrasil.com.br/tutoriais.php>. Acesso em Agosto de 2018.
- SAEB. Resultados do SAEB/2015. Disponível em <http://sistemasprovabrasil.inep.gov.br/provaBrasilResultados/>. Acesso em Julho de 2018.
- SBC. Currículo de Referência da SBC para Cursos de Graduação em Bacharelado em Ciência da Computação e Engenharia de Computação. Disponível em <http://www.sbc.org.br/documentos-da-sbc/summary/131-curriculos-de-referencia/760-curriculo-de-referencia-cc-ec-versao2005>. Acesso em Julho de 2018.
- SCRATCH BRASIL. Tutorial Scratch. Disponível em <http://www.scratchbrasil.net.br/index.php/materiais/tutoriais.html>. Acesso em Agosto de 2018.
- SOUZA, M.; MOREIRA, J.; LOBO, F.; ALENCAR, M. Uma Abordagem Metodológica voltada para o Ensino-Aprendizagem de Algoritmos. *Novas Tecnologias na Educação*, V.11, N. 1, julho, 2013.
- WANGENHEIM, C.; NUNES, V.; SANTOS, G. Ensino de Computação com SCRATCH no Ensino Fundamental – Um Estudo de Caso. *Revista Brasileira de Informática na Educação*, V. 22, N. 3, 2014.

Recebido em março de 2019

Aprovado para publicação em abril de 2019

Gracilene Américo Viana

Faculdade de Sistemas de Informação – Universidade Federal do Pará – UFPA, Campus do Tocantins/Cametá – CUNTINS, Brasil, gracaviana.1988@gmail.com

Carlos dos Santos Portela

Faculdade de Sistemas de Informação – Universidade Federal do Pará – UFPA, Campus do Tocantins/Cametá – CUNTINS, Brasil, csp@ufpa.br